MARILENA MARTINS PAMBOUKIAN

**MODELAGEM METABÓLICA E MATEMÁTICA DO
COMPORTAMENTO CINÉTICO DE CÉLULAS S2 DE *Drosophila*
melanogaster ADEQUADA À SUA FLEXIBILIDADE METABÓLICA**

**São Paulo
2013**

MARILENA MARTINS PAMBOUKIAN

**MODELAGEM METABÓLICA E MATEMÁTICA DO
COMPORTAMENTO CINÉTICO DE CÉLULAS S2 DE *Drosophila
melanogaster* ADEQUADA À SUA FLEXIBILIDADE METABÓLICA**

Tese apresentada à Escola Politécnica da
Universidade São Paulo para obtenção do
Título de Doutora em Engenharia.

Área de Concentração: Engenharia
Química

Orientador: Prof. Dr. Aldo Tonso
Co-orientadora: Profa. Dra. Elisabeth
Fátima Pires Augusto

**São Paulo
2013**

Autorizo a reprodução e divulgação total ou parcial deste trabalho , por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 06 de fevereiro de 2013.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Pamboukian, Marilena Martins

Modelagem metabólica e matemática do comportamento cinético de células S2 de drosophila melanogaster adequada à sua flexibilidade metabólica / M.M. Pamboukian -- ed.rev. --São Paulo, 2013.

400 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.

1. Metabolismo 2. Drosophila melanogaster 3. Simulação 4. Modelagem matemática 5. Modelagem metabólica I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química II. t.

Nome: PAMBOUKIAN, Marilena Martins

Título: Modelagem Metabólica e Matemática do comportamento cinético de células
S2 *Drosophila melanogaster* adequada à sua flexibilidade

Tese apresentada à Escola Politécnica da
Universidade São Paulo para obtenção do
Título de Doutora em Engenharia Química.

Aprovada em:

Banca Examinadora

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof. Dr. _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Ao Sergio por todo amor, incentivo, companheirismo, ensinamentos e por tudo!!

A minha mãe, meus irmãos Rubens e Sueli e aos meus filhos-sobrinhos Rodrigo e Alexandre por toda a força, carinho e compreensão.

AGRADECIMENTOS

Ao Prof. Dr. Aldo Tonso, meu orientador e amigo, obrigada pela oportunidade.

A Profa. Dra. Elizabeth Augusto, minha co-orientadora e amiga, pelos bons momentos de discussão.

Ao Dr. Carlos Augusto Pereira, pela contribuição de infraestrutura e tantos ensinamentos, conselhos e apoio.

A Profa. Dra. Soraia Attie Calil Jorge por tantos ensinamentos em biologia, passados com muito carinho e por sua amizade verdadeira.

Ao Dr. Renato Astray por boas discussões biológicas e por sempre me apoiar e incentivar.

Ao meu amigo-filho Marcelo Aguiar por todo carinho, amizade e pelos ensaios!

Aos meus amigos-filhos Daniella, Marcos e Cássia por toda amizade, troca de ideias e por me apoiarem e me ajudarem.

Aos maravilhosos e queridos amigos do LIV: Lúcia, Marisa, Vera, Ana Lia, Fernanda, Thaissa, Nayara, Juliana, Alexandre, Polyana, Isabel, Gustavo e Ana Elena por todas as discussões, apoio, incentivo e amizade que tornaram o dia a dia mais prazeroso.

Aos funcionários Elisete, Alexandre e Graça (Bloco 19/USP) que sempre estiveram prontos a colaborar no que fosse necessário.

Aos colegas do Projeto Temático, por todas as discussões e momentos de lazer e alegria.

A minha amiga-filha Adriana e seu príncipe Jonathan por todo apoio, carinho e presença certa em minha vida em todos os momentos.

A minha amiga Mônica, que mesmo a distância, sempre está presente em minha vida.

Aos meus pais Rubens (*in memoriam*) e Angelina que me educaram com amor para sempre respeitar a vida e ao próximo.

Ao meu irmão Rubens, pelo amor incondicional e por me ensinar a ser profissional, minha cunhada Sueli e meus sobrinhos Rodrigo e Alexandre que sempre estão presentes em minha vida tornando-a repleta de amor, amizade e felicidade.

Ao meu marido Sergio que é tudo o que alguém pode sonhar.

A minha tia Maria por todo amor e carinho que sempre nos deu.

Aos meus sogros, Vicente (*in memoriam*) e Íris e aos meus cunhados Celso e Maria Izabel, que são parte de minha vida, cobrindo-a com muito amor e companheirismo.

Gostaria, também, de agradecer o privilégio e a honra de ter trabalhado com um grupo tão seletivo e interdisciplinar quanto deste temático.

A toda a minha família e amigos por todo amor, incentivo e carinho, sem a qual eu nada seria.

A todos que direta ou indiretamente colaboraram com a realização deste trabalho.

Ao CNPq e Fapesp pelo suporte financeiro.

Apesar de tudo....

"Se você pode sonhar, você pode fazer, pois todos os nossos sonhos podem se tornar realidade, se tivermos coragem de persegui-los"

(Walt Disney)

"Sem riscos não conheceríamos o sabor das derrotas nem o paladar das vitórias, pois elas seriam um destino inevitável, não fruto de batalhas. É necessário correr riscos para transformar sonhos em realidade. A existência é um contrato de riscos"

(Augusto Curry)

RESUMO

PAMBOUKIAN, M. M. **Modelagem Metabólica e Matemática do comportamento cinético de células S2 de *Drosophila melanogaster* adequada em sua flexibilidade metabólica**, 2012, 390 f. Tese (Doutorado) - Departamento de Engenharia Química da Universidade de São Paulo, São Paulo, 2012.

O metabolismo das células S2 (Schneider 2) de *Drosophila melanogaster* ainda não é totalmente conhecido. Existem poucos estudos específicos sobre o metabolismo de células S2, sejam elas selvagens ou recombinantes (rS2), como por exemplo aquelas transfectadas para a expressão da glicoproteína do vírus da raiva (GPV). Como o genoma da *Drosophila melanogaster* já foi mapeado, as principais enzimas que atuam nos processos metabólicos em geral já foram identificadas e estão à disposição no KEGG (*Kyoto Encyclopedia of Genes and Genomes*). Assim, o KEGG apresenta todas as possíveis vias metabólicas com as enzimas que podem ser codificadas. Diante deste quadro, foi proposto um modelo metabólico baseado em um conjunto de vias de assimilação de glicose e glutamina e foram encontrados os modos elementares característicos do sistema através do programa Metatool. Em seguida, foi definido o modelo matemático mediante o equacionamento desses modos elementares. Esse processo se repetiu até se encontrar um conjunto de vias metabólicas que, através da modelagem matemática, respondesse coerentemente a um conjunto de dez ensaios em diferentes condições de concentrações iniciais de glicose, glutamina e oxigênio dissolvido. Chegou-se então, a um metabolismo básico para a rS2 contendo 33 vias metabólicas englobando a glicólise, a via das pentoses, o ciclo de Krebs e a fosforilação oxidativa. Dados anteriores indicavam elevada flexibilidade metabólica dessa célula, o que foi prevista através de algumas reações propostas como reversíveis nas vias de degradação e síntese de glutamina. Essa proposta de metabolismo resultou em 37 modos elementares. Outra característica interessante da modelagem foi a utilização da produção de purinas e pirimidinas para a estimativa do crescimento celular. Depois de realizada a modelagem, as mesmas condições iniciais dos ensaios foram simuladas através de um programa de simulação do comportamento cinético das células rS2 desenvolvido em MATLAB. Esse simulador foi utilizado também para simulação com diferentes meios e condições iniciais de cultivo. Chegando-se a um ajuste geral entre valores experimentais e simulados com coeficiente de correlação de 0,88.

Palavras Chave: Modelagem Matemática, Modelagem Metabólica, Metabolismo, *Drosophila melanogaster*, Simulação.

ABSTRACT

PAMBOUKIAN, M. M. **Metabolic and Mathematical Modelling of kinetic behavior of *Drosophila melanogaster* S2 cells appropriate to their metabolic flexibility**, 2012, 390 f. Tese (Doutorado) - Departamento de Engenharia Química da Universidade de São Paulo, São Paulo, 2012.

The metabolism of the S2 cells (Schneider 2) *Drosophila melanogaster* is not yet fully known. There have been few specific studies on the metabolism of S2 cells, whether recombinant or wild (rS2), such as those transfected for expressing the rabies virus glycoprotein (RVGP). As the genome of *Drosophila melanogaster* have been mapped, the key enzymes that act on the metabolic processes in general have been identified and are available in the KEGG (Kyoto Encyclopedia of Genes and Genomes). Thus, KEGG presents all possible pathways with the enzymes that can be encoded. Given this context, it was proposed a metabolic model based on a set metabolic glucose and glutamine assimilation pathways and were found characteristic elementary modes of the system through the Metatool program. Then the mathematical model was defined by addressing these elementary modes. This process was repeated until a set of metabolic pathways, by mathematical modelling, consistently responded to a set of ten experiments (in various conditions). We came to a basic metabolism for rS2 containing 33 pathways comprising glycolysis, pentose, Krebs cycle and oxidative phosphorylation. Previous data indicate that rS2 is a cell with high metabolic flexibility, which was confirmed by some reactions in the process proposed as reverse breakdown and synthesis of glutamine. The proposed metabolism resulted in 37 elementary modes. Another interesting model characteristic was the use of the production of purines and pyrimidines for the estimation of cell growth. After the modelling performed, the same initial runs conditions were simulated using a software of Simulation of the Kinetic behaviour of rS2 cells, developed in MATLAB. This simulator was also used for simulation of other experiments with different initial conditions and methods of cultivation. Coming to a general adjustment of experimental and simulated values with correlation coefficient of 0.88.

Keywords: Mathematical Modelling, Metabolic Modelling, Metabolism, *Drosophila melanogaster*, Simulation.

LISTA DE ILUSTRAÇÕES

Figura 1 – <i>Drosophila melanogaster</i> , macho (esquerda) e fêmea (direita)	39
Figura 2 – Esquema do vírus da raiva	41
Figura 3 – Esquema das principais vias metabólicas para a Sf9	46
Figura 4 - Tela do <i>software</i> Lissage	62
Figura 5 - Tela do <i>software</i> Emerson	63
Figura 6 - Tela do <i>software</i> Excel	63
Figura 7 - Mínimos quadrados	82
Figura 8 - Distribuição gaussiana (normal) para uma variável x , com média m e desvio padrão s	84
Figura 9 - Diagrama de blocos da metodologia utilizada no desenvolvimento deste trabalho	98
Figura 10 - Comparativo 1 entre os erros gerados pelos ajustes dos programas AjustaPar e AjuPar	110
Figura 11 - Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 1, com os dados originais (símbolos) e curva calculada pela simulação (linha)	112
Figura 12 - Comparativo 2 entre os erros gerados pelos ajustes dos programas AjustaPar e AjuPar	113
Figura 13 - Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 2, com os dados originais (símbolos) e curva calculada pela simulação (linha)	114
Figura 14 - Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 3, com os dados originais (símbolos) e curva calculada pela simulação (linha)	116
Figura 15 - Metabolismo proposto para a célula rS2 – Modelo 1	124

Figura 16 - Simulações do Modelo 1. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio	130
Figura 17 - Simulações do Modelo 1. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio	131
Figura 18 - Simulações do Modelo 1. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio	132
Figura 19 - Metabolismo proposto para a célula rS2 com a adição da via GOGAT (v ₁₉) - Modelo 2	133
Figura 20 - Metabolismo da glutamina, conforme está apresentado no KEGG, onde as enzimas destacadas em verde podem estar presentes na <i>Drosophila melanogaster</i>	134
Figura 21 - Nove Modos Elementares oriundos da análise das vias metabólicas para a proposta do Modelo 2	134
Figura 22 - Simulações do Modelo 2. Os dados experimentais são representados por símbolos e a linha representa os dados simulados de cada ensaio	136
Figura 23 - Simulações do Modelo 2. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio	137
Figura 24 - Proposição do Modelo 3 para a célula rS	138
Figura 25 - Modelo 3.05 para condições de excesso de GLC e/ou GLN para a célula rS2 na fase exponencial	140
Figura 26 - Simulação para o Modelo 3.05 para condições de excesso de GLC e GLN para a célula rS2 na fase exponencial	141
Figura 27 - Modelo 3.07 para condição limitante de GLC para a célula rS2 na fase exponencial	142
Figura 28 - Simulação para o Modelo 3.07 para condição limitante de GLC para a célula rS2 na fase exponencial	143

Figura 29 - Modelo 3.08 para condição limitante de GLN para a célula rS2 na fase exponencial	144
Figura 30 - Simulação para o Modelo 3.08 para condição limitante de GLC para a célula rS2 na fase exponencial	145
Figura 31 - Modelo Metabólico básico 4 com GOGAT - Modelo 4A	146
Figura 32 - Representação das vias metabólicas do Modelo 5	152
Figura 33 - Resíduos e Simulação do Ensaio Ma-07 em fase exponencial	157
Figura 34 - Resíduos e Simulação do Ensaio Ma-08 em fase exponencial	158
Figura 35 - Resíduos e Simulação do Ensaio Ma-09 em fase exponencial	159
Figura 36 - Resíduos e Simulação do Ensaio Ma-10 em fase exponencial	160
Figura 37 - Resíduos e Simulação do Ensaio Ma-11 em fase exponencial	161
Figura 38 - Resíduos e Simulação do Ensaio Ma-12 em fase exponencial	162
Figura 39 - Resíduos e Simulação do Ensaio Ma-14 em fase exponencial	163
Figura 40 - Resíduos e Simulação do Ensaio Ma-15 em fase exponencial	164
Figura 41 - Resíduos e Simulação do Ensaio Ma-16 em fase exponencial	166
Figura 42 - Resíduos e Simulação do Ensaio Ma-17 em fase exponencial	167
Figura 43 - Fator para o cálculo da velocidade específica de crescimento celular segunda a quantidade inicial de glicose	170
Figura 44 - Fator para o cálculo da velocidade específica de crescimento celular segunda a quantidade inicial de oxigênio dissolvido	171
Figura 45 - Fator para o cálculo da velocidade específica de crescimento celular segunda a quantidade inicial de glutamina	171
Figura 46 - Cálculo da GPV_e para condição $GLC \leq 32$ mM e $GLN \leq 7$ mM	173
Figura 47 - Cálculo da GPV_e para condição $GLC > 32$ mM e $GLN > 7$ mM	174
Figura 48 - Gráfico com os dados experimentais e simulados do ensaio Ma-07 e correspondentes resíduos	177
Figura 49 - Gráfico com os dados experimentais e simulados do ensaio Ma-08 e correspondentes resíduos	178

Figura 50 - Gráfico com os dados experimentais e simulados do ensaio Ma-09 e correspondentes resíduos	179
Figura 51 - Gráfico com os dados experimentais e simulados do ensaio Ma-10 e correspondentes resíduos	180
Figura 52 - Gráfico com os dados experimentais e simulados do ensaio Ma-11 e correspondentes resíduos	181
Figura 53 - Gráfico com os dados experimentais e simulados do ensaio Ma-12 e correspondentes resíduos	182
Figura 54 - Gráfico com os dados experimentais e simulados do ensaio Ma-14 e correspondentes resíduos	183
Figura 55 - Gráfico com os dados experimentais e simulados do ensaio Ma-15 e correspondentes resíduos	184
Figura 56 - Gráfico com os dados experimentais e simulados do ensaio Ma-16 e correspondentes resíduos	185
Figura 57 - Gráfico com os dados experimentais e simulados do ensaio Ma-17 e correspondentes resíduos	186
Figura 58 - Simulações do ensaio Ma-07 para as duas modelagens	188
Figura 59 - Simulações do ensaio Ma-08 para as duas modelagens.....	189
Figura 60 - Simulações do ensaio Ma-09 para as duas modelagens.....	190
Figura 61 - Simulações do ensaio Ma-10 para as duas modelagens.....	191
Figura 62 - Simulações do ensaio Ma-11 para as duas modelagens.....	192
Figura 63 - Simulações do ensaio Ma-12 para as duas modelagens.....	193
Figura 64 - Simulações do ensaio Ma-16 para as duas modelagens	194
Figura 65 - Velocidades das Vias Metabólicas da Faixa 1 ($GLC \geq 60$ mM)	197
Figura 66 - Velocidades das Vias Metabólicas da Faixa 2 ($6 \text{ mM} \leq GLC < 60 \text{ mM}$ e $GLN \geq 30 \text{ mM}$)	198
Figura 67 - Velocidades das Vias Metabólicas da Faixa 3 ($6 \text{ mM} \leq GLC < 60 \text{ mM}$ e $7 \text{ mM} \leq GLN < 30 \text{ mM}$)	199

Figura 68 - Velocidades das Vias Metabólicas da Faixa 4 ($6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $2,5 \text{ mM} \leq \text{GLN} < 7 \text{ mM}$)	200
Figura 69 - Velocidades das Vias Metabólicas da Faixa 5 ($6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $\text{GLN} < 2,5 \text{ mM}$)	201
Figura 70 - Velocidades das Vias Metabólicas da Faixa 6 ($\text{GLC} < 6 \text{ mM}$ e $\text{GLN} \leq 7 \text{ mM}$)	202
Figura 71 - Velocidades das Vias Metabólicas da Faixa 7 ($\text{GLC} < 6 \text{ mM}$ e $\text{GLN} < 7 \text{ mM}$)	203
Figura 72 - Gráfico com os todos os dados experimentais e simulados de crescimento celular e correspondentes coeficientes	204
Figura 73 - Gráfico com os todos os dados experimentais e simulados de glicose e correspondentes coeficientes	205
Figura 74 - Gráfico com os todos os dados experimentais e simulados de glutamina e correspondentes coeficientes	205
Figura 75 - Gráfico com os todos os dados experimentais e simulados de lactato e correspondentes coeficientes	206
Figura 76 - Gráfico com os todos os dados experimentais e simulados de amônio e correspondentes coeficientes	206
Figura 77 - Gráfico com os todos os dados experimentais e simulados de alanina e correspondentes coeficientes	207
Figura 78 - Gráfico com os todos os dados experimentais e simulados de glutamato e correspondentes coeficientes	207
Figura 79 - Gráfico com todos os dados experimentais e simulados de células viáveis, glicose e glutamina e correspondentes coeficientes	208
Figura 80 - Gráfico com todos os dados experimentais e simulados de todas as variáveis e correspondentes coeficientes	208
Figura 81 - Gráfico com a evolução dos resíduos ao longo do trabalho.....	210
Figura 82 - Simulação S2AcGPV com meio TC100AG, $\text{GLC} = 10 \text{ g/L}$, $\text{GLN} = 3,5 \text{ g/L}$ e $\text{DO} = 30\%$	213

Figura 83 - Simulação S2AcGPV com meio SF900 II e DO = 40%	214
Figura 84 - Simulação S2AcGPV com meio SF900 II e DO = 50%	215
Figura 85 - Simulação S2AcGPV com meio SF900 II, DO = 50% e temperatura = 28 – 22°C	216
Figura 86 - Simulação S2AcGPV com meio SF900 II, DO = 50% e temperatura = 28°C	217
Figura 87 - Simulação S2MtGPV-H com meio SF900 II, DO = 50% e temperatura = 28°C	218
Figura 88 - Simulação S2MtGPV-H com meio SF900 II, DO = 50% e temperatura = 28 – 22°C	219
Figura 89 - Simulação S2MtGPV-H com meio SF900 II, DO = 50 - 10% e temperatura = 28°C	220
Figura 90 - Simulação S2MtGPV-H com meio SF900 II, DO = 50 - 10% e temperatura = 28°C	221
Figura 91 - Simulação S2AcGPV com meio IPL-41, DO = 40%	222
Figura 92- Simulação S2AcGPV com meio IPL-41, DO = 5%	223
Figura 93 - Simulação S2AcGPV com meio IPL-41, DO = 20%	224
Figura 94 - Simulação S2AcGPV com meio IPL-41, DO = 60%	225
Figura 95 - Proposição das vias metabólicas básicas para as células rS2	227
Figura 96 - Planilha Excel com as condições de ensaio	250
Figura 97 - Planilha Excel com os dados experimentais	251
Figura 98 - Caixa de Diálogo - Selecionar arquivo de dados	253
Figura 99 - Caixa de Diálogo - Efetuar Ajuste de Curvas	253
Figura 100 - Painel de Controle e Janela de Gráficos	254
Figura 101 - Ajuste da fase exponencial	255
Figura 102 - Ajuste da reta para a curva do $\ln$ e X_v	256
Figura 103 - Gráfico com X_v ativo e gráficos de GLC, GLN e LAC normalizados ..	257
Figura 104 - Acompanhando a gravação de dados	259

Figura 105 - Gráficos de X_v e μ_x	259
Figura 106 - Conversão $Y_{LAC/X}$	260
Figura 107 - Acompanhando a gravação de dados de conversão	261
Figura 108 - Dados alisados de cada ensaio	264
Figura 109 - Dados experimentais de cada ensaio	264
Figura 110 - Informações de ajuste	264
Figura 111 - Equações cinéticas	265
Figura 112 - Escolha do método de otimização	268
Figura 113 - Informações de cada iteração	268
Figura 114 - Resultados do processamento	269
Figura 115 - Gráficos com pontos experimentais e pontos calculados pelo modelo ajustado	270
Figura 116 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando <i>fmincon</i> – <i>interior point</i>	272
Figura 117 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando <i>fmincon</i> – <i>sqp</i>	273
Figura 118 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando <i>fminsearch</i>	274
Figura 119 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando <i>pamboukian</i>	275
Figura 120 - Gráfico gerado pelo AjustaPar com pontos dos dados experimentais (*) e com dados resultantes da modelagem (linha), com <i>fminunc</i>	276
Figura 121 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 1	277

Figura 122 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 2	278
Figura 123 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 3	279
Figura 124 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 4	280
Figura 125 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 5	281
Figura 126 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 6	282
Figura 127 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 7	283
Figura 128 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 8	284
Figura 129 - Dados iniciais do ensaio	286
Figura 130 - Dados experimentais do ensaio (opcional)	287
Figura 131 - Arquivo Excel com as informações do ensaio	288
Figura 132 - Comparação com dados experimentais	288
Figura 133 - Resultados da simulação	289
Figura 134 - Gráficos com pontos experimentais e calculados pelo simulador	289
Figura 135 - Planilha Excel com as condições de ensaio	356
Figura 136 - Planilha Excel com os dados experimentais	356

Figura 137 - Dados alisados de cada ensaio	369
Figura 138 - Dados experimentais de cada ensaio	369
Figura 139 - Informações de ajuste	370
Figura 140 - Equações cinéticas	370

LISTA DE TABELAS

Tabela 1 - Fórmulas para o cálculo de resíduos	86
Tabela 2 - Atividades desenvolvidas neste trabalho	99
Tabela 3 - Ensaios para a modelagem	103
Tabela 4 - Equações cinéticas e parâmetros originais	109
Tabela 5 - Diferenças entre os programas AjuPar e AjustaPar	118
Tabela 6 - Resultados do estudo com cada método de minimização do AjustaPar	120
Tabela 7 - Resultados do estudo com cada fórmula do cálculo do resíduo da Tabela 1	121
Tabela 8 - Ensaios válidos para este trabalho	122
Tabela 9 - Sequência de modelos.....	123
Tabela 10 - Máximas velocidades específicas de consumo de substratos e formação de produtos dos ensaios para modelagem na fase exponencial, calculados através do TrataDados	129
Tabela 11 - Valores dos parâmetros a_i para cada ensaio no Modelo 1	130
Tabela 12 - Valores dos parâmetros a_i para cada ensaio no Modelo 2	135
Tabela 13 - Proposições metabólicas estudados no Modelo 3	139
Tabela 14 - Apresentação das vias metabólicas, numeração e enzimas do Modelo 4	147
Tabela 15 - Modelos metabólicos - Modelos 4A (com GOGAT)	150
Tabela 16 - Modelos metabólicos - Modelos 4B (sem GOGAT).....	151
Tabela 17 - Etapas da evolução do trabalho	167
Tabela 18 - Comparação entre os valores obtidos da velocidade específica máxima de crescimento celular	172
Tabela 19 - Divisão por faixas de concentração de substrato	172

Tabela 20 - Valores das velocidades específicas dos modos elementares (a) para as diferentes faixas (Tabela 18) [$1E-15$ / (cel*h)]	175
Tabela 21 - Valores das constantes de Michaelis por faixa K_i = [mM]	176
Tabela 22 - Fator multiplicador da máxima velocidade específica de crescimento, baseado na construção S2AcGPV em meio TC-100AG	211
Tabela 23 - Comparação entre os valores obtidos da velocidade específica máxima de crescimento celular	212
Tabela 24 - Resultados do estudo com cada método de minimização	276
Tabela 25 - Resultados do estudo com cada fórmula do cálculo do resíduo da Tabela 1	284

LISTA DE SIGLAS E SIMBOLOS

"a"	velocidade dos modos elementares
Δ	variação
[S]	concentração de substrato
μ_x	velocidade específica de crescimento celular (dia^{-1})
$\mu_{x\text{max}}$ ou μ_m	velocidade específica máxima de crescimento celular (dia^{-1})
13PG	1,3-fosfoglicerato
2PG	2-fosfoglicerato
3PG	3-fosfoglicerato
6PG	6-fosfogluconato
6PGL	6-fosfogliconatulactose
AC	aconitato hidratase
Ac	actina
AcCoA	acetil co-enzima A
ACT	aspartato carbamoiltransferase
ADP	adesonina difosfato
AFM	Análise de Fluxos Metabólicos
AKG	alfacetoglutarato
AKGDH	alfacetoglutarato dehidrogenase
ALA	alanina
AlaTA	alnina aminotransferase
APRT	aminodofosforibosiltransferase
ASP	aspartato
AspTA	aspartato aminotransferase
ATP	adesonina trifosfato

BHK	Baby Hamster Kidney
C	concentração de oxigênio dissolvido
CA	coeficiente angular
C _{crit}	concentração de oxigênio crítica
cel	células
CIT	citrato
CO ₂	gás carbônico
CoA	coenzima A
CS	citrato sintase
DNA	ácido desoxirribonucleico
E	enzima
EDO	Equação Diferencial Ordinária
EGFP	<i>Enhanced Green Fluorescent Protein</i>
ELISA	<i>Enzyme Linked Immunosorbent Assay</i>
ELISA	Enzyme-Linked Immuno Sorbent Assay
EMm	enzima málica mitocondrial
ENO	enolase
ES	complexo enzima-substrato
F16P	frutose 1,6-fosfato
F6P	frutose 6-fosfato
FAD	flavina adenina dinucleotídeo
FDPA	frutose difosfato aldolase
FuDH	fumarato dehidrogenase
FUM	fumarato
G	glutaminase
g	grama

G3F	gliceraldeído-3-fosfato
G3P	gliceraldeído 3-fosfato
G3PDH	gliceraldeído 3-fosfato desidrogenase
GDH	glutamato desidrogenase
GLC	glicose
GLN	glutamina
GLU	glutamato
GOGAT	glutamina 2-oxoglutarato amido transferase
GPV	glicoproteína do vírus da raiva
GPV _e	concentração de GPV específica (por célula)
GTP	guanosina trifosfato
h	hora
HBsAg	antígeno de superfície do vírus da hepatite B
HPLC	cromatografia líquida de alta resolução
ICB	Instituto de Ciências Biológicas
ICDH	isocitrato desidrogenase
ICIT	isocitrato
IPT	Instituto de Pesquisas Tecnológicas
K ₂ ou K _{cat}	valor máximo de reações enzimáticas catalisadas por segundo
KEGG	Kyoto Encyclopedia of Genes and Genomes
K _m ou K	constante de Michaelis-Menten
K _p	relação da purina total com a concentração celular
K _{pg}	relação da purina genômica com a purina total
L	litro
LAC	lactato
LDH	lactato desidrogenase

MAL	malato
MATLAB	Matrix Laboratory
MDH	malato desidrogenase
mg	miligrama
mM	milimolar
mol	$\text{mol} = 6,02 \cdot 10^{23}$ moléculas
Mt	metalotioneína
NADH	nicotinamida adenina dinucleotídeo
NADPH	nicotinamida adenina dinucleotídeo fosfato
NH_4^+	amônio
O_2	gás oxigênio
OD	oxigênio dissolvido
OXA	oxaloacetato
P	produto
pAc/GPV	promotor actina controla a expressão de GPV
PC	piruvato carboxilase
PDH	piruvato desidrogenase
PEP	fosfoenolpiruvato
PEPCK	fosfoenolpiruvato carboxiquinase
PFK	fosfofrutoquinase
P_g	quantidade de purina no genoma
PGDH	6-fosfogliconato desidrogenase
PGI	fosfoglucoisomerase
PGK	fosfoglicerato quinase
PGM1	fosfoglucomutase
PGM2	fosfoglicerato mutase

pH	potencial hidrogeniônico
P _i	fosfato inorgânico
PIRI	pirimidina
pMt/GPV	promotor metalotioneína controla a expressão de GPV
P _T	quantidade de purina total
PURI	purina
PYK	piruvato quinase
PYR	piruvato
QP	programação quadrática
R ²	coeficiente de correlação
R5P	ribose 5-fosfato
RNA	ácido ribonucleico
rpm	rotações por minuto
rS2	linhagem Schneider 2 de <i>Drosophila melanogaster</i> recombinante
S2	linhagem Schneider 2 de <i>Drosophila melanogaster</i>
S2AcGPV	células de S2 transfectadas com os vetores pAcGPV e pCoHygro
S2MtGPV-H	células de S2 transfectadas com o vetor pMtEGFP
S3	linhagem Schneider 3 de <i>Drosophila melanogaster</i>
SCoA	succinil co-enzima A
SCoAS	succinil-CoA sintase
SDH	succinil dehidrogenase
Sf21	linhagem de <i>Spodoptera frugiperda</i>
Sf9	linhagem de <i>Spodoptera frugiperda</i>
Sf900 II	meio de cultura da Gibco.
SQP	programação quadrática sequencial
SUC	succinato

TA	aminotransferase
UFPE	Universidade Federal de Pernambuco
UFSCar	Universidade Federal de São Carlos
UNICAMP	Universidade Estadual de Campinas
USP	Universidade de São Paulo
UV	ultravioleta
$V_{\max}$	velocidade máxima da reação
X_v ou X	células viáveis
$Y_{X/ATP}$	fator de conversão de ATP em células

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE SIGLAS E SÍMBOLOS

1 INTRODUÇÃO.....	32
2 OBJETIVOS.....	37
2.1 OBJETIVO GERAL	37
2.2 OBJETIVOS ESPECÍFICOS	37
3 REVISÃO BIBLIOGRÁFICA	39
3.1 CÉLULAS DE <i>Drosophila melanogaster</i>	39
3.2 GLICOPROTEÍNA DO VÍRUS DA RAIVA (GPV)	40
3.3 EXPRESSÃO DE GENES HETERÓLOGOS.....	41
3.4 SISTEMA DE REAÇÕES METABÓLICAS	44
3.4.1 Metabolismo celular	44
3.4.1.1 Metabolismo de carboidratos	46
3.4.1.2 Metabolismo de aminoácidos.....	49
3.4.1.3 Metabolismo do oxigênio.....	52
3.4.2 Células S2.....	53
3.5 TRATAMENTO DE DADOS.....	58
3.5.1 Alisamento de Curvas	60
3.5.1.1 Funções MATLAB para alisamento de curvas	61
3.5.1.2 Softwares para alisamento de curvas.....	62
• Lissage	62
• Emerson	63
• Excel.....	63
3.6 MATLAB	64

3.7 MODELAGEM MATEMÁTICA	65
3.7.1 Modelagem Matemática para células S2	68
3.7.2 Modelos não estruturados e estruturados	69
3.7.3 Equacionamento	70
3.7.3.1 Reações Metabólicas.....	70
3.7.3.2 Crescimento Biológico	71
3.7.4 Aspectos Biológicos das Vias Metabólicas.....	72
3.7.5 Modo Elementar	74
3.7.6 Análise de Fluxos Metabólicos.....	77
3.7.7 Metatool	78
3.7.8 Ajuste de Parâmetros.....	80
3.7.8.1 Ajuste do Modelo Matemático	80
3.7.8.2 Equações Diferenciais Ordinárias	82
3.7.8.3 Função Objetivo.....	83
• Mínimos Quadrados.....	83
• Outras formas para cálculo de resíduos	85
• Normalização de valores para cálculo de resíduos	86
3.7.8.4 Otimização de busca.....	87
• Principais técnicas de otimização de busca	88
❖ Métodos de busca direta ou não derivativos	88
○ Método Simplex	89
○ Método Simplex Modificado ou Poliedros Flexíveis.....	89
❖ Métodos de busca indireta ou derivativos.....	90
○ Método de Gauss-Seidel ou Newton	90
○ Método do Gradiente ou <i>Steepest Descent</i>	90
○ Método de Marquardt.....	90
○ Método Quasi-Newton.....	90
○ Método <i>Line Search</i>	91
❖ Métodos de busca de direção viável	91
• Classificação dos Problemas de Otimização	92
• MATLAB <i>Optimization Toolbox</i>	92
❖ Função <i>fminunc</i>	93
❖ Função <i>fminsearch</i>	93

❖ Função fmincon	93
○ Algoritmo <i>Trust-Region-Reflective</i>	93
○ Algoritmo SQP e <i>Active-Set</i>	94
○ Algoritmo <i>Interior-Point</i>	94
❖ Outros métodos de otimização com restrição	94
3.7.9 Análise Estatística	95
4 MATERIAIS E MÉTODOS	98
4.1 ATIVIDADES DESENVOLVIDAS	98
4.2 FERRAMENTAS COMPUTACIONAIS UTILIZADAS	99
4.2.1 MATLAB	100
4.2.2 Metatool	100
4.2.3 KEGG	101
4.3 ORIGEM DOS DADOS EXPERIMENTAIS	101
4.3.1 Células	101
4.3.2 Sistema de Cultivo	102
4.3.3 Conjunto de ensaios para modelagem	102
4.3.4 Metodologias	103
4.3.4.1 Concentração celular e viabilidade	103
4.3.4.2 Dosagem de glicose	103
4.3.4.3 Dosagem de ácido láctico	104
4.3.4.4 Dosagem de aminoácidos	104
4.3.4.5 Dosagem de amônio	104
4.3.4.6 Dosagem de glicoproteína da raiva (GPV)	104
4.3.5 Demais ensaios para simulação do modelo	104
4.4 SEQUÊNCIA DE TRABALHO PARA A MODELAGEM	105
5 RESULTADOS E DISCUSSÃO	106
5.1 TrataDados	106
5.2 AjustaPar	107
5.2.1 Comparação entre dois Programas de Ajuste de Parâmetros	108
5.2.1.1 Comparativo 1	110
5.2.1.2 Comparativo 2	113

5.2.1.3 Comparativo 3.....	115
5.2.2 Estudo do comportamento do AjustaPar para a escolha do método de trabalho	119
5.2.2.1 Métodos de minimização ao AjustaPar	119
5.2.2.2 Fórmulas para o cálculo de resíduos na Função Objetivo	120
5.3 LEVANTAMENTO DE ENSAIOS COM AS LINHAGENS rS2	121
5.4 MODELAGEM DA LINHAGEM rS2	123
5.4.1 Modelo 1	123
5.4.2 Modelo 2	132
5.4.3 Modelo 3	137
5.4.4 Modelo 4	144
5.4.5 Modelo 5	152
5.4.5.1 Modos Elementares do Modelo 5.....	153
5.4.5.2 Equacionamento dos Modos Elementares do Modelo 5.....	154
5.4.5.3 Simulação e Cálculo de Resíduo	156
5.4.5.4 Novas Proposições.....	166
5.4.5.5 Cálculo da concentração inicial de purina no Modelo 5.....	169
5.4.5.6 Definição das faixas de concentração do Modelo 5.....	172
5.4.5.7 Cálculo da GPV _e no Modelo 5.....	173
5.4.5.8 Resultados obtidos para o Modelo 5.....	174
• Velocidades Específica dos Modos Elementares	174
• Constantes de Michaelis-Menten.....	175
5.5 SIMULAÇÃO DO COMPORTAMENTO CINÉTICO UTILIZANDO O MODELO 5	176
5.5.1 Gráficos Finais e Resíduos dos dez ensaios modelados	176
5.6 COMPARAÇÃO ENTRE MODELOS	189
5.7 VELOCIDADE DAS VIAS METABÓLICAS	195
5.7.1 Faixa 1.....	197
5.7.2 Faixa 2.....	198
5.7.3 Faixa 3.....	199
5.7.4 Faixa 4.....	200
5.7.5 Faixa 5.....	201

5.7.6 Faixa 6.....	202
5.7.8 Faixa 7.....	203
5.8 VALIDAÇÃO DO MODELO 5	204
5.9 MODELO APLICADO A OUTROS ENSAIOS DO GRUPO	211
5.10 MODELO METABÓLICO FINAL	227
6 CONCLUSÃO	228
7 TRABALHOS FUTUROS	230
REFERÊNCIAS	231
APÊNDICES	248
Apêndice A - Manual do <i>software</i> TrataDados	249
Apêndice B - Manual do <i>software</i> AjustaPar	263
Apêndice C - Estudo para a escolha dos métodos do AjustaPar	271
Apêndice D - Manual do <i>software</i> de Simulação	255
Apêndice E - Listagem do <i>software</i> TrataDados	290
Apêndice F - Listagem do <i>software</i> AjustPar	357
Apêndice G - Listagem genérica do <i>software</i> AjustaPar utilizado no Modelo 5	371
Apêndice H - Listagem do <i>software</i> de Simulação	392

1. INTRODUÇÃO

Diferentes sistemas de expressão gênica podem ser utilizados para a produção de proteínas recombinantes: bactérias, leveduras e também células animais, dentre elas as de insetos. A escolha do sistema de expressão depende da aplicação que a proteína recombinante terá, do grau de purificação requerido e das condições de produção da célula.

Culturas de células animais têm sido largamente utilizadas pelos laboratórios de investigação básica e indústrias para a obtenção de uma grande variedade de produtos destinados à investigação, diagnósticos, uso terapêutico e também controle biológico na agricultura. Um grande esforço vem sendo despendido para se desenvolver metodologias que possam permitir culturas celulares em alta densidade e que levem à produção de grandes quantidades de um determinado produto. Assim, tecnologias para a preparação de produtos virais, tais como antígenos e vacinas em culturas celulares, bem como proteínas com fins terapêuticos e diagnósticos vêm sendo desenvolvidas, há alguns anos, com bastante sucesso. São tecnologias que permitem uma produção em larga escala, com alto grau de controle e reprodutibilidade, além de permitir a produção de proteínas, com características qualitativas aceitáveis em relação a sua capacidade imunizante e que podem ser utilizadas na preparação de vacinas (ALVES; CARRONDO; CRUZ, 2008). Dentre as células animais que têm sido utilizadas para este fim, destacam-se as células de mamíferos. Contudo, o cultivo destas células exige meios de cultura complexos e dispendiosos. Assim, o emprego de células de insetos, que requerem meios de cultivo menos complexos, destacou-se nos últimos anos, uma vez que estas células são capazes de realizar modificações pós-traducionais na complexidade requerida (JARVIS, 1998) e produzir proteínas recombinantes de grande interesse na área farmacológica. As células de insetos apresentam também elevado crescimento celular (SONDERGAARD, 1996; VALLE *et al.*, 2001) e estabilidade apropriada quando submetidas a processos de transfecção (SANTOS, 2006). Revelam-se ainda, significativamente mais robustas quanto ao cultivo, principalmente em relação à manutenção, facilidade de adaptação na mudança de sistemas aderentes para suspensos e, principalmente, por apresentarem maior resistência a estresse

ambiental, como mudanças de pH e de osmolalidade (IKONOMOU; SCHNEIDER; AGATHOS, 2003).

Células de insetos embrionárias de dípteros como *Drosophila melanogaster* Schneider 2 (S2) estão sendo utilizadas com sucesso na expressão destas proteínas (DEML; WOLF; WAGNER, 1999; DEML *et al.*, 1999). O sistema *Drosophila* de expressão é um sistema não lítico que utiliza plasmídeos simples para a expressão estável de proteínas (LIM; CHA, 2006; CHA *et al.*, 2005; SHIN; CHA, 2003). Segundo YOKOMIZO *et al.* (2007), a expressão proteica que emprega este sistema ocorre em células viáveis. As células S2 são similares às Sf9, no entanto, atingem densidades celulares superiores às observadas para as células de lepidópteros (AGATHOS, 1991). A linhagem S2, em particular, tem sido utilizada para a expressão e o estudo de proteínas recombinantes como transferrina humana (LIM; CHA, 2006), interleucina-2 (CHA *et al.*, 2005), antígeno de superfície do vírus da hepatite B (DEML; WOLF; WAGNER, 1999; DEML *et al.*, 1999; BACHMANN *et al.*, 2004), eritropoietina humana (SHIN; CHA, 2003), neuroserpina (HILL *et al.*, 2001), moléculas de adesão celular, oncogenes e fatores transcricionais (LEE *et al.*, 2000), anticorpos (KIRKPATRICK *et al.* 1995), enzimas (GIBSON *et al.*, 1993; BANKS; HUA; ADANG, 2003) e inibidores tumorais (VALLE *et al.*, 2001; JEON *et al.*, 2003).

Destaca-se que devido ao crescente interesse neste sistema de expressão, alguns trabalhos foram realizados empregando células S2 para a produção da glicoproteína G do vírus da raiva (GPV). Nesses trabalhos, a produção desta glicoproteína, potencialmente útil na composição de vacinas virais de uso humano e veterinário, foi avaliada no âmbito da biologia molecular (YOKOMIZO *et al.*, 2007), assim como no estudo de variáveis de processo (SWIECH, 2007; PAMBOUKIAN *et al.*, 2008; AGUIAR, 2010), formulação de meio de cultivo (GALESI, 2008) e modelagem matemática (BARRAL, 2010), entre outros.

No presente trabalho utilizaram-se células S2 de *Drosophila melanogaster* nas quais sistemas de expressão foram estabelecidos com sucesso (YOKOMIZO *et al.*, 2007; LEMOS, 2009) para a expressão da glicoproteína do vírus da raiva (GPV). Além da aplicação da GPV recombinante produzida como vacina, a sua expressão em células S2 também poderá ter uma importante aplicação na produção de componente de kits diagnósticos. Potencialmente este sistema de expressão de GPV associa as vantagens observadas em células de mamíferos (expressão

constitutiva e indutiva) com as vantagens observadas em células de insetos (facilidade de cultivo e baixo custo).

A forma de se operar um processo para a obtenção do produto desejável precisa ser vista como um fator destacado para se assegurar a viabilidade comercial do produto (AUGUSTO; BARRAL; PICCOLI, 2008). Para isso é muito importante conhecer o comportamento de uma linhagem celular sob diferentes condições de cultivo. Esse conhecimento permite atuar sobre os cultivos que geram produtos comerciais extremamente importantes, estabelecendo uma base mais criteriosa para incrementar a produtividade e o rendimento desses cultivos. Assim, identificam-se as limitações e inibições do crescimento celular e da síntese de produtos, o que permite melhorar a formulação do meio, definir as condições de cultivo mais adequadas e, por meio do controle de processos, fazer com que elas sejam mantidas em condições ótimas. Todas essas melhorias permitem o aumento das velocidades de reações ou do maior controle destas e, conseqüentemente, acontecem incrementos na produtividade, na concentração final do produto e na conversão de substratos em produtos (AUGUSTO; BARRAL; PICCOLI, 2008).

A sistematização e a quantificação desse conhecimento por equações matemáticas é o que se denomina modelagem matemática e relaciona as velocidades de crescimento, de consumo de nutrientes e de síntese de produtos com as condições de cultivo. A resolução dessas equações permite prever quantitativamente o comportamento das células, ou seja, representar, através de equações matemáticas, os balanços de massa para cada componente no interior do biorreator, associada às complexas transformações bioquímicas que ocorrem no processo e às velocidades com que essas transformações se processam (BONOMI; SCHMIDELL, 2001). Portanto, a modelagem matemática é provavelmente a técnica mais eficiente para se atingir uma otimização das condições de cultivo do bioprocesso (AUGUSTO; BARRAL; PICCOLI, 2008).

O metabolismo de uma célula em estudo, interfere diretamente na modelagem, pois é a relação entre os nutrientes e os produtos que faz a engrenagem da modelagem ser ativada.

Assim, faz-se necessário, também, um estudo sobre a complexidade desse metabolismo celular, tentando entender de que forma as reações acontecem e como

podem variar essas reações segundo as variações ambientais. Quanto mais claro for o entendimento do metabolismo da linhagem em questão, mais ferramentas se obtém para o equacionamento correto na modelagem.

O metabolismo das células rS2 (Schneider 2) de *Drosophila melanogaster* transfectadas para a expressão da glicoproteína do vírus da raiva (GPV) ainda não é totalmente conhecido. Existem poucos estudos específicos sobre o metabolismo de células S2, sejam elas selvagens ou recombinantes. Por outro lado, já existe um trabalho de modelagem matemática dessa linhagem, implementado por Barral (2010), que descreve com equações cinéticas tradicionais o seu comportamento cinético. Porém, essa modelagem apresenta deficiências devido a alta flexibilidade metabólica apresentada por essa linhagem. Diante deste quadro, a modelagem matemática e metabólica aplicada neste trabalho se concentra em estudar *in silico* o metabolismo, visando o crescimento celular, o consumo e a produção de metabólitos.

Diante deste quadro, o presente trabalho se propôs a elaborar um modelo metabólico, a partir de um conjunto de vias metabólicas. Isso permitiu encontrar modos elementares pertinentes a esse conjunto através do programa Metatool e, em seguida, propor uma modelagem matemática usando o equacionamento desses modos elementares.

Todos os ensaios utilizados na modelagem foram executados por Marcelo Aguiar (AGUIAR, 2010 e AGUIAR *et al.*, 2009). Outros ensaios, executados por vários autores (BATISTA, 2007; PAMBOUKIAN *et al.*, 2008; VENTINI, 2010), foram utilizados para testar o modelo gerado, extrapolando-o para condições de meios de cultura e construção diferentes do modelado.

Este trabalho foi desenvolvido no âmbito de um projeto temático financiado pela FAPESP intitulado “Expressão de genes heterólogos em células de dípteros: Biologia Molecular e Engenharia de Processos”, que visou a expressão gênica da glicoproteína G do vírus da raiva (GPV) e do antígeno de superfície do vírus da hepatite B (HBsAg) em células de *Drosophila melanogaster* (S2) para a produção de vacinas e kits diagnósticos. Esse projeto envolveu o Instituto Butantan (Laboratório de Imunologia Viral), a USP (Departamento de Engenharia Química), o IPT (Laboratório de Biotecnologia Industrial), a UNICAMP (Departamento de Processos

Biotecnológicos) e a UFSCar (Departamento de Engenharia Química). Há estudos de biologia molecular, otimização de meios de cultura, otimização de biorreatores e modelagem matemática. A contribuição do presente trabalho foi na direção de aproveitar e sistematizar os dados gerados pelo grupo, para geração de uma modelagem matemática que interpretasse a flexibilidade metabólica dessa linhagem.

2. OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo deste trabalho foi elaborar um modelo metabólico e matemático fenomenológico, estruturado e não segregado, que represente o comportamento cinético das células S2 (Schneider 2) de *Drosophila melanogaster* transfectadas (rS2) para a expressão da glicoproteína do vírus da raiva em bioprocessos. Como essa linhagem apresenta uma flexibilidade metabólica já observada nos experimentos, o modelo implementado neste trabalho tem a direção de responder a essa flexibilidade.

Para isso, foi feita uma correlação entre o estudo das vias metabólicas da linhagem rS2, no que tange a direcionamento e velocidade das reações e presença de enzimas, e modelagem matemática. Foi também objetivo deste trabalho a elaboração e implementação de novas rotinas de programação em MATLAB para o Tratamento de Dados Experimentais, para o Ajuste de Parâmetros da modelagem matemática e ainda um programa para simulação do comportamento cinético das células S2 recombinantes.

2.2 OBJETIVOS ESPECÍFICOS

Este trabalho foi desenvolvido em seis etapas contemplando os seguintes objetivos específicos:

1. especificação, projeto, implementação e validação de um programa para tratamento de dados experimentais de cultivos celulares em MATLAB;
2. especificação, projeto, implementação e validação de um programa para ajuste de parâmetros em MATLAB;
3. proposição de vias metabólicas para a linhagem rS2 (modelagem metabólica), a fim de se estabelecer o metabolismo básico dessa linhagem interpretando sua flexibilidade metabólica, utilizando o programa Metatool;

4. proposição de modelagem matemática do comportamento cinético da linhagem rS2 baseada na modelagem metabólica;
5. simulação do comportamento cinético das células rS2 através da modelagem metabólica e matemática frente a diferentes condições nutricionais, comparado-a com os dados experimentais;
6. especificação, projeto, implementação e validação de um programa para simulação do comportamento cinético das células S2 recombinantes em MATLAB, baseado na modelagem desenvolvida.

3 REVISÃO BIBLIOGRÁFICA

3.1 CÉLULAS DE *Drosophila melanogaster*

A *Drosophila melanogaster* é um inseto díptero (dois pares de asas) pertencente ao gênero das moscas da fruta como mostra a Figura 1. Esta espécie é um dos animais mais utilizados em experiências genéticas, sendo um dos mais importantes organismos modelo da Biologia.

Figura 1 – *Drosophila melanogaster*, macho (esquerda) e fêmea (direita).

A *Drosophila melanogaster* começou a ser estudada em 1909 pelo americano Thomas Hunt Morgan e nesse estudo ele conseguiu demonstrar as teorias de Mendel. Essas células são ainda hoje estudadas e representam grande importância para a genética e para a bioquímica. Dos quase 300 genes identificados no genoma humano e responsáveis por doenças hereditárias, mais da metade têm genes homólogos no genoma da *Drosophila* (COELHO, 2002).

Cerca de 94 linhagens de células de drosófila são independentemente cultivadas, das quais 12 têm sido mais facilmente manipuladas. Para a expressão de genes heterólogos, duas linhagens de células Schneider têm sido bastante utilizadas, as linhagens S2 e S3 (CHERBAS; MOSS; CHERBAS, 1994). Essas células se caracterizam pela semi-aderência e são células derivadas de cultura primária de embriões em estágios tardios de desenvolvimento (SCHNEIDER, 1972) de *Drosophila melanogaster*.

Várias proteínas já foram expressas em células de *Drosophila melanogaster* (S2), como por exemplo, a dopamina β hidroxilase humana (LI *et al.*, 1996) e o antígeno de superfície do vírus da hepatite B (HBsAg) (DELM; WOLF; WAGNER, 1999). Assim as células de drosófila têm a capacidade de expressar genes sem alterar

funções e propriedades da proteína, enquanto sistemas envolvendo infecção viral podem comprometer vias secretoras da célula.

3.2 GLICOPROTEÍNA G DO VÍRUS DA RAIVA (GPV)

A vacina da raiva protege as pessoas que foram lambidas, mordidas ou arranhadas por um animal que tenha a raiva. A raiva é uma grave doença, pois pode matar as pessoas que não forem tratadas e ataca o sistema nervoso central. A doença causa confusão, problemas de respiração e ataques. Estes sinais podem demorar a se manifestar, até oito semanas após o contágio. Nesse ponto, já não há mais a possibilidade de cura. A vacina deve ser administrada o mais rápido possível em todas as pessoas que tiveram contato com o animal doente.

Hoje tem-se 2,5 bilhões de pessoas expostas ao risco de contágio em cerca de 100 países. Nos países da Europa e América do Norte, a raiva silvestre representa a principal preocupação, estando a urbana controlada e até mesmo erradicada (GERMANO, 1994). Nos países da América Latina, Ásia e África, a raiva urbana é responsável, anualmente, por 50 a 60 mil mortes humanas, a maioria de crianças. Na Índia, por exemplo, já se atingiram 20 mil casos de raiva por ano e, na China, 5 mil. Na América Latina esses números não assumem proporções tão graves. No Brasil, do total de casos, têm-se uma média de contágio de 76% só na região nordeste. Morcegos, cães e gatos são os animais que apresentam maior risco de contrair raiva. Bovinos, suínos, caprinos e equinos apresentam risco médio (SVS/MS, 2012). Em 2011 foram registrados dois casos de raiva humana no Maranhão e em 2012 já houve 3 casos registrados sendo nos estados do Ceará, Minas Gerais e Mato Grosso (PORTAL DA SAÚDE, 2012; SVS/MS, 2012).

A replicação do vírus da raiva inicia-se através da adesão da espícula trimérica viral glicoproteína G, portanto a imunização se dá através de anticorpos protetores para essa glicoproteína. Na Figura 2 pode-se ver o esquema do vírus da raiva e a glicoproteína G.

Figura 2 – Esquema do vírus da raiva (CCAUFES, 2006).

Na vacina de primeira geração é utilizado substrato animal para a obtenção de massa viral, como tecido nervoso animal adulto (carneiros, cabras e coelhos), ovos embrionados e cérebro de animal lactente (camundongos - Fuenzalida, coelhos e ratos) e pode apresentar-se na forma inativada ou atenuada (PÉREZ; PAOLAZZI, 1997). A vacina de segunda geração é produzida em culturas celulares, como culturas de células primárias de mamíferos, células diploides e heteroplóides, e também pode apresentar-se na forma inativada ou atenuada. O soro antirrábico também é importante, pois o tratamento pós-exposição mostra-se bastante efetivo na indução da proteção. Atualmente, produz-se no Brasil, a vacina antirrábica em células Vero, em meio livre de soro fetal bovino em todas as etapas (desde a cultura celular), o que reduz o risco de transmissão de zoonoses e príons.

Todas as vacinas contra a raiva hoje existentes no mundo, de uso humano, são inativadas, ou seja, não apresentam vírus vivos. Porém as vacinas de cultivo celular são as que apresentam menos riscos de efeitos colaterais.

3.3 EXPRESSÃO DE GENES HETERÓLOGOS

A tecnologia do DNA recombinante, surgida no início dos anos 70, permitiu a adição de fragmento de ácido nucleico em um hospedeiro e a expressão posterior pelo hospedeiro das proteínas codificadas pelo fragmento introduzido. Dentre suas múltiplas aplicações, a produção de vacinas atraiu o interesse de pesquisadores e, em relativamente pouco tempo, obteve-se diferentes sistemas de expressão de

vários genes heterólogos. Em geral, o gene de interesse é inserido em um plasmídeo e este é introduzido na célula, a qual se torna então, capaz de transcrever e assim, gerar as proteínas de interesse (SCHATZMAYR, 2001).

Atualmente, diversos vetores genéticos podem ser construídos, visando a produção em larga escala de proteínas ou o estudo do metabolismo celular, como a própria expressão gênica, envolvendo transcrição e tradução, a replicação de DNA e até mesmo mecanismos envolvidos na metagênese e reparo do DNA.

Os vetores empregados em células eucarióticas podem se integrar ao genoma celular. Esses vetores são constituídos de uma completa unidade de transcrição e um marcador genético (genes que codificam enzimas capazes de conferir resistência a antibióticos ou outras drogas) que permite a seleção de células portadoras do DNA inserido. Esses vetores genéticos podem ainda permitir expressão gênica constitutiva ou indutiva. A utilização de vetores indutíveis permite controlar o período de expressão de um determinado gene pela indução do promotor, como o do gene da metalotioneína (Mt). Já a utilização de vetores constitutivos, com o promotor actina (Ac), permite que a expressão seja contínua. Esses vetores, uma vez construídos, devem ser introduzidos em células para expressar o gene heterólogo. Células eucarióticas tem grande potencial biotecnológico, já que são capazes não só de traduzir as proteínas através da adição de seu respectivo gene, como também de processá-las, realizando alterações pós-traducionais necessárias ao seu bom funcionamento. Assim, o processamento autêntico de proteínas partindo de organismos mais complexos é encontrado em células eucarióticas devido à manutenção nestas de todas as atividades biológicas.

Células de insetos, como a *Spodoptera frugiperda* (Sf9) têm sido, pelas condições facilitadas de cultura, largamente usadas para a expressão de genes transportados por baculovírus. Este sistema mostra, no entanto, o inconveniente de se constituir num sistema de expressão gênica transiente e não permanente, além de exigir especificamente o sistema derivado de baculovírus (PFEIFER *et al.*, 1997; SUMMERS; SMITH, 1987).

Sistemas de expressão em células de drosófilas têm obtido sucesso em relação aos demais, produzindo um elevado nível de expressão das proteínas heterólogas (BRIGHTY *et al.*, 1991; BUCKINGHAM *et al.*, 1996). Isto se deve, provavelmente, ao

uso de eficientes promotores gênicos e ao elevado número de cópias de plasmídeos encontrados nas células transfectadas (MCCARROLL; KING, 1997). Com isso, estratégias de produção de proteínas têm sido desenvolvidas em células de drosófilas (BENTING *et al.*, 2000; BUCKINGHAM *et al.*, 1996; LEE *et al.*, 2000).

No grupo do qual este trabalho faz parte, os antígenos virais, foram conseguidos através da clonagem dos genes da glicoproteína G do vírus da raiva (GPV) em vetores de expressão e da construção de células de dípteros (S2 – *Drosophila melanogaster*) estavelmente transformadas que expressam esses genes de forma constitutiva e indutível (YOKOMIZO *et al.*, 2007). As linhagens estavelmente transfectadas foram geradas pela co-transfecção de vetores contendo o gene da proteína de interesse e um segundo vetor que codifica um gene de resistência para um antibiótico, que é usado para seleção das células transfectadas. Na construção indutível (pMt/GPV), o gene GPV (glicoproteína G do vírus da raiva) está sob controle de um promotor de metalotioneína (Mt) de expressão indutível com metal (por exemplo, sulfato de cobre) e, na construção constitutiva (pAc/GPV), o gene GPV está sob controle da actina (Ac) de expressão constitutiva. A GPV recombinante é encontrada principalmente em lisados celulares e em menor quantidade no sobrenadante.

A transfecção constitui na introdução de um vetor genético em uma célula hospedeira e as condições ideais de cada transfecção devem ser estabelecidas através da adaptação dos protocolos de transfecção já existentes (GUY, 1997).

Para facilitar a penetração do DNA no interior das células, sua transferência dentro do núcleo e possivelmente no genoma, vários métodos de transfecção, com uma variedade de protocolos, têm sido utilizados. Isso reflete a necessidade de aumentar a eficiência de transcrição de genes heterólogos nas células hospedeiras, já que após a transfecção, a porcentagem de células que expressam a proteína de interesse é geralmente baixa. Isso se deve ao fato de que na maioria das vezes poucas células recebem o gene heterólogo e ainda assim em pequena quantidade (GUY, 1997). É importante ressaltar que, após a transfecção, o DNA é integrado no genoma de poucas células (mesmo que em quantidades elevadas) e que a eficiência da expressão do gene heterólogo está ligada à região do genoma em que este gene se inseriu. Ou seja, o gene, mesmo que integrado ao genoma celular,

pode ficar inativo se localizado em regiões de heterocromatina, uma vez que esta integração constitui-se em um evento aleatório (GUY, 1997).

A expressão das proteínas heterólogas em células de drosófila pode variar de acordo com as propriedades estruturais e funcionais da proteína, tornando-se, na maioria das vezes, um sistema eficiente, capaz de permitir o estudo da expressão de genes heterólogos para a produção de imunológicos em larga escala, com possível redução de custos e risco. Para tal, faz-se necessário o estabelecimento de condições apropriadas de multiplicação celular e expressão gênica (SANTOS, 2006), assim como formulações adequadas de meio de cultura e melhores condições de cultivo em biorreatores.

3.4 SISTEMA DE REAÇÕES METABÓLICAS

Inicialmente, para a construção das equações de balanço de massa do processo e posteriormente na elaboração das equações cinéticas, que representam a influência das variáveis de estado nas suas velocidades de geração e de consumo, é fundamental identificar o sistema de reações metabólicas inerente ao processo em estudo. Por sistema de reações metabólicas entende-se o conjunto simplificado das reações químicas que ocorrem nas células, conectando os metabólitos aos produtos gerados (entre os quais está incluída a população celular).

3.4.1 Metabolismo celular

O conhecimento do metabolismo da célula em cultura é essencial para a formulação de um novo meio de cultura, para o planejamento de estratégias de alimentação em cultivos, para se efetivar uma modelagem matemática com maior rigor e para se obter altas produtividades tanto de células quanto de produto (IKONOMOU; SCHNEIDER; AGATHOS, 2003). O metabolismo é o conjunto de todas as transformações que ocorrem no interior de uma célula, de forma coordenada. Esta rede de vias opera através de uma série de reações catalisadas por diferentes

enzimas. Cada passo do metabolismo é responsável por uma alteração específica, geralmente a remoção, transferência ou adição de um determinado átomo ou grupo funcional.

Em células animais, o carboidrato mais usualmente consumido é a glicose, que é considerado o mais a mais importante para o crescimento celular (BÉRDAD; TOM; KAMEN, 1993; BHATIA *et al.*, 1997; MENDONÇA; PALOMARES; RAMÍREZ, 1999) e que forma lactato como principal metabólito (DREWS; PAALME; VILU, 1995). O lactato é acumulado pelas células de inseto em baixos níveis e pode ser produzido sob condições de estresse provocado pela limitação de oxigênio (IKONOMOU; SCHNEIDER; AGATHOS, 2003) ou, ainda, reconsumido para o suprimento de carbono ou energia ao metabolismo celular. A glutamina consumida pode ser convertida em glutamato e amônio, mas em alguns casos pode ser sintetizada pelas próprias células em quantidades capazes de suprir o crescimento celular (ÖHMAN *et al.*, 1996).

Em geral, as concentrações de glicose e de aminoácidos podem ser fatores limitantes do crescimento celular. A falta destes nutrientes no cultivo pode ocasionar a interrupção do crescimento ou até a morte celular. Os principais subprodutos do catabolismo, a amônio e lactato, por sua vez, podem ser fatores inibidores do crescimento.

Dentre os fatores que afetam o crescimento e a produção de proteínas recombinantes em células de inseto, a concentração de oxigênio é essencial e sua completa exaustão pode ser fatal ao crescimento (SCOTT, BLANCHARD; FERGUNSON, 1992). A diminuição da concentração de oxigênio dissolvido abaixo do nível crítico durante a replicação e infecção de células Sf9 resulta na produção ineficiente de proteínas recombinantes (GOTOH *et al.*, 2002).

Na Figura 3, têm-se as rotas metabólicas de consumo de glicose e glutamina e formação de lactato e amônio para o cultivo de células Sf9.

Figura 3 – Esquema das principais vias metabólicas para a Sf9 (Adaptado de ÖHMAN; LJUNGGREN; HAGGSTROM, 1995)

3.4.1.1 Metabolismo de carboidratos

As células animais, em geral, obtêm sua energia de reações de oxidação-redução, sendo a glicose e a glutamina as principais fontes para estas reações. O catabolismo da glicose permite a produção de duas coenzimas (ATP – adenosina trifosfato e NADH – nicotinamida adenina dinucleotídeo) as quais são essenciais para a manutenção da viabilidade celular. Estas coenzimas podem ser usadas para a manutenção, metabolismo e/ou para a síntese de produtos de interesse (WAGNER, 1997¹ apud AMABLE; BUTLER, 2008).

A membrana citoplasmática é impermeável a moléculas polares como a glicose, de forma que as células devem utilizar transportadores proteicos específicos, localizados na membrana plasmática, para incorporar glicose ao interior da célula (ALTAMIRANO; GÓDIA; CAIRÓ, 2008). Na maioria das células animais, o mecanismo de transporte de glicose é a difusão facilitada, a qual se caracteriza por ser saturável, bidirecional e controlada pelo gradiente de concentração através da

¹ WAGNER, R. Metabolic control of animal cell culture processes. In: HAUSER, H.; WAGNER, R. (Ed.). Mammalian cell biotechnology in protein production. Berlin: Water de Gruyter, 1999. p. 193-232.

membrana citoplasmática. Uma vez que a concentração de glicose livre no citosol é, em geral, muito baixa, em decorrência da sua rápida fosforilação pela enzima hexoquinase, o transporte ocorre em direção à célula. Alternativamente, a glicose pode entrar na célula através do transporte simporte (acoplado unidirecional) com sódio, baseado no gradiente de concentração desse íon através da membrana plasmática.

Além de sua importância na geração de energia, a glicose é um precursor extremamente versátil, capaz de suprir uma enorme quantidade de metabólitos intermediários para reações biossintéticas. Em células animais, pode ser armazenada (na forma de polissacarídeos como o glicogênio); oxidada a um composto com três carbonos (piruvato) através da glicólise para produzir ATP e metabólitos intermediários; ou oxidada através da via das pentoses para produzir ribose 5-fosfato para a síntese de ácidos nucleicos e para gerar NADPH (Nicotinamida adenina dinucleotídeo fosfato), o qual tem função redutora em processos biossintéticos (AMABLE; BUTLER, 2008).

Quando os níveis de glicose são suficientes, o metabolismo segue as vias glicolíticas e das pentoses para formar gliceraldeído-3-fosfato (G3F). A via glicolítica ocorre completamente no citosol, não requer oxigênio e produz uma pequena quantidade de moléculas de ATP, de NADH e de piruvato por molécula de glicose. A conversão de G3F a piruvato depende da disponibilidade de NAD citoplasmático oxidado e, portanto, é necessária uma constante oxidação de NADH para gerar NAD e manter a glicólise em funcionamento. O NADH pode ser metabolizado na cadeia respiratória ou pela enzima lactato desidrogenase (LDH), convertendo piruvato a ácido láctico que, em altos níveis, pode ser tóxico para as células (ALTAMIRANO; GÓDIA; CAIRÓ, 2008).

Na presença de oxigênio, o piruvato gerado na via glicolítica é transportado à mitocôndria, onde é oxidado a CO_2 e H_2O em uma série de reações de oxidação conhecida como ciclo de Krebs. O piruvato é transportado através do citosol para a matriz mitocondrial, onde reage com a coenzima A (CoA), formando CO_2 e o intermediário Acetil-CoA, o qual entra no ciclo de Krebs. Esta via não só é responsável pela completa combustão do piruvato, como também pela combustão completa de carboidratos, ácidos graxos e aminoácidos. Tem como principais funções: a) sintetizar diferentes compostos, os quais são precursores para o

anabolismo celular e; b) gerar a maior parte da energia metabólica (AMABLE; BUTLER, 2008).

O Ciclo de Krebs está vinculado diretamente à geração de energia metabólica na forma de ATP, mediante a cadeia de transporte de elétrons e a fosforilação oxidativa, ambas localizadas na membrana interna da mitocôndria (ALTAMIRANO; GÓDIA; CAIRÓ, 2008), além de fornecer intermediários para a biossíntese.

De modo geral, o fornecimento de altas concentrações de glicose (4-24 mM) ao sistema resulta no acúmulo de alanina e lactato (BERDARD; TOM; KAMEM, 1993; BENSLIMANE *et al.*, 2005). Porém, o padrão de formação de lactato também varia de acordo com a linhagem celular.

Vários trabalhos indicam que em limitação de glicose, as células de insetos podem passar a consumir sacarose (GRACE, 1966; WANG; KWONG; BENTLEY, 1993) reduzindo a velocidade de crescimento. Assim, os elevados valores de consumo de glicose, ou outros intermediários da via glicolítica, sugerem a presença de uma via glicolítica e de um ciclo de Krebs bastante ativos (BHATIA *et al.* 1997).

Células de Sf9 não acumulam quantidades significativas de lactato (FERRANCE; GOEL; ATAAL, 1993; BÉRDARD; TOM; KAMEM, 1993; ÖHMAN; LJUNGGREN; HAGGSTROM, 1995; DOVERSKOG *et al.*, 2000; DREWS *et al.*, 2000), enquanto células de BM5 acumulam lactato até níveis inibitórios (STRAVOULAKIS; KALOGERAKIS; BEHIE, 1991; RHIEL; MITCHELL-LOEGEAN; MURHAMMER, 1997; BENSLIMANE *et al.*, 2005; SUGIURA; AMANN, 1996).

Há indicações de que a alanina pode ser consumida à medida que a concentração de glicose atinge valores limitantes (BÉRDARD; TOM; KAMEM, 1993).

Nas células Sf9 o excesso de amônio é excretado na forma de alanina, um subproduto não tóxico para as células (ÖHMAN *et al.*, 1996).

Neermann e Wagner (1996) compararam os fluxos específicos nas vias do metabolismo de glicose em células de inseto (Sf21) e células de mamíferos (BHK). Foi observado que o fluxo de glicose pela via glicolítica em ambas as células foi de aproximadamente 73% da glicose total utilizada pelas células. Entretanto, os fluxos de piruvato e de metabólitos da glicose no Ciclo de Krebs representam 60% da glicose consumida nas células de inseto, enquanto que nas de mamífero este número é de apenas 0,36%. Este resultado mostra a maior diferença entre os dois

metabolismos. A capacidade de produção de energia das células de inseto supera amplamente a de células de mamífero, considerando a produção de 36 mols de ATP produzidos por mol de glicose no Ciclo de Krebs, comparados aos 2 mols de ATP por mol de glicose produzidos na via glicolítica (NEERMANN; WAGNER, 1996).

Esta diferença entre as quantidades de glicose metabolizadas no Ciclo de Krebs pelas células de inseto e de mamíferos se deve à ausência de atividade das enzimas fosfoenolpiruvato carboxiquinase (PEPCK), piruvato carboxilase (PC) e piruvato desidrogenase (PDH) nas células BHK. Isto explica, em parte, o porquê de células de mamíferos produzirem quantidades significativamente maiores de lactato que células de inseto (NEERMANN; WAGNER, 1996).

O fluxo de glicose pela via das pentoses, por sua vez, é semelhante para ambas as linhagens celulares, o que indica que ambas precisam regenerar a mesma quantidade de NADPH ou requerem quantidades de ribose-5-fosfato semelhantes como precursores de ácidos nucleicos (NEERMANN; WAGNER, 1996).

3.4.1.2 Metabolismo de aminoácidos

Células eucarióticas superiores perderam, durante sua evolução, a capacidade de sintetizar uma série de aminoácidos. Esses aminoácidos são geralmente chamados de aminoácidos essenciais, enquanto aqueles que podem ser sintetizados são chamados de não essenciais. Essa nomenclatura pode ser considerada equivocada por duas razões. Primeiramente, alguns dos aminoácidos não essenciais, são na verdade, necessários para síntese de nucleotídeos (glicina, aspartato e glutamina). Segundo, a capacidade de síntese pode ser condicional, dependendo da situação nutricional, do estado proliferativo e das propriedades específicas das células (IKONOMOU; SCHNEIDER; AGATHOS, 2003).

Células de inseto utilizam aminoácidos para biossíntese e energia. Aminoácidos como glutamina, glutamato, aspartato, serina, arginina e metionina são utilizados para produção de energia (IKONOMOU; SCHNEIDER; AGATHOS, 2003). De acordo com Drews; Paalme e Vilu (1995), aminoácidos como histidina, lisina, treonina, glicina, valina, leucina, fenilalanina, tirosina, triptofano e isoleucina consumidos durante a fase de crescimento celular são incorporados em proteínas celulares. A oxidação destes aminoácidos, balanceados para produção de energia, é

insignificante. Glutamina e ácido aspártico (indiretamente também glutamato, asparagina e serina) podem ser utilizados para síntese de ácidos nucleicos. Serina pode ser utilizada para síntese de glicina e metionina para síntese de cisteína (DREWS; PAALME; VILU, 1995). A cisteína desempenha um papel indireto fundamental na proteção das células do estresse oxidativo e está presente, juntamente com a cistina, nas proteínas, como resíduos de aminoácidos. Resíduos da proteína de cisteína são formados pela oxidação de sulfidrilos de cisteína depois de terem sido incorporados. A formação e ruptura das pontes dissulfeto inter e intra proteína desempenham papéis importantes na determinação das estruturas terciárias.

Além de necessários para a constituição de proteínas e o crescimento celular, os aminoácidos também são importantes para manter o tamponamento e a osmolalidade. O tipo de aminoácido utilizado pelas células e sua eficiência de consumo dependem de cada linhagem celular (MITSUHASHI, 1976; FERRANCE; GOEL; ATAAL, 1993).

Ferrance; Goel e Atai (1993) demonstraram que os valores do fator de conversão de ATP em células ($Y_{X/ATP}$) aumentam à medida que se utiliza uma formulação de meio de cultura melhor balanceada em termos de aminoácidos. Ou seja, existe um melhor aproveitamento de nutrientes à medida que se diminui o seu excesso, aprimorando o acoplamento entre a produção e o consumo de ATP.

Além do importante papel como fonte de energia, a glutamina é um importante bloco estrutural para o metabolismo celular, sendo requerida como fonte de nitrogênio na síntese de pirimidinas e purinas, açúcares aminados, NAD e asparagina. A glutamina é incorporada às células mediante diferentes sistemas de transporte de aminoácidos, que não são específicos para a glutamina, mas que participam da incorporação de uma série de aminoácidos (ALTAMIRANO; GÓDIA; CAIRÓ, 2008).

Em células animais, o metabolismo da glutamina ocorre pela glutaminólise, começando pela conversão de glutamina a glutamato, através da liberação de um grupo amônio. O glutamato pode então seguir duas vias: via glutamato desidrogenase (GDH) ou via aminotransferases (TA). Na primeira, o glutamato é desaminado a α -cetoglutarato, liberando o segundo grupo amônio da glutamina como amônia. O α -cetoglutarato é incorporado ao ciclo de Krebs e pode gerar os metabólitos intermediários subsequentes do ciclo. A glutamina metabolizada via

GDH pode ser canalizada de duas formas diferentes, sendo completamente oxidada a CO₂, através da enzima málica mitocondrial (EMm), gerando até 27 mols de ATP por mol de glutamina consumida. Alternativamente, parte do esqueleto carbônico da glutamina sai da mitocôndria na forma de malato para ser, posteriormente, convertido a piruvato. Esta via é comumente conhecida como *malate shunt* (ALTAMIRANO; GÓDIA; CAIRÓ, 2008).

Na segunda via, a via TA, o glutamato também é convertido a α -cetoglutarato. Entretanto, o amônio liberado é estequiometricamente transferido ao oxaloacetato para formar aspartato (via aspartato aminotransferase, AspTA), ou a piruvato, para formar alanina (via alanina aminotransferase, AlaTA). Estas vias TA geram 9 moles de ATP por mol de glutamina e podem ser utilizadas como um mecanismo interno para balancear a produção excessiva de amônio (ALTAMIRANO; GÓDIA; CAIRÓ, 2008).

Do ponto de vista energético, a via GDH é mais eficiente que as vias TA, sendo usualmente favorecida quando o metabolismo da glutamina deve ser mais eficiente como, por exemplo, no caso de esgotamento de glicose no meio de cultivo. Observa-se que, em diferentes linhagens celulares, a célula ajusta os fluxos através das vias GDH, AspTA ou AlaTA dependendo das condições específicas de cultivo (MARTINELLE *et al.*, 1998; STREET *et al.*, 1993).

Drews *et al.* (2000) mostraram que, em células Sf9, o metabolismo de glutamina pode depender das condições de cultivo, seguindo mais de uma via metabólica. Em situações de limitação de glicose, a glutamina é consumida normalmente pela glutaminólise, gerando amônio como principal subproduto. Entretanto, quando há excesso de glicose, a glutamina é metabolizada por intermédio da glutamato sintase (GOGAT – glutamina 2-oxoglutarato amido transferase), de forma que o grupo amida da glutamina é transferido para a posição amina do glutamato que, posteriormente, é transaminado e, em vez de amônio, forma alanina, o principal subproduto desta via metabólica. Recentemente, Bernal *et al.* (2009) propuseram um modelo metabólico para a linhagem Sf9, considerando a existência da GOGAT.

Originalmente, a GOGAT é uma enzima que faz parte do sistema de assimilação de nitrogênio em bactérias, leveduras, fungos e plantas. Seshachalam; Subramanyam e Krishnamoorthy (1992) foram os primeiros a descobrir a glutamato sintase em um inseto, o bicho-da-seda (*Bombyx mori*). Sua presença em células Sf9 indica que esta

enzima também pode ser encontrada em outros insetos (DREWS *et al.*, 2000). Entretanto, deve-se notar que linhagens como *Trichoplusia ni* produzem amônio em situações de excesso de glicose (RHIEL; MITCHELL-LOEGEAN; MURHAMMER, 1997; BENSLIMANE *et al.*, 2005) indicando que outra via metabólica pode estar sendo utilizada.

3.4.1.3 Metabolismo do Oxigênio

No metabolismo celular, o oxigênio é o último elemento a receber elétrons na cadeia respiratória, sendo reduzido a água e permitindo que ocorra a re-oxidação das coenzimas que participam das reações de desidrogenação ao longo da glicólise e do ciclo de Krebs e a estocagem de energia através da transformação das moléculas de ADP em ATP, fundamentais para o funcionamento da máquina celular (manutenção, crescimento, síntese de produtos, transporte ativo entre outros). Para se obter elevadas velocidades de crescimento celular necessita-se de altas velocidades de consumo de fontes de carbono. Para que isso ocorra é necessário ter-se abundância de elétrons na cadeia respiratória para que a geração de ATP também ocorra e conseqüentemente, será necessário ter-se quantidade suficiente de oxigênio dissolvido no meio para que estes elétrons possam ser drenados ao final da cadeia (BAILEY; OLLIS, 1986).

O oxigênio deve, portanto, ser considerado um nutriente semelhante a outros dissolvidos no meio de cultura tais como glicose e aminoácidos. Entretanto, o oxigênio apresenta como limitação sua baixa solubilidade (inferior a 7 mg/L), quando comparada a de outros nutrientes, como a glicose, que podem ser mantidos em concentrações acima de 10 g/L (BAILEY; OLLIS, 1986; PAMBOUKIAN *et al.*, 2008). Estudos mostram que condições de ausência de oxigênio podem ter um impacto severo no crescimento celular. Concentrações muito altas de oxigênio dissolvido, por outro lado, acarretam danos oxidativos. De modo geral, uma faixa de oxigênio dissolvido entre 30% e 60% da saturação com ar é aceita como ótima para o crescimento celular, apesar de haver uma certa variação na sensibilidade de diferentes linhagens celulares (AMABLE; BUTLER, 2008). Por essa razão, torna-se necessário suprir continuamente o meio de cultura com oxigênio para assegurar a

necessidade da demanda metabólica da célula (BAILEY; OLLIS, 1986; PAMBOUKIAN *et al.*, 2008).

Ozturk e Palsson (1990) cultivaram células WI-38 em diferentes condições de oxigênio dissolvido e perceberam que, com esta variável a uma concentração de 5%, a velocidade específica de crescimento tem uma queda acentuada e o fator de conversão de glicose a lactato aumenta de 4 a 6 vezes.

Dentre os fatores que afetam a produção de proteínas recombinantes por células de inseto, a concentração de oxigênio dissolvido é conhecida como decisiva e sua completa exaustão pode ser fatal ao cultivo (SCOTT; BLANCHARD; FERGUNSON, 1992). A diminuição da concentração de oxigênio dissolvido abaixo do nível crítico durante a replicação e infecção de células Sf9 resulta na produção ineficiente de proteínas recombinantes (GOTOH *et al.*, 2002). No entanto, é difícil suprir oxigênio para atender a alta demanda requerida por células respirando ativamente, especialmente em culturas em altas densidades e em larga escala, devido a uma série de limitações: o oxigênio é pouco solúvel em água e, além disso, a transferência da fase gasosa para o meio é baixa quando comparada ao consumo de oxigênio pelas células, uma vez que o cultivo não pode ser muito agitado (100-250 rpm) devido à morte celular decorrente de altas tensões de cisalhamento. Assim, aerações brandas são necessárias para o cultivo de células de inseto que são muito frágeis e danificam-se facilmente pelas tensões de cisalhamento associadas à agitação mecânica e ruptura de bolhas (KUNAS; PAPOUTSAKIS, 1990; TRAMPER; VLAK; GOOIJER, 1996).

3.4.2 Células S2

A maioria dos estudos existentes sobre vias metabólicas para células de inseto refere-se a linhagens de lepidópteros, como a Sf9 de *Spodoptera frugiperda* (FERRANCE; GOEL; ATAAI, 1993; DREWS; PAALME; VILU, 1995; OHMAN; LJUNGGREN; HAGGSTROM, 1996; DOVERSKOG; HAN; HAGGSTROM, 1998; DREWS *et al.*, 2000; DOVERSKOG *et al.*, 2000; BENSLIMANE *et al.*, 2005; BERNAL *et al.*, 2009).

As informações disponíveis na literatura sobre o metabolismo de células S2 são, em sua grande maioria, específicas para a linhagem de células transformadas para expressão de glicoproteína da raiva (AGUIAR *et al.*, 2009; BATISTA *et al.*, 2009; BOVO *et al.*, 2008; GALESI *et al.*, 2008; PAMBOUKIAN *et al.*, 2008; SWIECH *et al.*, 2008a e b).

Vale à pena ressaltar que existem poucos estudos específicos sobre o metabolismo de células S2, sejam elas selvagens ou recombinantes. Como o genoma da *Drosophila melanogaster* já foi mapeado, já foram identificadas as principais enzimas que atuam nos processos metabólicos em geral e estão a disposição no KEGG (*Kyoto Encyclopedia of Genes and Genomes*). Porém, esse metabolismo representa o potencial de um organismo, mas é necessário saber em que momento da vida do organismo cada gene estará ativo, e consequentemente, qual enzima e que via metabólica estará ativa.

A linhagem S2 é constituída de células derivadas de cultura primária de embriões de *Drosophila melanogaster* em estágios tardios de desenvolvimento (SCHNEIDER, 1972). Por isso, não se pode fazer uma relação direta entre o metabolismo de um organismo adulto e uma célula embrionária. Além disso, de uma maneira geral, quando uma linhagem passa por adaptações ambientais para a sua imortalização, ela pode sofrer deleção e até inserção de genes, dependendo da maneira como ela é imortalizada.

O metabolismo de carboidratos em células S2 é ainda desconhecido, porém estudos sugerem que a glicose seja a principal fonte de carbono e, provavelmente, a trealose seja um nutriente essencial para o crescimento desta linhagem celular (GUY, 1997). No meio SF900 II a trelose está presente, porém no meio TC100 modificado não, o que vai contra a afirmação de Guy (1997).

Estudos realizados por Galesi *et al.* (2008) mostram que concentrações iniciais de 10 g/L de glicose e de 3,5 g/L de glutamina não apresentaram limitações para células S2AcGPV2 de *Drosophila melanogaster* quando cultivadas em meio TC100 modificado. Foram observados, no entanto, indícios de inibição do crescimento quando a concentração de amônio atingia valores entre 40 e 60 mg/L.

Galesi *et al.* (2008) também sugeriram que a produção da glicoproteína da raiva é inibida quando a concentração de amônio chega a concentrações próximas de 50 mg/L.

Pamboukian *et al.* (2008) observaram que cultivos de S2 selvagem e da linhagem S2AcGPV em meio SF900 II não são limitados por glicose, porém observa-se a interrupção do crescimento exponencial quando a glutamina chega a níveis abaixo de 0,8 g/L.

Swiech *et al.* (2008a, b) observaram que a glicose não é limitante no cultivo de células de *Drosophila melanogaster* quando a condição inicial de cultivo é de 10 g/L. Entretanto, o crescimento é interrompido quando se observa esgotamento de prolina. A produção de GPV, por sua vez, é limitada pelos aminoácidos serina e cistina (SWIECH *et al.*, 2008a).

Quando a glutamina atinge concentrações abaixo de 300 mg/L, observa-se a interrupção do crescimento exponencial. Além disso, o esgotamento deste aminoácido resulta na diminuição da viabilidade celular e da concentração de células viáveis e provoca uma mudança metabólica na qual se observa o consumo de outros aminoácidos, dentre os quais a alanina, e o consequente aumento na produção de amônio (SWIECH *et al.*, 2008a). Swiech *et al.* (2008a) também sugerem que a linhagem S2AcGPV é resistente ao amônio por terem observado a produção de concentrações de 1,12 g/L deste subproduto. Esta tolerância a amônio também já havia sido observada no inseto *Drosophila melanogaster* (BORASH *et al.*, 2000).

Em ensaios realizados com a linhagem S2AcGPV, Batista *et al.* (2009) verificaram que o crescimento celular é limitado por glicose, quando o cultivo foi realizado com meio IPL-41 suplementado. Por outro lado, em cultivos com o meio SF900 II, o substrato limitante foi a glutamina.

Estudos do efeito da concentração do oxigênio dissolvido mostraram que concentrações de oxigênio dissolvido de 95% inibem tanto o crescimento celular quanto a produção de GPV (BATISTA *et al.*, 2009), sendo que os melhores resultados de ambos foram obtidos com concentrações de 5 e 40%.

Swiech (2007) mostraram que células S2 são extremamente bem adaptadas à hipoxia, mantendo alta viabilidade durante longos períodos. Esta característica, entretanto, não impede que o crescimento seja limitado por falta de oxigênio.

Bovo *et al.* (2008) realizaram ensaios cinéticos com as células S2 selvagens e observaram que o esgotamento da glicose aparentemente leva à morte celular, comportamento também observado por Sondergaard (1996). Entretanto, cultivos com altas concentrações de glicose mostram outras limitações que não por este substrato e/ou inibições ainda não identificadas. Bovo *et al.* (2008) também notaram que em meios nos quais as concentrações de glicose e de glutamina são relativamente baixas (abaixo de 1 g/L), ambos os substratos podem limitar o crescimento.

Essa variabilidade de respostas pode ser explicada por uma grande flexibilidade da linhagem S2 em se adaptar às condições ambientais, aparentemente alterando os fluxos metabólicos pelas principais vias de assimilação da glicose e da glutamina (BARRAL, 2010).

Com relação ao consumo de oxigênio, especificamente sobre células S2 transfectadas para a expressão da glicoproteína do vírus da raiva, a velocidade específica máxima de consumo de oxigênio gira em torno de 5×10^{-11} mmolO₂/cel.h (BATISTA *et al.*, 2009; PAMBOUKIAN *et al.*, 2008). Esses trabalhos mostram que a velocidade específica de respiração da célula S2 selvagem é cerca de 5 vezes menor que a das células recombinantes. Comparando-se os resultados de S2 selvagens e transfectadas com as células Sf9, observa-se que as Sf9 têm uma velocidade específica de respiração maior que as demais (PAMBOUKIAN *et al.*, 2008). Nos casos de células Sf9 infectadas por vírus, estas costumam respirar mais intensamente que as não-infectadas (KIOUKIA *et al.*, 1995).

Quanto aos valores de menor concentração de oxigênio dissolvido que garantam velocidade específica de consumo de oxigênio máximo (C_{crit}) das várias linhagens S2 estabelecidas, temos: S2AcGPV em torno de 10%, S2MtEGFP em torno de 5% e S2AcHBsAgHy em torno de 12% (PAMBOUKIAN *et al.*, 2009).

Com relação à influência da concentração de oxigênio dissolvido (C) na expressão de proteínas heterólogas, segundo verificado por Batista *et al.* (2009) a maior expressão de GPV é conseguida com C controlado a 40% enquanto que com C em

5% tem-se uma redução de 18% desse valor. Nas demais condições estudadas (20%, 60% e 95%) esse valor cai para em média de 22%. Segundo Aguiar (2010), a síntese de GPV acompanha o crescimento, tendo valores máximos em cultivos com oxigênio dissolvido controlado em 30 %, em relação às outras condições testadas (5, 50 e 80 %).

Segundo Aguiar (2010), níveis de oxigênio superiores a 30% OD têm efeito inibitório no crescimento celular, indicado pela redução de 9 vezes no valor da velocidade específica de crescimento celular (μ_x) e de 20 vezes no valor de produção de células. Incrementos na disponibilidade de oxigênio causaram, ainda, alterações no metabolismo, diminuindo a eficiência no consumo de glicose e glutamina e gerando, assim, maiores quantidades de amônio e alanina por molécula de substrato consumido. Nos ensaios com variação de oxigênio, o maior valor de teor celular de GPV foi observado na condição de oxigênio dissolvido de 50%. Apesar disso, valores máximos de concentração e de produtividade em GPV foram obtidos para concentrações de oxigênio igual ou inferior a 30% (AGUIAR, 2010).

Uma característica muito importante que deve ser levada em consideração é a capacidade da *Drosophila melanogaster* de se recuperar da ausência de oxigênio sem sofrer dano algum. Swiech *et al.* (2008b) mostraram que células S2 são extremamente bem adaptadas à hipóxia, mantendo alta viabilidade durante longos períodos de tempo. Esta característica, entretanto, não impede que o crescimento seja limitado por falta de oxigênio. Algumas das razões para esta tolerância a baixos níveis de oxigênio são apresentadas a seguir:

- Hipóxia induz uma diminuição da permeabilidade da membrana, que reduz dramaticamente o custo de energia para o fluxo de íons (BOUTILIER, 2001).
- Estes organismos possuem uma capacidade não usual de entrar e retornar rapidamente de estados metabolicamente deprimidos utilizando metabolismo anaeróbico para sustentar as reduzidas trocas de energia durante hipóxia. Claramente, o efeito desta regulação metabólica de depressão é o que conserva o substrato para geração de energia, reduz formação de produtos finais deletérios e estende o tempo de vida do organismo (SUZUKI *et al.*, 2000 e ZHONG *et al.*, 2000).

- A habilidade de realocar a energia celular para demandas essenciais quando, por exemplo, os nutrientes são limitados. Outro exemplo, a economia de energia pela redução da síntese de proteínas é realocada para funções celulares mais críticas envolvendo homeostase osmótica e iônica. O aumento da regulação de proteínas *heat-shock* ocorre durante a limitação por oxigênio e foi demonstrado que esta situação origina uma variedade de tipos celulares resistentes a hipóxia *in vitro* (SUZUKI *et al.*, 2000 e ZHONG *et al.*, 2000).

Também se observou diferenças no metabolismo entre um inseto adulto e larva, principalmente quando se altera os níveis de oxigênio (FEALA *et al.*, 2009). Em níveis baixos de oxigênio (4% OD) a glicose é mais consumida (FEALA *et al.*, 2009). Zhou *et al.* (2008) obtiveram uma linhagem adaptada a hipóxia, após várias gerações em hipóxia. Nessa linhagem adaptada observaram a completa inatividade de várias enzimas do metabolismo principal.

3.5 TRATAMENTO DE DADOS EXPERIMENTAIS

Estudar o comportamento cinético de um sistema significa compreender como variam as concentrações dos componentes representativos desse sistema (célula, substratos, produtos e subprodutos) ao longo do tempo. Assim, a partir da análise da evolução dessas variáveis no tempo é possível calcular uma série de grandezas cinéticas que caracterizam o sistema e que é a base para a identificação dos fenômenos que o controlam (AUGUSTO; BARRAL; PICCOLI, 2008).

O tratamento dos dados experimentais é uma atividade imprescindível para o cálculo preciso das grandezas cinéticas. A qualidade da análise cinética, da identificação dos fenômenos relevantes e, posteriormente, do ajuste de parâmetros do modelo, são diretamente dependentes da precisão da informação inicial. A correta análise dos dados experimentais é a sustentação para o entendimento e compreensão dos fenômenos ocorridos, além da relevância para a construção dos modelos matemáticos.

A situação de quando ocorre mudança das variáveis de estado em função do tempo é o tratamento mais difícil e corresponde aos dados obtidos em processo

descontínuo, descontínuo alimentado ou fase transiente de um contínuo. No caso de um processo contínuo em estado estacionário, os valores das variáveis de estado e das velocidades não sofrem alteração com tempo, logo o cálculo das velocidades resulta da solução de uma equação algébrica (AUGUSTO; BARRAL; PICCOLI, 2008).

Velocidade específica de crescimento celular, velocidade específica de consumo de substrato e fator de conversão de substrato a célula são algumas das importantes grandezas cinéticas que devem ser encontradas de forma sistemática, precisa e rápida. Assim, também a determinação das fases (lag, exponencial e declínio) do crescimento celular tem grande importância no entendimento dos fenômenos que ocorrem.

O erro experimental é outro aspecto de extrema relevância na questão do tratamento de dados. A dimensão do erro analítico depende de vários fatores, incluindo tipo de metodologia adotada, a capacitação do operador, a padronização do preparo das células, entre outros. Os erros são relativamente elevados em sistemas biológicos: por exemplo, de 10 a 30% para a medida da concentração de proteínas; de 10 a 18% para a medida da concentração celular e da viabilidade, especialmente se a viabilidade for baixa; de 15 a 25% em medidas de cromatografia líquida (glicose, lactato e aminoácidos). Os erros nas medidas obtidas de processos descontínuos são normalmente maiores que os obtidos no estado estacionário de processos conduzidos em modo contínuo, porque no primeiro caso ocorrem variações nas condições do cultivo ao longo de todo o processo. É sempre possível proceder ao cálculo das grandezas cinéticas diretamente a partir dos dados experimentais. Entretanto, os resultados terão maior qualidade se os erros forem identificados e atenuados antes de se empregar os métodos de tratamento para obtenção das grandezas cinéticas. Essa atenuação pode ser feita por métodos de alisamento, que devem permitir extrapolar dados, visto que o número de pontos experimentais é, em geral, reduzido. Além disso, os métodos de alisamento devem também possibilitar o cálculo de velocidades nesses pontos (AUGUSTO; BARRAL; PICCOLI, 2008).

As principais técnicas de alisamento de dados experimentais são apresentadas a seguir:

- suavização manual dos pontos experimentais;
- ajuste de funções matemáticas simples, como regressão linear, logarítmica ou polinomial;
- ajuste de funções matemáticas por trechos, como o método *spline*;
- abordagem mista, que envolve dois ou mais dos itens anteriores simultaneamente.

Uma vez que os erros experimentais tenham sido atenuados, os dados estarão preparados para os cálculos das grandezas cinéticas. Com muita frequência, é necessário determinar as velocidades específicas, e as abordagens possíveis são:

- derivação das funções matemáticas ajustadas aos dados;
- cálculo da derivada por métodos geométricos, como o de Le Duy e Zajic (LE DUY; ZAJIC, 1973), que pode ser aplicado aos dados experimentais alisados. Esse método foi implementado em Excel por Gombert em Hiss (2001).

A determinação dessas velocidades possibilita o cálculo direto dos fatores de conversão instantâneos. Essa forma de calcular é empregada quando os valores dos fatores variam ao longo do cultivo. Alternativamente, é possível determiná-los a partir da regressão linear da concentração do produto em função da concentração de substrato (AUGUSTO; BARRAL; PICCOLI, 2008).

3.5.1 Alisamento de curvas

O levantamento de dados experimentais, em geral, fornece um conjunto de pontos relacionados que obedecem a uma determinada função matemática, muitas vezes desconhecida ou de difícil determinação. Matematicamente é possível fazer uma aproximação desta função com o objetivo de traçar uma curva contínua que represente o processo.

Em Zamboni e Monezzi Jr (2002) pode-se ver a descrição de diversas técnicas para aproximar uma função de fórmula desconhecida ou de difícil manipulação por outra função, chamada de função aproximante. As principais técnicas são:

- interpolação, onde utiliza-se pontos em comum entre as duas funções. Neste caso a função aproximante também é chamada de interpolante;
- regressão, onde utiliza-se o valor mínimo para a soma de quadrados ou soma de módulos das diferenças entre as imagens das duas funções em determinados pontos;
- integral mínima, onde utiliza-se o valor mínimo para a integral (área, volume etc.) do quadrado ou módulo da diferença entre as duas funções;
- derivadas iguais, onde utiliza-se as derivadas de várias ordens das duas funções em um mesmo ponto ou em pontos diferentes e determinados;
- *splines*, hastes flexíveis em português, onde utiliza-se o valor mínimo da energia de deformação para a função aproximante ou outra técnica de suavidade condizente;
- séries, aproximação utilizando uma soma de funções.

3.5.1.1 Funções MATLAB para alisamento de curvas

O MATLAB possui um *toolbox* específico para o ajuste de curvas, o *Curve Fitting Toolbox* (MATHWORKS, 2010a). Suas principais funções são:

- *spline* – para o ajuste de uma *spline* cúbica;
- *csaps* – para o ajuste de uma *smoothing spline*;
- *pchip* – para o ajuste de uma curva de Hermite.

3.5.1.2 Softwares para alisamento de curvas

Diversos softwares foram desenvolvidos com o objetivo de fazer o alisamento de dados experimentais. A seguir descrevemos alguns deles.

- **Lissage**

O programa *Lissage* foi desenvolvido em 1991 pelo laboratório do “*Institut National des Sciences Appliquées de Toulouse*” França, por Ardaillon-Simões, Arroyo e Uribe Larrea.

Este programa baseia-se no método *spline*, ajustando polinômios de grau 3 ou 4 entre dois pontos experimentais consecutivos. Em cada ponto, a derivada do polinômio ajustado no intervalo anterior deve ser igual à derivada do polinômio ajustado no novo intervalo. Segundo o autor Ardaillon-Simões, os coeficientes do polinômio são determinados mediante a minimização das distâncias entre os pontos calculados e os pontos experimentais que estão à direita do intervalo considerado. O coeficiente de Lissage n indica a quantidade de pontos à direita que serão levados em consideração nessa minimização. Por exemplo, se estamos determinando o 3º polinômio (entre o 3º e 4º pontos experimentais) e o coeficiente de Lissage escolhido pelo usuário for 4, isto significa que só entrarão na minimização dos quadrados quatro pontos experimentais à direita (o 4º, o 5º, o 6º e o 7º pontos). Porém, se o coeficiente fosse 5 entraria também o 8º ponto na determinação do polinômio. O objetivo disto é procurar “antecipar”, na determinação do polinômio relativo a um certo intervalo, o comportamento “futuro” da curva.

A Figura 4 mostra um exemplo de tela deste software. Os pontos fornecidos para o ajuste foram:

$$x = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9] \text{ e } y = [2 \ 4 \ 7 \ 5 \ 1 \ 3 \ 3 \ 7 \ 4 \ 0].$$

Figura 4 - Tela do software *Lissage*

É importante salientar que, nos sistemas operacionais Windows Vista e Windows 7 este aplicativo não funciona, pois não é possível abrir uma janela MS-DOS em tela cheia.

- **Emerson**

O software Emerson foi desenvolvido por Wagner Bruna do Instituto de Pesquisa Tecnológicas (IPT) e possui vários métodos para o ajuste de curvas: *spline*, *B-spline*, *smooth cubic spline* e outros.

Na Figura 5 podemos ver um exemplo que utiliza uma *smooth cubic spline* com coeficiente de alisamento 0,9. Os pontos fornecidos para o ajuste foram:

$x = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$ e $y = [2 \ 4 \ 7 \ 5 \ 1 \ 3 \ 3 \ 7 \ 4 \ 0]$.

Figura 5 - Tela do *software* Emerson

- **Excel**

O Excel também pode ser utilizado para o ajuste de curvas. A Figura 6 mostra um gráfico de dispersão gerado pelo ajuste dos valores

$x = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$ e $y = [2 \ 4 \ 7 \ 5 \ 1 \ 3 \ 3 \ 7 \ 4 \ 0]$.

Figura 6 - Tela do *software* Excel

3.6 MATLAB

O MATLAB (MATrix LABoratory), foi criado no final da década de 1970 por Cleve Moler, então presidente do Departamento de Ciência da Computação da Universidade do Novo México, EUA. Sua versão original, baseada nas bibliotecas LINPACK e EISPACK escritas em FORTRAN, logo se espalhou por outras universidades e despertou grande interesse na comunidade matemática. Em 1983, o engenheiro Jack Little teve seu primeiro contato com o MATLAB e, reconhecendo seu potencial comercial, juntou-se a Moler e Steve Bangert para reescrever MATLAB em C e, em 1984, fundar a MathWorks para dar continuidade ao desenvolvimento desta ferramenta. MATLAB foi adotado pela primeira vez por engenheiros de projeto de controle, a especialidade de Little, e rapidamente se espalhou para outros campos de aplicação. Atualmente, é muito utilizado nas áreas da educação, em especial no ensino da álgebra linear e análise numérica, sendo também muito popular entre os cientistas envolvidos com o processamento de imagem (MATHWORKS, 2009).

Principais componentes do MATLAB:

- linguagem de alto nível para computação científica;
- ambiente de desenvolvimento para gerenciamento de códigos, arquivos e dados;
- ferramentas interativas para exploração, design e resolução de problemas;
- funções matemáticas para álgebra linear, estatística, análise de Fourier, filtragem, otimização e integração numérica;
- funções para visualização de dados em gráficos 2D e 3D;
- ferramentas para construção de interfaces gráficas
- funções para integração de algoritmos baseados em MATLAB com aplicativos e linguagens de programação externas como C, C++, Fortran, Java, COM e Microsoft Excel (MOLER, 2004).

3.7 MODELAGEM MATEMÁTICA

Do ponto de vista da engenharia bioquímica, o desenvolvimento da modelagem matemática de bioprocessos permite atingir, entre outros, os seguintes objetivos: organizar informações desconexas a respeito dos fenômenos biológicos em um conjunto coerente; analisar (e calcular) logicamente a respeito de quais componentes e interações são importantes em um sistema complexo; descobrir novas estratégias para explicar o comportamento das células submetidas a determinados ambientes; corrigir falhas eventualmente existentes no entendimento convencional de determinados fenômenos e, finalmente, entender as características qualitativamente essenciais de determinados processos (BAILEY, 1998).

A modelagem matemática é o desenvolvimento das equações que relacionam as diferentes variáveis (de entrada e de saída) e a determinação dos parâmetros associados.

As variáveis de entrada são manipuladas e perturbações (que afastam o processo do seu estado estacionário, e são a principal razão para o uso de controle de processos). Com esta finalidade, são usadas equações de balanço (massa, energia e momento) que descrevem o comportamento do processo a partir das leis que regem os fenômenos físicos e químicos. A esta forma de obtenção dos modelos dá-se o nome de modelagem fenomenológica.

As diversas aproximações de sistemas celulares são classificadas de acordo com o número de componentes usados na representação celular e, também, de acordo com a forma pela qual o conjunto de células é visualizado, se heterogêneo ou homogêneo. Modelos multicomponentes de sistemas celulares são chamados de estruturados, enquanto modelos que não consideram componentes intracelulares são denominados não-estruturados. Modelos segregados abordam o sistema celular de forma heterogênea, enquanto a abordagem homogênea é denominada não segregada. Entretanto, a opção pela forma de representação de um sistema biológico é função da quantidade e da qualidade de informações que se possui, ou que se deseja obter a respeito do mesmo.

Modelos matemáticos são ferramentas muito úteis para a descrição e simulação de diversos sistemas químicos. Eles podem ser usados, basicamente, para um melhor entendimento de processos, para controle, otimização, *scale-up* e projeto de unidades de produção.

A aplicação mais simples de um modelo matemático é seu uso na simulação de processos. Neste caso, as equações do modelo são resolvidas para diferentes condições iniciais e de contorno, além de diferentes valores das variáveis operacionais. Enquanto a obtenção de dados experimentais, por meio da realização de ensaios, pode levar horas ou mesmo dias, a simulação de um modelo em um computador de configuração adequada pode fornecer as informações desejadas quase de forma instantânea. Estudos de simulação permitem, ainda, o teste de novas estratégias de condução do processo, como mudanças, por exemplo, entre *batch* e *fed-batch* (VOLESKY; VOTRUBA, 1992).

De uma forma geral, o primeiro passo para a construção de um modelo matemático representativo é a identificação dos principais fenômenos que interferem no processo produtivo em análise: limitações e inibições por substratos, principalmente no que se refere à existência e ao número de substratos limitantes e/ou inibidores, tipo de produto gerado inibitório ou não, existência ou não de associação da produção dos produtos metabólicos com o crescimento celular, entre outros (BONOMI; SCHMIDELL, 2001). Com isso, a modelagem matemática e simulação preveem o comportamento dinâmico e estacionário do processo, inclusive em condições não testadas empiricamente, possibilitando a determinação das condições operacionais economicamente ótimas do sistema, auxiliando no projeto e ajuste de algoritmos de controle, no qual o modelo matemático formulado passa ser integrante do mesmo (HEINZLE; SANER, 1991).

Os processos bioquímicos incorporam uma série de características que os diferenciam dos processos químicos, o que pode explicar as dificuldades encontradas na formulação de modelos matemáticos que representem adequadamente estes processos, ao contrário do que ocorre com os processos químicos convencionais. Entre essas características podem ser citadas as seguintes: baixas concentrações e baixas velocidades de reação, como resultado da utilização de um meio diluído; conhecimento insuficiente de vários dos fenômenos limitantes das velocidades de produção e falta de sensores para automação on-line;

problemas de esterilidade, segurança e eventualmente da toxicidade dos processos fermentativos (ENGASSER, 1988).

Contudo a modelagem matemática e o controle de processos biotecnológicos esbarram em três características específicas desses sistemas: a falta de medidas, especialmente as medidas *on line*, a variabilidade (não reprodutibilidade) de dados e a complexidade cinética dos processos.

Segundo Fredicson e Tsuchiya (1977), para um correto entendimento da dinâmica de um sistema celular genérico é necessário:

- Identificar o processo que rege o comportamento cinético da população;
- Determinar os fatores ambientais que afetam as mudanças na população e como as diversas velocidades são afetadas por eles;
- Determinar como o ambiente externo é afetado pelo comportamento cinético da população.

Embora as células animais, diferentemente dos microrganismos, tenham a habilidade de produzir proteínas de alto valor agregado, com padrões de modificações pós traducionais adequados para a aplicação clínica, há necessidade urgente de aumentar a eficiência e a produtividade dos bioprocessos que as utilizam. A redução de custos é uma questão essencial para assegurar que a cultura de células animais permaneça competitiva quando comparada a tecnologias alternativas. A modelagem matemática é provavelmente a técnica mais eficiente para atingir esse objetivo (AUGUSTO; BARRAL; PICCOLI, 2008).

Em resumo, a modelagem matemática do processo representa um esforço para identificar, sistematizar e quantificar os principais fenômenos que afetam o crescimento celular e a síntese do produto de interesse e, dessa maneira, racionalizar o processo e otimizar a sua produção, contribuindo para alcançar a sua viabilidade técnico-econômica.

Para estudar a dinâmica de um processo bioquímico, deve-se buscar:

- Identificar os processos que alteram o estado das populações envolvidas (crescimento celular (variação do tamanho da célula), reprodução celular, manutenção da viabilidade celular, morte e lise celular, alterações

morfológicas das células e os processos físicos que incluem entrem outros a aderência das células a superfícies sólidas);

- Identificar os fenômenos ambientais que afetam as velocidades de alteração do estado das populações;
- Identificar como as velocidades de alteração do estado das populações são afetadas;
- Identificar como o ambiente é afetado pelos processos de alteração do estado das populações.

3.7.1 Modelagem Matemática para células S2

Com relação à modelagem da linhagem S2, Barral (2010) encontrou enorme dificuldade de representar as alterações no metabolismo celular empregando um equacionamento do tipo fenomenológico, não estruturado, adotando formulações matemáticas para interpretar essa flexibilidade e utilizando uma sequência de “if’s” lógicos para direcionar o metabolismo.

Esse trabalho propôs um modelo matemático para o cultivo da linhagem de células de inseto S2 e produção da glicoproteína G que considerou oito variáveis de estado. São elas: concentração de células viáveis, células não-viáveis, glicose, glutamina, alanina, glutamato, amônio e GPV e vinte parâmetros. Seu modelo considerou os seguintes fenômenos: a) a velocidade específica de crescimento é limitada por glicose, glutamina e glutamato e inibida por amônio, b) consumo preferencial de glutamina em relação ao glutamato, c) velocidade específica de morte limitada por amônio e inibida por glicose e glutamina, d) velocidade específica de produção de amônio e GPV associada ao crescimento e, finalmente, degradação de GPV.

Para incorporar a variabilidade metabólica observada nos cultivos foi estudada a influência das concentrações de oxigênio, glicose e glutamina na velocidade específica de crescimento máxima e nos fatores de conversão. Além disso foram definidas quatro situações características do processo que apresentam dinâmicas diferentes e que foram implementadas por “if’s” lógicos (sem limitação de GLC e GLN; sem limitação de GLC e com limitação de GLN; com limitação de GLC e sem limitação de GLN; com limitações de GLC e GLN). Os parâmetros foram ajustados

usando a técnica de otimização dos poliedros flexíveis e as equações diferenciais do modelo foram integradas pelo método Gear (BARRAL, 2010).

3.7.2 Modelos não estruturados e estruturados

Os modelos não estruturados e não segregados são os mais utilizados na simulação de cultivos celulares, pois são a forma simples de representar a complexidade biológica. Vale a pena reafirmar que, nos modelos não estruturados e não segregados, a população celular é representada por uma única célula média (AUGUSTO; BARRAL; PICCOLI, 2008).

Entende-se por crescimento balanceado o crescimento no qual a velocidade de produção de um componente da biomassa por unidade de biomassa é constante, igual para todos os componentes da biomassa e igual à velocidade específica de crescimento da própria biomassa. Somente nessa condição de crescimento é que a formulação de modelos não estruturados é perfeitamente justificada. Na prática, o crescimento balanceado só ocorre no estado estacionário em processos contínuos e durante a fase exponencial de crescimento em batelada. Dessa forma, na maioria dos casos, a caracterização da atividade biológica simplesmente pela concentração total de biomassa é insuficiente para uma representação adequada dos dados experimentais pelo modelo matemático formulado (BONOMI; SCHMIDELL, 2001). Vários experimentos têm mostrado que a composição da biomassa de uma população celular varia em resposta a alterações nas condições do ambiente, essas variações na atividade e composição da biomassa requerem uma descrição mais complexa do metabolismo celular e uma estratégia mais estruturada para modelar a cinética (BONOMI; SCHMIDELL, 2001).

Já os modelos estruturados e não segregados consideram a população celular representada por uma célula média, porém as estruturas intracelulares são discriminadas, de forma que a célula é compartimentada e torna-se uma estrutura heterogênea. Dessa forma, é possível buscar uma caracterização do comportamento intrínseco da célula e prever sua interação com o meio com maior precisão (AUGUSTO; BARRAL; PICCOLI, 2008).

As dificuldades na compreensão de todos os fenômenos decorrentes da interação da célula com as condições ambientais em permanente transformação torna necessário detalhar as estruturas celulares, característica dos modelos estruturados. Dependendo da quantidade de detalhes e da estrutura fisiológica, esses modelos estruturados melhoram sua capacidade de previsão e tornam-se ferramentas de grande valor para a compreensão da fisiologia das células.

Embora mais robustos que os não estruturados, os modelos estruturados descrevem a resposta celular dentro de uma faixa estreita de condições de interesse. Assim, a fim de prever a resposta celular para uma maior variedade de estratégias de operação em biorreatores, pode-se ampliar o modelo com um maior número de estruturas, de maneira a acumular informações metabólicas, ou associar a dinâmica de fluxos metabólicos ao modelo estruturado (FOLLISTAD *et al.*, 1999). Essa metodologia de balanço de fluxos, que utiliza uma série de reações a partir do metabolismo da glicose e calcula todas as velocidades das reações do metabolismo no estado estacionário, permite conhecer melhor o comportamento celular e simular condições de operação mais eficientes (AUGUSTO; BARRAL; PICCOLI, 2008).

3.7.3 Equacionamentos

3.7.3.1 Reações Metabólicas

Uma reação química pode ser analisada quanto à sua termodinâmica ou quanto à sua cinética. A Termodinâmica estuda a viabilidade e a reversibilidade das equações, a partir da análise do conteúdo energético dos estados inicial e final de uma transformação - no caso das reações químicas, o conteúdo energético dos produtos e reagentes. Nada esclarece, porém, sobre a velocidade com que a transformação ocorre. Essa informação é dada pela Cinética (TORRES, 2001). As reações metabólicas de uma célula são catalisadas por enzimas.

3.7.3.2 Crescimento Biológico

Em 1942, Monod descreveu um modelo que é, hoje, o ponto de partida para a maioria dos tratamentos matemáticos dados ao crescimento biológico. Monod

reconheceu, primeiramente, que um meio de cultura poderia ser formulado de maneira que apenas a concentração de uma determinada substância afetasse a taxa de crescimento. Assumindo que esta substância é um substrato (S) para o crescimento, ela foi denominada de substrato limitante.

Sob a hipótese de que há apenas um substrato limitante presente no meio de cultura, a velocidade específica de crescimento irá depender apenas da sua concentração. Monod assumiu que esta dependência poderia ser expressa pela equação de Michaelis-Menten, advinda da cinética enzimática (MARZZOCO; TORRES, 1990).

$$\mu = \mu_m \cdot \frac{S}{K_s + S} \quad (\text{MONOD, 1949}) \quad (1)$$

Estudos posteriores comprovaram a aplicabilidade da equação de Monod para inúmeros casos de crescimento biológico, a menos da fase lag e do fenômeno de inibição por um substrato limitante que, entre outros fenômenos, não são representados por essa equação. A constante μ_m é a velocidade específica máxima de crescimento da população, a qual é alcançada quando S é muito maior que K_s e, por sua vez, K_s é conhecida como constante de saturação. Em particular K_s é a concentração de substrato limitante na qual μ é metade de μ_m (HISS, 2001).

A expressão de Monod (Eq. 1) é um modelo que não leva em conta o efeito inibidor, tanto pelo substrato como pelo produto formado. Essa, porém, não é a única interpretação referente a tal condição ideal de cultivo.

Vários outros modelos para a representação do crescimento celular estão presentes na literatura (MOSER, 1985):

- Equação de Teissier:
$$\mu = \mu_m \cdot \left(1 - e^{-\frac{S}{K_s}} \right) \quad (2)$$

- Moser:
$$\mu = \mu_m \cdot \frac{S^n}{K_s + S^n} \quad (3)$$

- Contois e Fujimoto:
$$\mu = \mu_m \cdot \frac{S}{K_s X + S} \quad (4)$$

- Powell:
$$\mu = \mu_m \cdot \frac{S}{(K_s + K_D) + S} \quad (5)$$

Há pelo menos mais seis outras expressões, propostas por outros autores e que não levam em conta o fenômeno da inibição. A ausência da inibição é, na verdade, uma situação pouco comum na prática, principalmente durante um cultivo descontínuo, onde há um crescente acúmulo de metabólitos que acabam interferindo desfavoravelmente sobre o metabolismo e crescimento celular (HISS, 2001).

O efeito do substrato se manifesta quando um valor alto da concentração inicial S pode, ao invés de aproximar μ_x de μ_m , provocar um efeito contrário, ocasionando uma inibição no crescimento celular (HISS, 2001).

Com o objetivo de explicar essa redução na velocidade específica de crescimento, provocada pelos altos valores iniciais de concentração de substrato, uma modificação na expressão de Monod (Eq. 6), foi proposta (MOSER, 1985):

$$\mu = \mu_{\max} \cdot \frac{S}{K_S + S} \cdot \frac{K_{I,S}}{K_{I,S} + S} \quad (6)$$

Onde K_S é a constante de saturação e $K_{I,S}$ é a constante de inibição pelo substrato a que se refere, como K_S , ao valor de S para o qual $\mu_x = \mu_m/2$, porém para um valor de S que provoque a inibição, sendo assim superior ao correspondente S da equação de Monod (HISS, 2001).

3.7.4 Aspectos Biológicos das Vias Metabólicas

Existem dois tipos possíveis de metabolismo: o anabolismo e o catabolismo. O anabolismo compreende a síntese ou formação de compostos através do uso de energia e do consumo de reagentes, enquanto o catabolismo é a degradação, ou “quebra” de compostos, liberando energia e produzindo compostos que servem de reagentes para outras reações (MOTTA, 2011). Tanto um quanto outro se dão através de reações químicas.

A velocidade em que as reações ocorrem nas células é um fator determinante para a sustentação da vida dessas células. Entretanto, as maneiras usuais pelas quais os pesquisadores estimulam, ou catalisam, as taxas de reações, tais como a elevação da temperatura, o aumento da pressão, o aumento da concentração de um dos reagentes ou a adição de ácidos e bases, não ocorrem naturalmente em nível

celular. As enzimas são os catalisadores das reações químicas das células, aumentando as taxas de reações em várias ordens de magnitude. Geralmente, uma enzima catalisa um pequeno número de reações químicas e para que estas reações ocorram é necessária a participação desta enzima específica. Desta forma, as enzimas desempenham o papel de incentivar a conversão dos substratos em produtos (LEHNINGER; NELSON; FOX, 2004).

As atividades enzimáticas são classificadas e catalogadas pelo *International Union of Biochemistry and Molecular Biology* (NC-IUBMB, 2012). Cada enzima é classificada por um *EC Number*, que é um código hierarquicamente definido através de 4 números, separados por pontos, no formato X.Y.W.Z, cada número refinando progressivamente a ação enzimática realizada (LARNER, 1974).

O primeiro número, X, define o tipo de reação catalisada, os dois números seguintes, Y e W, indicam a classe e subclasse da enzima enquanto Z, o último número, é um identificador sequencial da enzima dentro desta classificação hierárquica (NC-IUBMB, 2012).

O metabolismo da célula permite que ela mantenha-se viva, cresça e se reproduza. Para que essas fases da vida celular ocorram é necessária a presença contínua de fontes de energia que servirão como substratos iniciais para uma série de reações encadeadas que terão como produto final os constituintes celulares, propiciando tanto a garantia da vida celular quanto o crescimento e a reprodução da célula.

Organismos incapazes de realizar fotossíntese obtêm energia de maneira secundária, alimentando-se de organismos que possuem algum tipo de energia química armazenada (MOTTA, 2011). ATP e NADP são biomoléculas energéticas pois armazenam energia quimicamente e funcionam como fontes de energia para o metabolismo celular, de tal forma que quando reagem com outras moléculas no interior das células, a energia liberada é utilizada em outros processos intracelulares (GARRET; GRISHAM, 1999).

O metabolismo de um organismo é também chamado de rede metabólica ou de vias metabólicas do organismo. Essas outras denominações procuram deixar mais evidente o fato de que as reações ocorridas nas células estão inter-relacionadas, visto que os produtos de uma reação geralmente servem como substratos para outras reações.

O processo de reconstrução de uma rede metabólica consiste em inferir, a partir de uma rede metabólica de referência e do genoma de um organismo, as reações e vias metabólicas presentes neste organismo. Esse processo de reconstrução da rede metabólica de um organismo parte do genoma deste organismo, identificando-se a partir daí os genes que sintetizam proteínas e, destas proteínas, quais são enzimas. O passo seguinte é identificar as funções catalíticas de cada enzima sintetizada pelo organismo, sabendo-se assim as reações que devem estar presentes no organismo (SAGOT, 2008).

Para realizar a reconstrução de uma rede metabólica já existem várias ferramentas que automatizam grande parte do esforço, localizando genes e identificando as enzimas, cruzando informações em bancos de dados específicos para descobrir as funções destas enzimas, obtendo os seus *EC numbers* e a partir disso reconstruindo a rede metabólica do organismo. Uma das ferramentas mais utilizadas para este propósito é o KEGG - *Kyoto Encyclopedia of Genes and Genomes* (KEGG, 2012).

O KEGG é um banco de dados contendo informações sobre vias lineares, além de informações específicas sobre compostos, enzimas e reações, dentre outras. As vias metabólicas contidas na base de dados são apresentadas em formato gráfico, com *hiperlinks* para facilitar a navegação entre vias relacionadas. Outra facilidade é a obtenção de dados adicionais sobre as enzimas que catalisam as reações apresentadas. Além da apresentação das vias lineares, o banco oferece recursos para obter esses dados em formato XML.

3.7.5 Modo Elementar

A estequiometria de uma reação são as quantidades necessárias de cada um dos reagentes desta reação e as quantidades produzidas de cada um dos produtos da reação. A seguir apresenta-se os conceitos de modo e modo elementar e de como eles podem ser utilizados para modelar redes metabólicas, atendendo a restrições estequiométricas, termodinâmicas, entre outras. Uma abordagem de modelagem de rede metabólica é dita ser baseada em restrições quando ela visa contemplar todas as restrições quantitativas, termodinâmicas, regulatórias, etc. existentes em nível celular.

Em uma abordagem baseada em restrições deve-se adicionar regras para restringir os fluxos permitidos para a rede, através dos dados estequiométricos, restrições termodinâmicas ou outras restrições quaisquer. As principais regras servem para garantir o estado de equilíbrio (*steady state*), para o qual todo composto produzido deve ser consumido, e as restrições termodinâmicas, que ditam que reações irreversíveis só podem ocorrer na direção apropriada.

Aplicando-se essas restrições à uma rede metabólica, apenas fluxos possíveis, isto é, que atendam às restrições definidas, são de interesse. Um fluxo admissível, chamado modo, em uma rede metabólica na qual definiram-se restrições é um conjunto de reações que ocorrem determinadas vezes e em determinada sequência, transformando substratos em produtos, de forma que todos os compostos envolvidos fiquem balanceados – atendam às restrições estequiométricas – e que as reações irreversíveis ocorram sempre na direção apropriada. Um problema relacionado a essa definição de modo é a análise de fluxo buscando-se um modo que otimize uma função objetivo, como a biomassa ou a produção de ATP (CHIERICHETTI *et al.*, 2009).

Para os casos em que não há uma função objetivo a ser otimizada, todos os modos são igualmente interessantes e um problema importante passa a ser encontrar um conjunto de modos elementares, isto é, modos que tem a propriedade de não conterem nenhum outro modo. O conjunto de modos elementares é, desta forma, um conjunto gerador de todos os modos da rede metabólica (CHIERICHETTI *et al.*, 2009).

Modos elementares representam uma formalização da definição biológica de via metabólica, já que um modo pode ser visto como um conjunto de enzimas – ou de reações que essas enzimas catalisam – que operam juntamente em estado de equilíbrio para produzir determinados produtos e um modo é elementar se a remoção de uma das enzimas faz com que os produtos não sejam mais produzidos.

Uma matriz estequiométrica S de uma rede metabólica é uma matriz com n linhas e m colunas, n sendo o número de metabólitos e m o número de reações. A posição $S(i, j)$ da matriz S tem valor k se a reação j produz k unidades do metabólito i , neste caso dizemos que i é a saída da reação j . A posição $S(i, j)$ da matriz S tem valor $-k$ se a reação j consome k unidades do metabólito i , neste caso dizemos que i é a entrada da reação j . A posição $S(i, j)$ da matriz S tem valor 0 caso o metabólito i não

esteja envolvido na reação j . O valor k é o coeficiente estequiométrico do metabólito i na reação j . Uma matriz estequiométrica resume a estrutura de uma rede metabólica (CHIERICHETTI *et al.*, 2009).

Considere que o conjunto de reações de uma rede metabólica seja dividido em dois: O conjunto R contendo todas as reações que são reversíveis e o conjunto I contendo as reações irreversíveis. Um vetor de fluxo – ou distribuição de fluxo – v é um vetor de dimensão m do espaço de reações R_m , no qual o elemento v_i descreve o fluxo na i -ésima reação. Um vetor de fluxo é um modo se:

- $Sv = 0$;
- $v_i > 0, \forall i \in I$.

Chama-se de $R(v)$ o conjunto de reações que participam do fluxo. Desta forma, um modo v é dito ser um modo elementar caso não haja um outro modo w tal que $R(w) \subset R(v)$. Assim, podemos utilizar álgebra linear para verificar se o vetor v é um modo e depois verificar se o modo é elementar.

A abordagem utilizando-se modos elementares é matematicamente mais precisa e permite uma análise quantitativa da rede metabólica, em detrimento da análise estritamente qualitativa proporcionada pela modelagem através de um grafo bipartido, por exemplo, através da qual obtém-se uma modelagem estrutural sem, contudo, incluir quaisquer informações regulatórias ou regras relacionadas às quantidades dos compostos envolvidos nas reações.

Todavia, é computacionalmente inviável avaliar todos os modos elementares existentes para uma rede metabólica, a fim de obter-se uma enumeração dos conjuntos de nutrientes possíveis para a sintetização de determinados compostos. O que se procura fazer, devido a essas restrições de desempenho, é realizar uma estratégia mista, em que se obtém uma série de conjuntos de reações candidatas a serem modos elementares a partir de uma análise qualitativa, fazendo-se em um segundo passo a validação quantitativa destes conjuntos candidatos, avaliando-se então se são modos elementares.

Assim, a um conjunto mínimo de reações enzimáticas irreversíveis e em regime estacionário que formam um modo de fluxo indecomponível, dá-se o nome de Modo Elementar de Fluxo. Um conjunto de modos elementares é único para uma dada rede metabólica (SCHUSTER; FELL; DANDEKAR, 2000). Com isso, a análise de

equações elementares se tornou um método importante para o estudo de redes metabólicas (SCHUSTER; FELL; DANDEKAR, 2000; SCHUSTER *et al.*, 2002) e que permite enumerar caminhos mínimos sistematicamente independentes por uma rede que é estequiométrica e termodinamicamente possível.

3.7.6 Análise de Fluxos Metabólicos

A obtenção desses modos elementares de fluxos permite testar se conjuntos de enzimas formam uma via metabólica consistente, compreendendo, principalmente, o direcionamento das reações, a irreversibilidade (GOMBERT; NIELSEN, 2000). A estequiometria das reações metabólicas e sua reversibilidade precisam ser conhecidas e os metabólitos classificados como externos ou internos. Os metabólitos internos têm que ser equilibrados pela produção e consumo das redes metabólicas.

Os primeiros trabalhos que fizeram uso de um modelo estequiométrico para a interpretação de dados experimentais são de décadas passadas (AIBA; MATSUOKA, 1979; PAPOUSTSAKIS; MEYER, 1985). No entanto, somente com o trabalho de Holms (1986) a Análise de Fluxos Metabólicos (AFM) foi introduzida como uma técnica de análise do metabolismo e mais tarde, com o trabalho de Vallino e Stephanopoulos (1990), esta técnica foi sistematizada através do uso de álgebra linear e cálculo matricial.

Com o rápido avanço, no início da década de 90, das técnicas de biologia molecular e, principalmente, da tecnologia do RNA recombinante, aumentou muito o espectro de aplicações desta tecnologia em vários setores da biotecnologia, como por exemplo de organismos utilizados na obtenção de produtos de interesse do homem.

Para a análise de vias metabólicas, uma das ferramentas mais eficientes é a Análise de Fluxos Metabólicos (AFM) (NIELSEN, 1993). A AFM traz informações precisas sobre o funcionamento do metabolismo de um organismo numa determinada condição e pode ser utilizada, em conjunto com outras ferramentas analíticas e matemáticas.

3.7.7 Metatool

Nos últimos anos, novas aproximações para o cálculo de equações elementares das redes metabólicas foram desenvolvidas. Como consequência, o programa Metatool (PFEIFFER *et al.*, 1999) é um dos primeiros programas que se dedicou a este propósito. O *software* Metatool, que é uma implementação eficiente da análise de modos elementares, é utilizado como auxiliar ao entendimento das principais reações metabólicas que acontecem no interior das células. Porém, não é utilizado para trabalho em engenharia metabólica visando modificações genéticas para alterações metabólicas. Para esse fim, pode-se utilizar o *GEPASI* (www.gepasi.org), que é um programa para modelagem de sistemas bioquímicos e simula a cinética dos sistemas das reações bioquímicas fornecendo uma série de ferramentas para ajustar modelos aos dados, otimizando o modelo e realizando análise de controle metabólico e estabilidade linear (MENDES, 1997). O programa *FluxAnalyzer* (KLANT *et al.*, 2003), fornece uma interface gráfica e algumas transformações nos modos elementares (KLANT; GILLES, 2004), porém necessita ser usado juntamente com o *SEPARATOR* (SCHUSTER *et al.*, 2002) para intercomunicação de dados. O *YANA* oferece um ambiente de modelagem integrado com capacidades de troca de dados padronizados e além da análise dos modos elementares, permite a edição de modificações convenientes das redes metabólicas e possui extração automática de redes metabólicas do KEGG (SCHWARZ *et al.*, 2005).

Outro programa que pode ser utilizado para a Engenharia Metabólica, mas que não utiliza o conceito de modos elementares, é o *OptFlux* (ROCHA *et al.*, 2010), utilizado para introduzir mudanças genéticas específicas em células microbianas, prevendo seus efeitos.

A análise de equações elementares é uma ferramenta poderosa na modelagem de redes metabólicas. Nos últimos anos, novas aproximações para o cálculo de equações elementares das redes metabólicas foram desenvolvidas.

A estequiometria das reações metabólicas e sua reversibilidade têm que ser conhecidas e os metabólitos têm que ser classificados como externos ou internos. Os metabólitos internos têm que ser equilibrados pela produção e consumo das redes metabólicas.

Os coeficientes estequiométricos são coletados em uma matriz N onde as linhas correspondem aos metabólicos internos e as colunas às reações. A reversibilidade das reações não é integrada na matriz N , mas é levada em consideração posteriormente.

A modelagem fenomenológica é baseada nos balanços materiais das variáveis de estado e nas relações cinéticas macroscópicas. Embora estes modelos sejam bastante empregados nas simulações e otimizações dos processos, as reações que ocorrem dentro das células não são consideradas. Se notarmos que diferentes fenômenos como inibição (substratos e/ou produtos) e limitação pelos substratos ocorrem simultaneamente no interior da célula, dificultando a resposta do modelo, parece bastante coerente o desenvolvimento de ferramentas que consideram a dinâmica dos fluxos de metabólitos internos da célula. A análise do fluxo metabólico propõe estimar o fluxo intracelular de metabólitos obedecendo à estequiometria das reações metabólicas que representam o sistema e empregando as medidas de concentrações extracelulares.

Nessa direção, estuda-se o metabolismo de células de maneira a compreender o controle das reações metabólicas e assim relacionar a modelagem dinâmica macroscópica e a análise dos fluxos metabólicos. A consequência direta dessa metodologia é permitir a predição das concentrações de produtos não medidos e prever novas formas de cultivos com diferentes conversões e produtividades.

O software Metatool é capaz de gerar, a partir do conjunto das reações prováveis da via metabólica, diferentes subconjuntos que representam os caminhos mais simples capazes de conectar os substratos com os produtos finais.

Alguns trabalhos utilizando o conceito de Modos Elementares foram elaborados utilizando o Metatool, dentre eles Provost e Bastin (2004) produziram uma identificação de modelos matemáticos de processos biológicos muito comuns onde medidas extracelulares no meio de cultura são os únicos dados disponíveis além da medida da própria biomassa. A partir dos modos elementares calculados pelo Metatool eles empreenderam uma metodologia matemática para cálculo dos fluxos metabólicos, sempre mantendo o crescimento celular baseado na velocidade específica de crescimento celular do próprio ensaio. Além disso, a metodologia matemática implantada nos cálculos dos fluxos metabólicos leva em conta a igualdade de equações e incógnitas, zerando incógnitas para atingir essa igualdade.

Provost *et al.* (2006) trabalharam distintamente nas fases exponencial, estacionária e declínio, propondo um metabolismo para cada etapa e utilizando a velocidade específica de crescimento celular própria de cada etapa. Como método matemático utilizado para encontrar os fluxos metabólicos, Provost *et al.* (2006) trabalharam com cones convexos, onde a solução são as bordas de um cone convexo poliédrico.

Outros trabalhos utilizaram o Metatool para expressar os modos elementares como Raicher (2011) e Taciro (2008). Já Piccoli *et al.* (2009) utilizaram o Metatool para os modos elementares e aplicaram um método matemático baseado no *fmincom* do MATLAB para encontrar os fluxos metabólicos, partindo também do crescimento celular baseado na velocidade específica de crescimento celular do próprio ensaio. A quantidade de modos elementares encontrados em cada trabalho depende da complexidade da rede metabólica estudada.

3.7.8 Ajuste de Parâmetros

3.7.8.1 Ajuste do Modelo Matemático

Os bioprocessos, em geral, podem ser modelados através de um conjunto de equações cinéticas e diferenciais ordinárias, onde cada equação descreve o comportamento de uma variável de estado do processo (concentrações de glicose, glutamina, amônio, alanina, etc.) em relação ao tempo.

As equações diferenciais e cinéticas possuem alguns parâmetros (fator de conversão de substrato a célula, por exemplo) que precisam ser determinados com precisão para que essas equações possam descrever adequadamente o processo que está sendo modelado.

As estimativas iniciais dos parâmetros podem ser obtidas através da exploração dos dados experimentais ou diretamente da literatura e passam a ser as sementes (comumente chamada de “chute inicial” ou “estimativa inicial”). A semente é uma estimativa da solução e, quanto mais próximas estiverem as duas (semente e solução), mais fácil será a otimização.

Para o refinamento na precisão dos parâmetros, em geral, utiliza-se um método numérico de otimização de busca (direta ou indireta) que faz o ajuste dos parâmetros de forma que as equações diferenciais possam prever com o máximo de exatidão a variação do processo ao longo do tempo. Os métodos numéricos de otimização da busca procuram a solução a partir de uma semente.

Os métodos numéricos de otimização de busca, em geral, seguem a seguinte dinâmica:

1. definir um conjunto de dados experimentais que será utilizado para determinar se o ajuste é adequado. Esse conjunto de dados possui os valores de todas as variáveis de estado do processo em um determinado intervalo de tempo. Tais dados devem ser tratados previamente para evitar erros. Esse tratamento de dados tem por finalidade, por exemplo, eliminar ou corrigir pontos que foram lidos erroneamente e corrigir distorções de interpretação dos dados;
2. definir uma semente, que é a estimativa inicial dos parâmetros a serem ajustados;
 1. definir limites (valores máximos e mínimos) permitidos para cada parâmetro;
 2. estabelecer o balanço de massa para o sistema, i.e., conjunto de equações diferenciais ordinárias que descrevem o processo;
 3. estabelecer as equações cinéticas que definem fenômenos presentes no sistema;
 4. selecionar o método numérico para a resolução de equações diferenciais ordinárias (Euler, Runge-Kutta, etc.). Assim, através de um método numérico para a resolução de equações diferenciais ordinárias, se resolve o conjunto de equações utilizando os parâmetros adotados e o mesmo intervalo de tempo dos dados experimentais (BONOMI; SCHMIDELL, 2001);
 5. definir a função objetivo, normalmente uma minimização ou maximização de valores. O mais usual neste tipo de problema é a minimização do resíduo, i.e., a soma da diferença entre os valores das variáveis de estado calculados pelo modelo e os valores obtidos experimentalmente (mínimos quadrados, máxima verossimilhança, etc.).

6. a otimização é um processo iterativo onde, a cada repetição, os valores dos parâmetros são recalculados de forma a obter um ajuste mais preciso. Os passos de 1 a 3 são repetidos até que o valor do resíduo, entre uma iteração e outra, seja inferior a uma precisão estipulada (10^{-8} , por exemplo).

Os problemas, frequentemente, encontrados ao efetuar regressões não lineares são:

- maior dependência da qualidade da estimativa inicial adequada dos parâmetros;
- existência de mínimos locais na função objetivo, ou seja, a função resíduo apresenta diversos valores mínimos, que atraem a solução do método de regressão empregado, dificultando a convergência do algoritmo para o mínimo absoluto;
- necessidade da escolha adequada da função objetivo (mínimos quadrados, máxima verossimilhança, etc.);
- existência de interação entre parâmetros.

3.7.8.2 Equações Diferenciais Ordinárias

As Equações Diferenciais são casos particulares de equações funcionais com diferenciais e/ou derivadas. É fato que, embora se chame de Equações Diferenciais, tais equações relacionam mais diretamente derivadas e não diferenciais. Geralmente o relacionamento com as diferenciais se dá de uma forma indireta quando consideramos as derivadas como um quociente de diferenciais (ZAMBONI; MONEZZI JR, 2002).

O estudo de Equações Diferenciais tem dois objetivos principais:

- determinar a equação que melhor explica um determinado fenômeno;
- determinar a solução mais apropriada para esta equação.

Uma Equação Diferencial Ordinária (EDO) é aquela que relaciona derivadas de uma função incógnita de uma variável real $y = y(x)$. Os algoritmos de Euler, de Runge-Kutta de Segunda Ordem e de Runge-Kutta de Quarta Ordem são três dos muitos algoritmos utilizados na resolução numérica de Equações Diferenciais. O Algoritmo

de Euler é conhecido como de primeira ordem, pois o polinômio considerado na aproximação é de ordem 1. O Algoritmo de Runge-Kutta de Segunda Ordem considera um polinômio de ordem 2 e o de Runge-Kutta de Quarta Ordem um de ordem 4. Em Sermutlu (2004) se pode ver uma comparação entre os algoritmos de Runge-Kutta de quarta e de quinta ordem. Para passos grandes, o algoritmo de ordem 4 é melhor, pois obtém resultados mais precisos em menor tempo. Porém, quando o passo diminui o algoritmo de ordem 5 apresenta melhores resultados.

O *software* MATLAB (MATHWORKS, 2010a) também possui algumas funções implícitas para a resolução algébrica e numérica de EDOs: *dsolve*, *ode23*, *ode45*, *ode113*, *ode15s*, *ode23s*, *ode23t* e *outras* (MATHWORKS, 2001), onde *ode45* representa Runge-Kutta de quarta e quinta ordem.

3.7.8.3 Função Objetivo

Os métodos matemáticos utilizados para a otimização, em geral, possuem uma Função Objetivo que deve ser maximizada ou minimizada para que a modelagem seja adequada ao processo. No caso específico de bioprocessos que descrevem o crescimento celular, a Função Objetivo calcula os resíduos entre os valores encontrados pelo modelo e os valores experimentais. O ajuste de parâmetros deste tipo de modelo matemático tenta minimizar o valor da Função Objetivo, minimizando assim a discrepância entre os valores calculados e os valores obtidos experimentalmente.

Um dos aspectos que tem se mostrado crucial no ajuste de parâmetros por diferentes métodos de regressão não linear é a forma como os resíduos são calculados na Função Objetivo.

- **Mínimos Quadrados**

Em 1809, Carl Friedrich Gauss (1777-1855) publicou um artigo demonstrando que a melhor maneira de determinar um parâmetro desconhecido de uma equação pré estabelecida é minimizando a soma dos quadrados dos resíduos, mais tarde chamado de Mínimos Quadrados por Adrien-Marie Legendre (1752-1833). Em abril

de 1810, Pierre-Simon Laplace (1749-1827) apresenta no *memoir* da Academia de Paris (OEUVRES, 1810) a generalização a problemas com vários parâmetros desconhecidos.

Um programa de mínimos quadrados sempre começa com a minimização da soma:

$$S = \sum_{i=1}^N (y_i^o - y_i)^2 \quad (7)$$

Onde chamamos de : y_i^o = valores observados de y

y_i = valores calculados de y

Ou seja, mínimos quadrados implica em minimizar os quadrados dos resíduos. Ou melhor, mínimos quadrados é o somatório do quadrado de desvios entre valores preditos por um modelo e valores experimentais (Figura 7).

Figura 7 - Mínimos quadrados

Por que este critério é considerado um bom critério e não simplesmente minimizar os resíduos ou o cubo dos resíduos? A resposta formal é que os mínimos quadrados são corretos se os resíduos tiverem uma distribuição gaussiana (normal) (Figura 8).

Figura 8 - Distribuição gaussiana (normal) para uma variável x, com média m e desvio padrão s.

É simples notar que se os resíduos forem minimizados diretamente, um grande resíduo negativo pode ser anulado por um grande resíduo positivo, enquanto que com o quadrado minimiza-se os módulos das diferenças.

- **Outras fórmulas para cálculo de resíduos**

A Tabela 1 apresenta várias formas para o cálculo de resíduos entre os valores calculados pelo modelo e os valores experimentais, indicando, quando for o caso, os problemas observados quando da sua utilização. Nesta tabela são indicadas as fórmulas para cálculo dos resíduos que apresentaram melhores resultados. Sabe-se, entretanto, que a melhor fórmula para cálculo do resíduo depende do método de ajuste e também da estimativa inicial dos parâmetros empregada (BONOMI; SCHMIDELL, 2001).

Na Tabela 1:

- R é o resíduo;
- y_i é o valor experimental da variável;
- $\tilde{y}_i$ é o valor calculado da variável;
- $(y_i)_m$ é o máximo valor da variável experimental;
- $(\tilde{y}_i)_m$ é o máximo valor da variável calculada;
- r é o coeficiente de regressão linear entre as variáveis experimentais e calculadas;
- a é o coeficiente angular combinado entre as variáveis experimentais e calculadas.

Tabela 1 - Fórmulas para o cálculo de resíduos (Adaptado de BONOMI; SCHMIDELL, 2001)

Número	Fórmula	Problemas na utilização
1	$R = \sum_i (\tilde{y}_i - y_i)^2$	Variáveis com elevado valor absoluto são privilegiadas no ajuste
2	$R = \sum_i \left(\frac{\tilde{y}_i}{(\tilde{y}_i)_m} - \frac{y_i}{(y_i)_m} \right)^2$	Tendência a ajustar melhor as variáveis próximas aos valores máximos
3	$R = \sum_i \left(\frac{\tilde{y}_i - y_i}{\tilde{y}_i} \right)^2$	Resíduos muito elevados para valores muito pequenos da variável calculada
4	$R = \sum_i \left(\frac{\tilde{y}_i - y_i}{y_i} \right)^2$	Resíduos muito elevados para valores muito pequenos da variável experimental
5	$R = \sum_i \left(\frac{\tilde{y}_i - y_i}{\left(\frac{\tilde{y}_i + y_i}{2} \right)} \right)^2$	Resíduos elevados para valores muito pequenos e diferentes das variáveis calculada e experimental
6	Idem fórmula "5", porém R só é calculada quando $y_i > \varepsilon (y_i)_m$	-
7	$R = \sum r-1 + \sum a-1 $, r e a são calculados para cada variável e para cada ensaio	-
8	$R = r-1 + a-1 $ r e a são calculados com todas as variáveis normalizadas e todos os ensaios ajustados por uma única reta	-

• Normalização de valores para o cálculo de resíduos

Na modelagem matemática de bioprocessos que descrevem o crescimento celular, as variáveis de estado envolvidas podem ter valores de magnitudes muito diferentes, alguns muito pequenos e outros muito grandes. Para que a diferença de magnitude entre os valores que estão sendo utilizados no ajuste de parâmetros não tenha interferência sobre o valor do resíduo calculado no processo de ajuste de parâmetros, muitas vezes, o cálculo de resíduos é efetuado utilizando-se valores calculados e valores experimentais normalizados.

Um vetor normalizado possui duas características importantes:

- todos os seus componentes possuem valores entre -1 e 1;
- o módulo do vetor é sempre igual a 1, por isso, um vetor normalizado é conhecido também como vetor unitário.

O processo de normalização é bem simples: basta dividir cada componente do vetor pelo módulo do mesmo, obtendo assim um vetor unitário (HIMMELBLAU, 1970):

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum x_i^2}} \quad (8)$$

Como exemplo, vamos considerar o vetor:

$$x = [1, 3, -2, 0, 5] \quad (9)$$

O módulo deste vetor pode ser calculado

$$|x| = \sqrt{1^2 + 3^2 + (-2)^2 + 0^2 + 5^2} = \sqrt{39} \quad (10)$$

O vetor normalizado (unitário) resultante é:

$$\tilde{x} = \left[\frac{1}{\sqrt{39}}, \frac{3}{\sqrt{39}}, \frac{-2}{\sqrt{39}}, \frac{0}{\sqrt{39}}, \frac{5}{\sqrt{39}} \right] = [0.1601, 0.4804, -0.3203, 0.0000, 0.8006] \quad (11)$$

O *software* MATLAB já possui funções prontas para normalizar linhas e colunas de uma matriz. A função *normc* é utilizada para colunas e a função *normr* é utilizada para linhas.

3.7.8.4 Otimização da busca

Essa otimização envolve a maximização ou minimização de funções, através da variação de seus parâmetros. A solução de um problema de minimização é o conjunto de valores de parâmetros que torna uma dada função mínima em um espaço de busca (ou espaço de soluções). O espaço de soluções é o domínio de valores possíveis para os parâmetros a otimizar. Quando esse espaço é finito, diz-se que o problema possui restrições. Estas restrições podem ter as mais variadas causas. A otimização numérica da busca de soluções que satisfaçam funções objetivo multivariáveis não lineares requer a aplicação de técnicas eficientes e robustas. A eficiência é importante, já que os problemas são resolvidos de forma iterativa. Tentativa e erro torna-se impraticável para mais de três ou quatro variáveis. A robustez, habilidade de se alcançar a solução ótima, torna-se desejável, pois funções objetivo não lineares geralmente são imprevisíveis em seu comportamento.

- **Principais técnicas de otimização de busca**

Técnicas de otimização de busca são utilizadas para encontrar um conjunto de soluções (neste caso valores de parâmetros) que podem ser definidos de alguma forma como ótimos. Em um caso simples, pode ser a maximização ou minimização de alguma característica do sistema que dependa da função. Neste caso, é uma otimização sem restrição (*unconstrained*). Em uma formulação mais avançada, a função objetivo a ser maximizada ou minimizada, pode sofrer restrições na forma de equações ou inequações, bem como a imposição de limites superiores e inferiores para cada parâmetro. Neste caso, é uma otimização com restrição (*constrained*) (MATHWORKS, 2010b).

Apesar de existir uma grande quantidade de métodos para minimização sem restrição, tais métodos podem ser agrupados de acordo com o uso ou não das informações provenientes das derivadas da função objetivo (MATHWORKS, 2010b). Se na busca pelo ponto de mínimo da função objetivo é feito uso de derivadas parciais desta função, o método utilizado é classificado como derivativo. Caso contrário, o método é classificado como não derivativo (HIMMELBLAU, 1970).

Além dos métodos de busca direta e indireta, que são utilizados para problemas de otimização sem restrição, também existem métodos de busca de direção viável, utilizados para otimização com restrição, como o método de programação quadrática sequencial (SQP - *Sequential Quadratic Programming*). A seguir tem-se uma breve descrição dos principais métodos de otimização. Além destes métodos, vários autores descrevem outros métodos para otimização não linear (BAZARRA; SHERALI; SHETTY, 1979; GILL; MURRAY; WRIGHT, 1981; WILDE; BEIGHTLER, 1967; BEVERIDGE; SCHECHTER, 1970).

- ❖ **Métodos de busca direta ou não derivativos**

O método de busca direta proposto por Hooke e Jeeves (1961) consiste na estimativa de valores iniciais para os parâmetros do modelo em questão, incrementando-os um a um, por um valor Δ previamente estabelecido. O valor assumido pela função objetivo é obtido para cada novo parâmetro individualmente. Caso não haja melhoria no valor da função objetivo, este parâmetro não é alterado,

permanecendo pois, com o seu valor anterior. Sequencialmente, um novo parâmetro é testado até o alcance da configuração global de um novo conjunto de parâmetros, o qual define uma direção de sucesso na otimização em busca do mínimo (HIMMELBLAU, 1970).

Entre as principais técnicas derivativas ou busca direta destacam-se os métodos Simplex e Poliedros Flexíveis, descritos a seguir:

- **Método Simplex**

O Método Simplex, formulado por (SPENDLEY; HEXT; HIMSWORTH, 1962), usa uma figura regular geométrica (um simplex), selecionando os vértices desta figura, nos quais é avaliada a função objetivo. A direção a ser seguida pelo método, na busca do ótimo, depende de qual vértice apresenta os valores mais elevados, quando do cálculo da função objetivo (piores parâmetros).

- **Método Simplex Modificado ou Métodos dos Poliedros Flexíveis**

O método de Nelder e Mead é um método simplex geométrico flexível, conhecido como o método dos poliedros flexíveis (HIMMELBLAU, 1970). Este método minimiza uma função de n variáveis independentes, usando $(n+1)$ vértices de um poliedro no espaço E^n . Cada vértice é definido por um vetor x (neste caso, por um conjunto de parâmetros). O vértice em E^n que fornece o maior valor da função objetivo (neste caso o maior resíduo entre as variáveis calculadas e as variáveis experimentais) é projetado através do centro de gravidade dos vértices remanescentes. Melhores (menores) valores da função objetivo são obtidos substituindo, sucessivamente, o ponto com maior valor de $f(x)$ por pontos melhores, até obter o mínimo de $f(x)$ (BONOMI; SCHMIDELL, 2001). Em suma, a ideia básica no método dos poliedros flexíveis é deformar, a cada iteração, um poliedro, de modo que este caminhe em uma direção descendente. Outras operações realizadas pelo método, além da reflexão, são: a expansão, a redução e a contração, as quais servem para deformar o poliedro em cada iteração. Todas essas quatro operações realizadas pelo método possuem coeficientes - α , γ , β e δ - respectivamente para a reflexão, expansão, contração e redução. Esses coeficientes servem para incremento dos pontos para a iteração seguinte. Piccoli *et al.* (1995) estudaram o comportamento dos coeficientes α , γ e β em modelos com 9, 13 e 24 parâmetros ajustáveis. Os valores $\alpha=1,0$, $\gamma=1,5$ e

$\beta=0,8$ resultaram em menores valores das funções objetivo para os modelos propostos.

❖ Métodos de busca indireta ou derivativos

Entre as principais técnicas derivativas ou de busca indireta destacam-se os métodos de Gauss-Seidel ou Newton, do Gradiente ou *Steepest Descent*, de Marquardt, Quasi-Newton e *Line Search*, descritos a seguir.

○ Método de Gauss-Seidel ou Newton

Este método baseia-se na linearização do modelo ao redor de um ponto correspondente a uma estimativa do conjunto de parâmetros, com o objetivo de fazer uso de análises lineares e alcançar o mínimo via procedimento iterativo (HIMMELBLAU, 1970).

○ Método do Gradiente ou *Steepest Descent*

O método do gradiente baseia-se na linearização da função objetivo ao redor de um ponto correspondente ao conjunto de parâmetros. A direção de busca é determinada pelo negativo do vetor gradiente da função objetivo (HIMMELBLAU, 1970).

○ Método de Marquardt

O método de Marquardt associa as principais virtudes do método do gradiente com o de Gauss-Seidel. O método do gradiente apresenta uma convergência rápida quando a solução está longe do ótimo, mas muito lenta quando próximo. Por outro lado, o de Gauss-Seidel apresenta uma convergência rápida próximo ao ponto ótimo (MARQUARDT, 1963). Entretanto, o método dos poliedros flexíveis tem se mostrado mais efetivo que Marquardt, quando o número de parâmetros a serem estimados é muito grande (AUGUSTO; BONOMI; GIUDICI, 1994).

○ Método Quasi-Newton

O método Quasi-Newton é uma generalização do método das secantes para a determinação da raiz da primeira derivada em problemas multidimensionais. Este método é utilizado para encontrar máximos e mínimos locais de uma função através

da localização do ponto onde o gradiente é zero (MATHWORKS, 2010b; SCHOENBERG, 2001; ANDREW, 2008). A matriz Hessiana para um problema sem restrição é uma matriz que contém a segunda derivada da função objetivo. Para calcular a matriz Hessiana existem vários métodos como o BFGS (Broyden, Fletcher, Goldfarb, Shanno) (BROYDEN, 1970; FLETCHER, 1970; GOLDFARB, 1970; SHANNO, 1970) e o DFP (Davidon, Fletcher, Powell) (DAVIDON, 1959; FLETCHER; POWELL, 1963).

- **Método *Line Search***

O método *Line Search* é um método de busca utilizado como parte de um algoritmo mais amplo de otimização. A cada passo do algoritmo principal, o *Line Search* faz uma busca ao longo da linha que contém o ponto corrente seguindo a direção de busca determinada pelo algoritmo principal, em busca do ponto com o menor valor de função objetivo (MATHWORKS, 2010b; SCHITTKOWSKI, 2012; SCHOENBERG, 2001; GROSAN; ABRAHAM, 2007; WALTZ *et al.*, 2006).

- ❖ **Métodos de busca de direção viável**

Os métodos de direção viável caracterizam-se pela movimentação de um ponto viável para outro ponto mais próximo do ponto ótimo. Um dos principais métodos de direção viável é o método de programação quadrática sequencial (SQP - *Sequential Quadratic Programming*). Segundo Mathworks (2010b), o método SQP representa o estado da arte entre os métodos de programação não linear. Em Schittkowski (1985), por exemplo, foi implementada e testada uma versão que supera qualquer outro método testado em termos de eficiência, precisão e porcentagem de soluções com sucesso, entre um grande número de problemas testados. O algoritmo de programação quadrática sequencial (SQP), também é conhecido como programação quadrática iterativa (*Iterative Quadratic Programming*) e programação quadrática recursiva (*Recursive Quadratic Programming*), possuem este nome, pois resolve um problema de programação quadrática (QP - *Quadratic Programming*) a cada iteração. Em cada iteração é feita uma aproximação da matriz Hessiana da função Lagrangeana utilizando o método Quasi-Newton. Então, isto é usado para gerar um subproblema de programação quadrática (QP) cuja solução determina a direção de busca para o procedimento *Line Search*. Neste método, a função objetivo é

aproximada por uma função quadrática e as restrições por funções lineares (MATHWORKS, 2010b). Uma descrição mais detalhada deste método pode vista em: EDGAR; HIMMELBLAU (1988); NOCEDAL; WRIGHT (2006); FLETCHER (1987); GILL; MURRAY; WRIGHT (1981); POWELL (1983); HOCK; SCHITTKOWSKI (1983).

- **Classificação dos problemas de otimização**

Os problemas de otimização podem ser classificados de acordo com dois critérios: rigidez e escala. Esses critérios são importantes para a escolha do método mais adequado para a solução de um problema.

Problemas rígidos (*stiff problems*) são de difícil determinação, mas podem ser mais bem reconhecidos pelo seu comportamento quando aproximados por métodos numéricos. Apesar de a solução ser extremamente estável, a solução numérica pode ser extremamente instável. Não existe uma definição formal do termo rigidez (*stiffness*), mas a ideia principal é que as equações diferenciais possuem alguns termos que levam a uma rápida variação na solução. Devido a este comportamento, as equações diferenciais podem ser divididas em rígidas e não rígidas, cada qual com métodos específicos para a sua solução (RATTENBURY, 2005).

- **MATLAB *Optimization Toolbox***

O *toolbox* de otimização do MATLAB provê os algoritmos mais utilizados para a otimização em problemas de média e larga escala. Estes algoritmos resolvem problemas contínuos (com ou sem restrição) e discretos. Também possui funções para resolver problemas de programação linear, programação quadrática, otimização não linear, mínimos quadrados não lineares, sistemas de equações não lineares e outros (MATHWORKS, 2010b).

Este toolbox possui quatro categorias de funções: *minimizers*, *Multiobjective minimizers*, *Equation Solvers* e *Least-Square Solvers*.

Neste trabalho, serão estudadas apenas as funções *fminunc* e *fminsearch*, utilizadas para a otimização não linear sem restrição, e a função *fmincon*, utilizada para a

otimização não linear com restrição, pois são as que se mostraram mais adequadas à solução do tipo de problema que envolve a modelagem deste trabalho.

❖ **Função *fminunc***

O algoritmo de larga escala da função *fminunc* é baseado no conceito de região confiável (*trust region*). Os algoritmos de média escala utilizados na função *fminunc* são Quasi-Newton e Line Search, já descritos anteriormente. As funções também utilizam uma matriz Hessiana aproximada, se a mesma não for fornecida. Sempre que possível, deve-se fornecer os gradientes e as matrizes Hessianas para diminuir o tempo de processamento e tornar o resultado mais preciso (MATHWORKS, 2010b).

❖ **Função *fminsearch***

A função *fminsearch* utiliza o algoritmo de Nelder-Mead que é um método simplex geométrico flexível, também conhecido como método dos Poliedros Flexíveis sem restrição. Este algoritmo foi proposto inicialmente em 1965 por Nelder e Mead (NELDER; MEAD, 1965) e revisto por Himmelblau (HIMMELBLAU, 1970; HIMMELBLAU, 1972). Este método também foi discutido em diversos artigos (LAGARIAS *et al.*, 1998; OLSSON; NELSON, 1975; QUINTANA *et al.*, 1988). A função *fminsearch* implementa este algoritmo como descrito em Lagarias *et al.* (1998) e adota os valores padrões para os coeficientes α , γ , β e δ (respectivamente, 1, 2, $\frac{1}{2}$ e $\frac{1}{2}$).

❖ **Função *fmincon***

A função *fmincon* permite a escolha de quatro algoritmos diferentes: *Trust-Region-Reflective*, *Active-Set*, *SQP* e *Interior-Point*.

○ **Algoritmo *Trust-Region-Reflective***

O algoritmo *Trust-Region-Reflective* é de larga escala, mas também pode manipular de forma eficiente problemas de média escala. Para a utilização deste algoritmo é necessário que um gradiente seja fornecido. Este algoritmo é baseado no conceito

de região confiável (*trust region*) já descrito anteriormente (MATHWORKS, 2010b; BYRD; SCHNABEL; SHULTZ, 1988; STEIHAUG, 1983; COLEMAN; LI, 1996; CONN; GOULD; TOINT, 2000; BYRD; GILBERT; NOCEDAL, 2000; MORÉ; SORENSEN, 1983; WALTZ *et al.*, 2006).

- **Algoritmos SQP e Active-Set**

O algoritmo SQP pode ser utilizado apenas para solução de problemas de média escala. Tanto a opção *active-set* quanto a opção *sqp* da função *fmincon* utilizam o método SQP para solucionar os problemas propostos. O *sqp* é uma reformulação do *active-set* mais robusta e eficiente, disponível apenas nas versões mais recentes do MATLAB (MATHWORKS, 2010b; GOCKENBACH, 2003; SCHITTKOWSKI, 2012; LEYFFER; MAHAJAN, 2012; BOGGS; TOLLE, 1995).

- **Algoritmo Interior-Point**

O algoritmo *Interior-Point* é de larga escala, mas manipula problemas esparsos e de larga escala tão bem quanto os problemas de menor densidade. O método utilizado para resolver problemas de minimização com restrição é resolver uma sequencia de problemas de minimizações aproximadas (MATHWORKS, 2010b; LEYFFER; MAHAJAN, 2012; BYRD; GILBERT; TOINT, 2000; BYRD; HRIBAR; NOCEDAL, 1999; MEHROTRA, 1992; ZHANG, 1995). Ao contrário de algoritmo simplex, que apenas evolui ao longo de pontos na fronteira da região factível, métodos de ponto interior podem se mover pelo interior da região factível.

- ❖ **Outro método de otimização com restrição implementado**

Além dos métodos disponíveis outros métodos têm sido implementados em diversas linguagens como FORTRAN. Destaca-se aqui o Método dos Poliedros Flexíveis com restrição implementado em FORTRAN.

O método dos Poliedros Flexíveis foi concebido inicialmente para trabalhar com problemas de otimização sem restrição. Porém, em determinadas situações, os parâmetros encontrados no ajuste podem se distanciar muito dos valores iniciais, proporcionando resultados incoerentes.

Em Bonomi *et al.* (1999) foi feita uma implementação deste método, em linguagem FORTRAN, que impõe limites superiores e inferiores para cada parâmetro, e utilizado em vários trabalhos de pós-graduação (FERRAZ, 1999; PICCOLI, 2000; TAVARES, 2005; CASTILLO, 2002; BARRAL, 2010). Nessa implementação, a fórmula de cálculo de resíduos na função objetivo é a soma do quadrado da diferença dos valores calculados e experimentais normalizados pela média entre o valor calculado e o experimental em cada ponto. Como essa fórmula resulta em resíduos elevados para valores muito pequenos e diferentes das variáveis calculada e experimental, é necessário aplicar uma restrição: o resíduo é calculado apenas quando o valor experimental é superior a um valor mínimo fixado previamente: $y_{experimental\ i} > \varepsilon (y_{máximo\ experimental\ i})$. Habitualmente, esse valor de ε é arbitrado como 10^{-6} . No algoritmo de ajuste, o método de resolução de equações diferenciais ordinárias não lineares utilizado foi o Método de Gear da rotina Sdriv2 (KAHANER, 1988).

3.7.9 Análise Estatística

O ajuste dos parâmetros do modelo proposto a um conjunto de ensaios experimentais é normalmente avaliado e considerado satisfatório ou não, por simples inspeção visual do conjunto de ensaios, além da análise dos resíduos (BONOMI; SCHMIDELL, 2001). Apesar dessa constatação, é importante submeter os ajustes obtidos a uma análise estatística específica.

Em um primeiro momento, calcula-se o coeficiente angular (CA) e o coeficiente de correlação (R^2) do gráfico que apresenta os valores calculados pelo modelo em função dos valores experimentais normalizados, construído com todos os pontos usados no ajuste. Os valores obtidos são comparados com os valores ótimos (CA=1 e $R^2=1$) (BARRAL, 2010).

A análise estatística que deve ser feita para se garantir que o modelo obtido represente satisfatoriamente o conjunto de ensaios ajustados é o Teste F, que se baseia na obtenção do chamado "erro experimental". Essa estimativa do "erro experimental" deve levar em conta, entre outros, a falta de reprodutibilidade dos processos biotecnológicos, a dificuldade em manter condições homogêneas dentro

do reator e os próprios erros analíticos e de amostragem comuns na atividade laboratorial (BONOMI; SCHMIDELL, 2001).

Para aplicar o Teste F, é calculada a estimativa do erro do modelo ajustado. Calcula-se a estimativa da variância do erro do modelo (s_c^2) comparando os valores das variáveis de estado obtidas pelo modelo ajustado em relação aos dados experimentais disponíveis, conforme a Equação 12.

$$s_c^2 = \frac{\sum_{j=1}^n (\tilde{y}_{ij} - y_{ij})^2}{(nv)_c - p} \quad (12)$$

onde:

$\tilde{y}_{ij}$ é o valor da variável calculada pelo Modelo;

y_{ij} é o valor experimental da variável;

$(nv)_c$ é o número de pontos ajustados;

p é o número de parâmetros do Modelo.

Como não se dispõe de uma medida precisa da estimativa do erro experimental, uma vez que não foram obtidas repetições de um mesmo ensaio, a partir das quais esta estimativa seria obtida, a aplicação do Teste F será modificada, calculando-se o erro experimental que, se existente, garante que o modelo ajustado representa adequadamente os dados experimentais disponíveis (BONOMI; SCHMIDELL, 2001).

Para tanto, obtém-se, a partir dos valores das variáveis de estado medidas experimentalmente (na mesma unidade), a somatória do quadrado de todas elas, conforme a Equação 13.

$$s_e^2 = \frac{\left(\sum_{i=1}^n (Xv_{\text{exp}})^2 + \sum_{i=1}^n (GLC_{\text{exp}})^2 + \sum_{i=1}^n (GLN_{\text{exp}})^2 \dots \right) \varepsilon^2}{(nv)_c - v} \quad (13)$$

onde:

s_e^2 é a estimativa da variância do erro experimental;

v é número de variáveis de estado

Se $s_c^2 < s_e^2$, ou seja, a estimativa da variância do erro do modelo tem que ser menor que a estimativa da variância do erro experimental para que o modelo ajustado seja adequado.

4 MATERIAIS E MÉTODOS

4.1 ATIVIDADES DESENVOLVIDAS

Figura 9 - Diagrama de blocos da metodologia utilizada no desenvolvimento deste trabalho.

A metodologia deste trabalho seguiu uma rotina, conforme ilustrado na Figura 9. Inicialmente, foi desenvolvido o *software* TrataDados para o tratamento dos dados cinéticos de cultivo experimental utilizados na modelagem. Em seguida, foi desenvolvido o *software* AjustaPar para o ajuste de parâmetros da modelagem. Foi feito então, uma análise de todos os ensaios executados durante o Projeto Temático, para que se tivesse um panorama geral das condições de cultivo e do comportamento da célula frente às variáveis. A seguir, foram propostos vários modelos de vias metabólicas e estes foram analisados através do programa Metatool que forneceu os Modos Elementares. As macro-reações advindas dos modos elementares foram equacionadas, por equações cinéticas de Michaelis-Menten, que com a utilização do programa AjustaPar chegou-se a um conjunto de

parâmetros, fazendo-se, então, uma simulação de comportamento cinético (através de um programa de simulação também desenvolvido pela autora). Essa sequencia de trabalho seguiu-se até que se chegasse ao melhor modelo.

Na Tabela 2, apresentam-se as atividades desenvolvidas durante este trabalho, as ferramentas utilizadas e correspondentes objetivos.

Tabela 2 - Atividades desenvolvidas neste trabalho

Atividades Desenvolvidas	Ferramentas Utilizadas	Objetivos
TrataDados	MATLAB	Desenvolvimento de um <i>software</i> para padronizar e facilitar o tratamento de dados experimentais.
AjustaPar	MATLAB	Desenvolvimento de um <i>software</i> como ferramenta para o ajuste de parâmetros em uma modelagem.
Análise de dados Experimentais	Resultados dos ensaios demais membros do grupo	Levantamento geral de todos os ensaios do grupo do Projeto Temático. Esses ensaios foram utilizados para análise e fundamentação da proposição de vias metabólicas e também para validação do modelo final.
Modelagem Metabólica	Metatool	Proposições de conjuntos de vias metabólicas se chegar a uma resposta, através da modelagem matemática, que corresponda aos dados experimentais obtidos.
Modelagem Matemática	MATLAB e simulador do comportamento cinético da S2AcGPV	Equacionamento dos modos elementares obtidos pelo Metatool para simulação do comportamento cinético.

4.2 FERRAMENTAS COMPUTACIONAIS UTILIZADAS

As ferramentas utilizadas para a execução desta modelagem estão descritas a seguir.

4.2.1 MATLAB

O MATLAB (MATrix LABoratory), desenvolvido atualmente pela empresa MathWorks, é uma ferramenta computacional que inclui uma linguagem de programação de alto nível e um ambiente interativo que podem ser utilizados para desenvolvimento de algoritmos, visualização de dados, análise de dados e computação numérica. O MATLAB possui aplicações em diversas áreas, incluindo processamento de sinais e imagens, comunicações, sistemas de controle e aquisição de dados, estatística e análise de dados, cálculo numérico e otimização, análise numérica, análise financeira, computação distribuída etc. (MATHWORKS, 2009). O MATLAB foi utilizado inicialmente na versão 7.4 (R2007a) e mais recentemente foram utilizados os novos recursos do MATLAB 7.10.0 (R2010a).

4.2.2 Metatool

O software Metatool é uma implementação eficiente da análise de modos elementares, é um programa de linhas de comando e não traz o conforto e usabilidade de uma interface gráfica de usuário (GUI), porém traz o princípio puro de transformar complexas redes metabólicas em modos elementares. Ele é utilizado como auxiliar ao entendimento das principais reações metabólicas que acontecem no interior das células.

O desempenho do Metatool foi melhorado significativamente e a versão 5.1 pode ser executada dentro do MATLAB, com um ambiente que permite o uso mais flexível e a integração com outras ferramentas (KAMP; SCHUSTER, 2006). Nas versões prévias do Metatool, o algoritmo descrito por Schuster; Feel e Dandekar (2000) foi usado para calcular as equações elementares. Em contrapartida, a versão atual está baseada no algoritmo proposto por Urbanczik e Wagner (2005), o qual empiricamente mostrou um alto desempenho. O Metatool foi utilizado na versão 5.1 para MATLAB 7 ou superior.

4.2.3 KEGG

O KEGG (*Kyoto Encyclopedia of Genes and Genomes*) é um banco de dados utilizado para a compreensão de funções de alto nível e utilitários do sistema biológico, como o celular, o organismo e o ecossistema, a partir de informações em nível molecular, especialmente em grande escala, de conjuntos de dados gerados pelo sequenciamento do genoma e outros de elevada importância para as tecnologias experimentais. Ele utiliza esse banco de dados para hierarquizar vias metabólicas e vias regulatórias do metabolismo do organismo em questão. Ou seja, através do genoma, o KEGG apresenta todas as possíveis vias metabólicas com as enzimas que podem ser codificadas. O KEGG pode ser acessado em: <http://www.genome.jp>. e suas informações estão disponíveis *on line*.

4.3 ORIGEM DOS DADOS EXPERIMENTAIS

As amostras dos ensaios utilizadas para a formulação do modelo cinético do cultivo de células S2 recombinantes de *Drosophila melanogaster* para a expressão da glicoproteína do vírus da raiva, foram obtidas e tratadas em Aguiar (2010). Abaixo se encontram resumidas as metodologias empregadas nesse estudo.

4.3.1 Células

Os ensaios considerados utilizaram células S2 de *Drosophila melanogaster* transfectadas com o vetor pAcGPVHygro para expressar a glicoproteína do vírus da raiva (GPV), segundo procedimento descrito por Yokomizo *et al.* (2007) e denominadas S2AcGPV. As células foram adaptadas em meio TC100 (Vitrocell, Campinas, SP, Brasil) suplementado com 8,9 g/L de glicose (Sigma, EUA), 2,9 g/L de glutamina (Sigma,EUA), 0,3% (g/g) de Pluronic F68 (Sigma, EUA), 3 g/L de yeastolate (Extrato de leveduras ultrafiltrado, Gibco BRL, Carlsbad, CA, EUA) e 1% (v/v) de emulsão lipídica (Gibco BRL,EUA) (GALESI *et al.*, 2008). Para os ensaios

em reator (que serão os utilizados para a modelagem neste trabalho) a concentração de Pluronic F68 foi aumentada para 0,6% (g/g) e, dependendo da condição estudada, as concentrações de glicose e glutamina também sofreram alterações.

4.3.2 Sistema de Cultivo

Foi utilizado biorreator Biostat MD/CPU (B. Braun, Alemanha) com dorna de 3 L (1,5 L de volume útil), agitado a 90 rpm por dois impelidores tipo *pitched blade* com duas pás e a transferência de oxigênio foi feita por difusão através de membrana de silicone (SI-Schlauch, Alemanha) com as seguintes dimensões: 3 mm de diâmetro externo, 0,35 mm de espessura e 5 m de comprimento. O biorreator dispunha de controle automático de: a) temperatura (28°C) e b) oxigênio dissolvido (variável, ver item 4.5.2), com o controle feito mediante uso de eletrodo polarográfico (Mettler-Toledo, modelo A12/220, Columbus, OH, EUA) para monitoramento e um misturador de gases (B. Braun Biotech- Gas Mix Unit, Alemanha). O ar insuflado era uma mistura de ar (compressor BarionKar, modelo EL-20157G, Brasil), nitrogênio ultrapuro (White Martins, Rio de Janeiro, RJ., Brasil) e oxigênio ultrapuro (White Martins, Brasil).

4.3.3 Conjunto de ensaios para a modelagem

Foi utilizada para a modelagem matemática das células rS2 o conjunto de ensaios conforme Tabela 3. Este conjunto de ensaios possui variadas condições iniciais de glicose, glutamina e oxigênio dissolvido.

Tabela 3 – Ensaios para a modelagem

Ensaio	Nome do Ensaio (AGUIAR, 2010)	Glicose inicial (g/L)	Glutamina inicial (g/L)	OD (% saturação em ar)
1	Ma_07	10	3,5	5
2	Ma_08	10	3,5	50
3	Ma_09	10	3,5	80
4	Ma_10	10	3,5	30
5	Ma_11	1	3,5	30
6	Ma_12	5	3,5	30
7	Ma_14	15	3,5	30
8	Ma_15	10	0,6	30
9	Ma_16	10	1,75	30
10	Ma_17	10	7,0	30

4.3.4 Metodologias

4.3.4.1 Concentração celular e viabilidade

A concentração de células foi determinada, através da média da contagem em duas câmaras de Neubauer (FRESHNEY, 1994), utilizando microscópio ótico. Amostras contendo células eram diluídas em azul de trypan 0,05%.

4.3.4.2 Dosagem de glicose

As amostras, armazenadas a -20 °C, foram descongeladas à temperatura ambiente e desproteinizadas usando solução de ácido tricloroacético (TCA) na concentração de 240 g/L, numa proporção de 4:1 e centrifugadas a 10.000 rpm, por 10 minutos a temperatura ambiente. O sobrenadante foi filtrado com membrana com poros de 0,45 µm antes e analisadas por cromatografia líquida (HPLC) por troca iônica.

4.3.4.3 Dosagem de ácido láctico

As amostras foram desproteinizadas como descrito no item 4.3.4.2 e analisadas por cromatografia líquida (HPLC) por troca iônica.

4.3.4.4 Dosagem de aminoácidos

A medida de concentração de aminoácidos foi realizada através de cromatografia líquida de alta resolução (HPLC) e a metodologia utilizada foi a de derivatização pós-coluna, a 130 °C, com ninidrina.

4.3.4.5 Dosagem de amônio

Para a determinação de amônio, foi utilizado um potenciômetro e um eletrodo específico que detecta variações de milivoltagem quando ocorre a conversão de amônio em amônia, pela adição de NaOH.

4.3.4.6 Dosagem da glicoproteína do vírus da raiva (GPV)

A detecção da glicoproteína do vírus da raiva (GPV) foi realizada por teste do tipo ELISA (*Enzyme-Linked Immuno Sorbent Assay*), empregando o kit *Rabies Glycoprotein Enzyme Immunoassay* (Instituto Pasteur, França), conforme descrito por Astray *et al.* (2008).

4.3.5 Demais ensaios para simulação do modelo

Utilizando-se de todos os ensaios gerados em biorreator pelo grupo do Projeto Temático, foi feita uma seleção de quais ensaios, em adição aos acima descritos, poderiam ser utilizados para simulação da modelagem final. Chegou-se a um conjunto de ensaios que são descritos em Batista (2007), Pamboukian *et al.* (2008) e Ventini (2010).

4.4 SEQUÊNCIA DE TRABALHO PARA A MODELAGEM

Para a execução da modelagem matemática, procedeu-se aos seguintes passos:

- proposição de vias metabólicas para representar o comportamento cinético da linhagem;
- definição dos modos elementares correspondentes a essas vias metabólicas aplicando o software Metatool;
- equacionamento desses modos elementares através de equações cinéticas diferenciais;
- resolução das equações diferenciais e determinação de parâmetros do modelo em *software* desenvolvido na linhagem de programação MATLAB;
- simulação do modelo ajustado calculando os resíduos resultantes (diferença entre a comparação dos dados experimentais e simulados);
- avaliação dos resultados e proposição de nova modelagem metabólica.

Foram propostos 5 modelos metabólicos básicos e dentro de cada proposição foram feitas várias alternativas aos fluxos metabólicos até se encontrar o conjunto que melhor respondesse ao problema apresentado.

5 RESULTADOS E DISCUSSÃO

5.1 TrataDados

O *software* TrataDados foi desenvolvido durante este trabalho, em MATLAB, para o tratamento de dados cinéticos de cultivo de células animais e que também pode ser adaptado para cultivos de micro-organismos. Visa facilitar a análise por diferentes profissionais (engenheiros, biólogos, químicos, farmacêuticos, etc.) e inclui os resultados e análises envolvidos em um processo biotecnológico como:

- efetuar a leitura de dados experimentais de variáveis de estado diretamente de uma planilha Excel;
- efetuar o ajuste de curva suavizada para cada variável de estado, utilizando diferentes métodos: *spline*, *smoothing spline* e Hermite, sendo que no *smoothing spline* pode ser definido o coeficiente p de suavização;
- efetuar o ajuste de curva utilizando os recursos disponíveis para se obter um melhor alisamento, como:
 - remover pontos experimentais (claramente errados, para melhorar o ajuste da curva);
 - inserir pontos experimentais auxiliares para melhorar o ajuste da curva;
 - mover pontos experimentais, gerando novos pontos auxiliares, para melhorar o ajuste da curva;
- determinar graficamente os instantes que são importantes durante o crescimento celular como início da fase exponencial, término da fase exponencial, tempo final de crescimento, tempo de início da morte e outros (adaptado de Aguiar (2010));
- efetuar um pré-ajuste dos pontos da fase exponencial e utilizá-los no ajuste da curva de μ_x de forma a caracterizar tal fase com maior precisão (adaptado de Aguiar (2010));
- elaborar o gráfico de $\ln X_v$, calcular o coeficiente de correlação para a reta ajustada à fase exponencial e determinar o valor de $\mu_{x_{max}}$;
- durante a edição da curva referente a uma variável de estado, permitir a visualização do gráfico normalizado de outras variáveis;
- calcular as velocidades específicas de consumo de substratos;

- efetuar o cálculo dos fatores de conversão, com possibilidade de ajuste de pontos e elaboração de gráficos;
- salvar os dados referentes ao tratamento dos dados em uma planilha Microsoft Excel;
- salvar os gráficos gerados no formato bmp.

Com a utilização desse aplicativo, tem-se menos tempo despendido para o tratamento de dados e também uma padronização de resultados, obtendo melhores resultados em sua análise, apresentando as informações de maneira mais clara.

Toda a descrição do *software* e sua sequência de utilização encontram-se detalhados no Apêndice A. A listagem completa do programa está no Apêndice E.

Outra aplicabilidade deste *software* é a sua utilização em aulas para estudar-se a atuação de ensaio com diversas variações para a curva da velocidade específica de crescimento celular.

5.2 AjustaPar

Como visto anteriormente, um modelo matemático é formulado por um conjunto de equações cinéticas e diferenciais não lineares que representam a dinâmica de todas as variáveis de estado ao longo do processo. Assim, para simular o modelo, que é constituído de parâmetros, variáveis de estado e equações cinéticas e diferenciais, é necessário atribuir valores a esses parâmetros (AUGUSTO; BARRAL; PICCOLI, 2008).

O *software* AjustaPar foi desenvolvido durante este trabalho, em MATLAB, para ser utilizado como ferramenta básica para o ajuste de parâmetros utilizados na modelagem matemática de bioprocessos. O ajuste pode ser feito utilizando diversos métodos que foram descritos no Capítulo 3 item 3.7.8.

Nesta programação, utilizaram-se sete métodos para a solução das Equações Diferenciais Ordinárias, dentre eles o método de Runge-Kutta de quarta e quinta ordem. Existem, à disposição do usuário, 8 fórmulas diferentes para o cálculo de resíduo para que se faça a escolha daquela que tem a melhor resposta frente a cada problema, conforme descrito no Capítulo 3 item 3.7.8.3. Para a otimização da

busca de parâmetros têm-se as seguintes rotinas em MATLAB (descritas no Capítulo 3 item 3.7.8.4) *fminunc*, *fminsearch* (Poliedros Flexíveis sem restrição) e *fmincon* (*Active-set*; *Interior-point* e *SQP*) e ainda o método dos Poliedros Flexíveis com restrição implementado em MATLAB.

Como o MATLAB não apresenta a função Poliedros Flexíveis com restrição, essa função foi implementada pelo autor desta tese baseado no código disponível em ZHU; CHEN (2000). O código foi modificado para acrescentar uma restrição de limites (máximo e mínimo) para cada parâmetro. Com esta modificação, o código implementado passou a encontrar ótimos resultados. Esse método foi denominado como 'pamboukian' para identificação no AjustaPar.

O processo de ajuste é muito parecido para todos os métodos: o método de busca, a cada iteração, estima um conjunto de parâmetros e faz diversas avaliações da função objetivo até que seu valor (resíduo) esteja dentro do critério imposto, ou seja, o que se tenta sempre é a minimização do resíduo. O programa, por sua vez, utiliza a função *ode45*, entre outras funções a escolha do usuário, para resolver o sistema de equações diferenciais para os parâmetros correntes e calcula os resíduos através da fórmula escolhida pelo usuário. Esse processo ocorre até que se chegue a um dos critérios estabelecidos: tolerância, precisão, quantidade de rodadas, quantidade de avaliações da função objetivo ou quando o resíduo não apresentar mais variações significativas.

Ao término do ajuste, são exibidas as informações relativas ao ajuste e os parâmetros encontrados.

Toda a descrição do *software* e a sequência de utilização encontram-se detalhadas no Apêndice B. A listagem completa do programa está no Apêndice F.

5.2.1 Comparação entre dois Programas de Ajuste de Parâmetros

Conforme visto anteriormente, um programa de nome AjuPar, implementado em FORTRAN por Bonomi *et al.* (1999), tem sido utilizado em vários trabalhos de pós-graduação (FERRAZ, 1999; PICCOLI, 2000; TAVARES, 2005; CASTILLO, 2002; BARRAL, 2010). Esse programa utiliza o método de Poliedros Flexíveis com

restrição. Como neste trabalho foi implementado, em MATLAB, o programa AjustaPar e dentre os métodos de busca disponíveis tem-se, também, o método dos Poliedros Flexíveis com restrição, foram feitos estudos com ambos os programas considerando dados simulados com as equações cinéticas e valores dos parâmetros apresentados na Tabela 4 (denominados "parâmetros originais") para verificar se os resultados dos ajustes de parâmetros se assemelham. Portanto, o programa AjuPar foi utilizado para validar o programa AjustaPar e fazer uma comparação das diferenças entre o comportamento dos dois programas. Em ambos os programas, utilizou-se o método dos poliedros flexíveis com restrição e a Fórmula 6 (Tabela 1) para cálculo dos resíduos.

Tabela 4 – Equações cinéticas e parâmetros originais

Equações Cinéticas	Parâmetros	
	Originais (Valor de referência)	Unidade
$\mu_X = \mu_{Xm} \left(\frac{GLN}{GLN + K_{GLN}} \right) \left(\frac{K_{NH4}}{NH4 + K_{NH4}} \right) \left(\frac{K_{LAC}}{LAC + K_{LAC}} \right)$	(1) $\mu_X = 0,055$	h^{-1}
	(2) $K_{GLN} = 2,65$	mg/L
$\frac{dX}{dt} = \mu_X * X$	(3) $K_{NH4} = 160$	mg/L
	(4) $K_{LAC} = 3$	g/L
$\frac{dGLC}{dt} = - \left(\frac{\mu_X * X}{Y_{X/GLC}} \right) - \left(\frac{\delta * \mu_X * X}{Y_{LAC/GLC}} \right)$	(5) $\delta = 0,55$	mg/mg
	(6) $Y_{X/GLC} = 0,75$	10^3 cel/g
$\frac{dGLN}{dt} = - \left(\frac{\mu_X * X}{Y_{X/GLN}} \right) - \left(\frac{\delta_1 * \mu_X * X}{Y_{NH4/GLN}} \right)$	(7) $Y_{LAC/GLC} = 2.10^{-3}$	10^3 cel/g
	(8) $\delta_1 = 0,021$	mg/mg
$\frac{dP}{dt} = \alpha * \mu_X * X + \beta * X$	(9) $Y_{X/GLN} = 8,67$	10^3 cel/mg
	(10) $Y_{NH4/GLN} = 0,39$	10^3 cel/mg
$\frac{dLAC}{dt} = \alpha_1 * \mu_X * X + \beta_1 * X$	(11) $\alpha = 1,13$	$g/10^3 \text{ cel}$
	(12) $\beta = 0,31.10^{-3}$	$g/10^3 \text{ cel.h}$
$\frac{dNH_4}{dt} = \alpha_2 * \mu_X * X + \beta_2 * X$	(13) $\alpha_1 = 52$	$g/10^3 \text{ cel}$
	(14) $\beta_1 = 0,5.10^{-3}$	$g/10^3 \text{ cel.h}$
	(15) $\alpha_2 = 0,52$	$mg/10^3 \text{ cel}$
	(16) $\beta_2 = 7,05$	$mg/10^3 \text{ cel.h}$

Foram feitas 5 rodadas consecutivas em cada programa, ou seja, os resultados dos parâmetros ajustados pela primeira rodada foram utilizados como estimativa inicial

em uma segunda rodada dos programas (cada programa utilizando os resultados gerados pelo próprio programa) e assim sucessivamente até a quinta rodada. Foi feita então uma relação entre os parâmetros originais (valor de referência), e os parâmetros obtidos pelos programas AjuPar e AjustaPar, apresentada a seguir.

5.2.1.1 Estudo Comparativo 1

As estimativas iniciais precisam ser geradas através do estudo de cada parâmetro, tendo como referência os dados originais. Como aqui a intenção era a comparação da resposta gerada pelos programas, foram utilizadas estimativas iniciais tomando como base os valores originais e utilizando-se valores da mesma ordem de grandeza desses valores (valores próximos aos originais) e para o limite superior e inferior de 20 vezes cada. Assim, o gráfico das variações entre os valores originais de parâmetros e aqueles obtidos pelos dois programas (AjuPar e AjustaPar), neste primeiro comparativo com as 5 rodadas com cada programa, está na Figura 10.

Figura 10 – Comparativo 1 entre os valores dos parâmetros gerados pelos ajustes dos programas AjustaPar e AjuPar

Na Figura 11 são apresentados os gráficos resultantes da última rodada com cada programa com os valores dos dados originais (símbolos) e os dados simulados (linha) a partir dos parâmetros gerados pelos ajustes obtidos com o AjuPar e AjustaPar (5ª rodada).

Observa-se que os programas (a partir das mesmas sementes) obtêm valores similares de parâmetros, com a exceção dos parâmetros 5 e 6 que no programa AjuPar obtiveram erros maiores (Figura 10). Porém, na avaliação visual dos gráficos gerados, com os dados originais e os dados simulados, a semelhança é grande e somente os gráficos de glicose e lactato apresentaram pequenas diferenças.

A avaliação destes resultados está centrada em dois aspectos:

1. A comparação dos programas AjuPar e AjustaPar, pois se queria ter um novo programa de Ajuste de Parâmetros e o método dos Poliedros Flexíveis com restrição utilizado pelo AjuPar há muito tempo utilizado pelo grupo do IPT, se mostra com bons resultados;
2. A capacidade do método dos poliedros flexíveis com restrição de reproduzir os parâmetros originais. Em alguns casos o poliedros não é capaz de retornar valores de parâmetros semelhantes ao original, como no caso dos parâmetros 3, 4 e 11 como mostra a Figura 10.

a) Gráficos com os ajustes feitos pelo AjuPar

b) Gráficos com os ajustes feitos pelo AjustaPar

Figura 11 – Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 1, com os dados originais (símbolos) e curva calculada pela simulação (linha).

5.2.1.2. Estudo Comparativo 2

Para testar a capacidade dos programas de ajuste de parâmetros em condições de busca mais restritivas, neste segundo comparativo foram utilizadas sementes com valores da mesma grandeza dos parâmetros originais, porém a faixa de busca de valores foi restritiva a valores cujos limites superior e inferior foram de 100 vezes. O gráfico das variações entre os valores originais de parâmetros e aqueles obtidos pelos dois programas (AjuPar e AjustaPar), nas 5 rodadas está apresentado na Figura 12.

Figura 12 – Comparativo 2 entre os valores gerados pelos ajustes dos programas AjustaPar e AjuPar

Observa-se que neste comparativo, os parâmetros 7, 8, 12, 14 e 15 apresentam as maiores diferenças em relação aos parâmetros originais. Isso sugere que uma faixa maior de amplitude dos limites pode dificultar o encontro dos melhores valores e que para se utilizar os poliedros flexíveis o valor da semente inicial é muito importante.

Na Figura 13 são apresentados os gráficos resultantes da última rodada com cada programa com os valores dos dados originais (símbolos) e os dados simulados (linha) com os parâmetros gerados pelos ajustes obtidos no AjuPar e no AjustaPar (5ª rodada).

a) Gráficos com os ajustes feitos pelo AjuPar

b) Gráficos com os ajustes feitos pelo AjustaPar

Figura 13 – Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 2, com os dados originais (símbolo) e curva calculada pela simulação (linha).

Apesar dos parâmetros 7, 8, 12, 14 e 15 terem maiores diferenças entre os valores originais em relação ao comparativo 1, a análise visual dos dados simulados em relação aos dados originais gerados pelos dois programas é similar. Observa-se,

porém, que os valores simulados a partir dos parâmetros obtidos pelo AjustaPar representam melhor os dados originais que aqueles pelo AjuPar.

5.2.1.3. Estudo Comparativo 3

Foi feito então, mais um teste para se ver o comportamento dos programas frente a maiores dificuldades. Para a semente inicial dos parâmetros foram dados valores completamente aleatórios. A faixa de busca de valores foi restritiva a valores cujos limites superior e inferior foram de 1000 vezes maiores e menores que o valor original. Assim, em uma única rodada, tem-se o gráfico mostrado na Figura 14 com dados originais e simulados com os parâmetros gerados pelos ajustes obtidos no AjuPar e AjustaPar.

Como esperado, já que as sementes iniciais foram aleatórias, os valores encontrados não se ajustaram bem, no entanto podemos notar que a resposta observada pelo gráfico para ambos os programas, representa razoavelmente bem os dados originais. Porém, a simulação com os parâmetros advindos do programa AjustaPar está mais próxima dos dados originias que aquela resultante do uso do AjuPar.

a) Gráficos com os ajustes feitos pelo AjuPar

b) Gráficos com os ajustes feitos pelo AjustaPar

Figura 14 – Conjunto de gráficos obtidos após a última rodada dos programas de ajuste no Comparativo 3, com os dados originais (símbolo) e curva calculada pela simulação (linha).

Sendo assim, apesar dos dois programas trabalharem com o mesmo método de otimização para minimização de funções (Poliedros Flexíveis com restrição), o

comportamento do programa AjustaPar, desenvolvido em MATLAB parece convergir para melhores resultados do ajuste de parâmetros.

Para tornar mais claras as diferenças de programação entre AjuPar e AjustaPar, na Tabela 5 são descritas as suas principais características.

A maior diferença entre os dois programas está no método de resolução das Equações Diferenciais Ordinárias, que no caso do AjustaPar foi utilizado Runge-Kutta de 4^a e 5^a ordem, o que pode explicar o diferente comportamento nos comparativos executados acima, e no AjuPar foi utilizado Adams/Gear.

No trabalho com o AjustaPar, pôde ser observado que ele consegue atingir valores muito próximos dos parâmetros originais em uma única rodada, não sendo necessários novos ajustes. Depois de uma rodada, mesmo que se faça essa retroalimentação os valores permanecerão praticamente inalterados, sendo que essa diferença entre uma rodada e outra resulta em 1% (dados não mostrados).

O AjustaPar apresenta ainda maior interação com o usuário e possibilidades de escolha para a otimização. Possui à disposição do usuário 8 fórmulas diferentes para cálculo de resíduos, 7 métodos de resolução das equações diferenciais ordinárias, além de 6 métodos de ajustes de parâmetros disponíveis.

Tabela 5 – Diferenças entre os programas AjuPar e AjustaPar

	AjuPar	AjustaPar
Autor	Bonomi <i>et al.</i> , 1999	Pamboukian
Linguagem de programação	FORTTRAN	MATLAB
Ano	Utilizada versão de 1998	Criado em 2010 e última versão em 2012
Métodos de ajustes de parâmetros	<ul style="list-style-type: none"> • Poliedros Flexíveis com restrição (baseado em HIMMELBLAU, 1972 e modificado por Piccoli <i>et al.</i>, 1995) 	<ul style="list-style-type: none"> • Poliedros Flexíveis com restrição (baseado em ZHU; CHEN, 2000) • <i>fminsearch</i> (poliedros flexíveis sem restrição) • <i>fmincon</i> • <i>fminunc</i>
Método de resolução de equações diferenciais ordinárias	Pré-estabelecida na programação SDRIV2 (Adams/Gear)	7 opções diferentes
Método de cálculo de Resíduo	Pré-estabelecida na programação (Fórmula 6, Tabela 1)	Pode utilizar 8 Fórmulas diferentes (Tabela 1)
Matriz inicial (Poliedros Flexíveis)	Fornecida pelo usuário	Calculada automaticamente
Função Objetivo	Minimização do resíduo	Minimização do resíduo
Critério de parada	Número de iterações; precisão	Número de iterações; precisão; tolerância; quantidade de avaliação da função objetivo
Velocidade de processamento	Lento	Até 60% mais rápido no caso da rotina <i>fmincon</i> .
Aproximação	Rodadas sucessivas	Rodada única
Gráficos	Não fornece	Geração automática
Definição de dados e equações	Através de arquivo texto (.dat)	Através de planilha do Excel
Interação com o usuário	Apenas através de arquivo texto	Através de menus
Quantidade de ensaios, pontos por ensaio, variáveis, parâmetros, etc.	Dimensional	Ilimitada
Normalização de variáveis de estado	Manual	Automática (item 3.7.8.3)

5.2.2 Estudo do comportamento do AjustaPar para a escolha do método de trabalho

Como o AjustaPar forneceu melhores resultados de ajuste como anteriormente mostrado e é um programa que oferece várias possibilidades de escolha de métodos de busca, bem como de resíduo, foi feito um estudo visando a escolha da melhor metodologia a ser aplicada na modelagem matemática deste trabalho.

5.2.2.1. Métodos de minimização do AjustaPar

Para a otimização da busca de parâmetros visando a minimização da função objetivo estão disponibilizadas ao usuário do *software* AjustaPar as seguintes técnicas:

- método *fmincon* (com restrição) - *Active-Set*, *SQP* e *Interior-Point*
- poliedros flexíveis sem restrição (*fminsearch*)
- método *fminunc* (sem restrição)
- poliedros flexíveis com restrição (*pamboukian*)

Utilizando a fórmula 6 para cálculo do resíduo (Tabela 1), as mesmas estimativas iniciais e os mesmos limites superior e inferior, verificaram-se as respostas de cada método. Porém, para o método *fmincon-SQP* com a fórmula 6 o programa para e não consegue seguir adiante, somente com a fórmula 1 conseguiu-se trabalhar (Apêndice C). Para essa análise foi utilizado o ensaio Ma_07 (Capítulo 4 item 4.3.3) de Aguiar (2010) com valor experimental de $\mu_{x_{\max}}$ de $0,05 \text{ h}^{-1}$. No Apêndice C estão descritos todos os testes realizados com os seis métodos de busca disponíveis e na Tabela 6 são apresentados os resultados com a utilização de cada método. A precisão é calculada como sendo a diferença entre o resíduo da última e da penúltima rodada do programa.

Neste comparativo, o método de busca que se mostrou mais eficiente foi o da função *fmincon* do MATLAB utilizando o algoritmo *Interior Point*. Este método foi o que proporcionou maior precisão, alta velocidade de processamento, baixo resíduo, pequena quantidade de iterações, valores coerentes de $\mu_{x_{\max}}$, ótimos resultados representados graficamente (Figura 117) e menor resíduo. Já os métodos das funções *fminsearch* e *fminunc* do MATLAB se mostraram aceitáveis, mas como não

existe definição de limites (sem restrições), podem resultar em valores incoerentes (negativos, por exemplo). O método *fminsearch* do MATLAB e o método dos Poliedros Flexíveis implementado pelo autor deste trabalho se mostraram lentos devido à grande quantidade de iterações necessários para se efetuar o ajuste. Portanto, será utilizado, na modelagem alvo deste trabalho, o método com a função *fmincon* com o algoritmo *Interior Point*.

Tabela 6 – Resultados do estudo com cada método de busca de parâmetro do AjustaPar

	Método	Precisão	Tempo de processamento	Valor do Resíduo	Nº de iterações	Fórmula (Tabela 1)	$\mu_{x_{\max}}$ (h ⁻¹)
fmincon	ActiveSet	O método não conseguiu processar o cálculo					
	Interiorpoint	1,0.10 ⁻⁸	8 s	11,59	125	6	5,1.10 ⁻²
	SQP	4,2.10 ⁻⁷	5 s	2,47	43	1	1,7.10 ⁻¹
fminsearch		8,5.10 ⁻⁵	71 s	11,65	10000	6	9,9.10 ⁻²
pamboukian		7,3.10 ³	120 s	36,30	9000	6	1,2.10 ⁻¹
fminunc		6,3.10 ⁻⁶	90 s	36,11	130	6	2,1

5.2.2.2. Fórmulas para o cálculo de resíduos

Estão à disposição do usuário 8 fórmulas de cálculo de resíduo, descritas no capítulo 3 (Tabela 1). Utilizando-se o ensaio Ma_07 (AGUIAR, 2010) e trabalhando-se com a função *fmincon* com o algoritmo *Interior Point*, verificou-se o resultado apresentado através da utilização de cada uma das fórmulas. No Apêndice C estão descritos todos os testes realizados com as 8 fórmulas disponíveis e na Tabela 7 são apresentados os resultados com a utilização de cada fórmula.

Neste comparativo, as fórmulas 5 e 6 apresentaram os melhores resultados: melhor precisão, ótimos resultados representados graficamente (Figuras 125 e 126), valores de $\mu_{X_{\max}}$ coerentes com os experimentais e melhor resíduo relacionado ao resultado. Utilizam uma quantidade reduzida de iterações, fornecem um resíduo baixo e possuem alta velocidade de processamento. A fórmula 6 possui uma vantagem adicional sobre a fórmula 5. Como a fórmula resulta em resíduos elevados para valores muito pequenos e diferentes das variáveis calculada e experimental, é necessário aplicar uma restrição: o resíduo é calculado apenas quando o valor

experimental é superior a um valor mínimo fixado previamente: $y_{experimental\ i} > \varepsilon (y_{máximo\ experimental\ i})$, evitando-se que valores pouco significativos sejam utilizados no cálculo dos resíduos. Os valores com as fórmulas 5 e 6 são iguais pois, como o ε é muito pequeno, não fez diferença. Assim, na modelagem alvo deste trabalho será utilizada a Fórmula 6.

Tabela 7 – Resultados do estudo com cada fórmula do cálculo do resíduo da Tabela 1

Fórmula	Precisão	Tempo de processo	Valor do Resíduo	Nº de Iterações	$\mu_{x_{max}} (h^{-1})$
1	$1,5 \cdot 10^{-4}$	2 s	9,18	6	$4,2 \cdot 10^{-2}$
2	$4,2 \cdot 10^{-7}$	4 s	3,48	43	$1,4 \cdot 10^{-1}$
3	$3,9 \cdot 10^{-5}$	364 s	125,63	6663	$4,3 \cdot 10^{-1}$
4	$2,2 \cdot 10^{-9}$	11 s	9,68	166	$5,9 \cdot 10^{-2}$
5	$7,4 \cdot 10^{-11}$	9 s	11,59	140	$5,1 \cdot 10^{-2}$
6	$7,4 \cdot 10^{-11}$	9 s	11,59	140	$5,1 \cdot 10^{-2}$
7	$3,2 \cdot 10^{-2}$	19 s	0,71	118	$2,5 \cdot 10^{-1}$
8	$2,4 \cdot 10^{-8}$	7 s	0,000017	87	$1,1 \cdot 10^{-1}$

5.3 LEVANTAMENTO DE ENSAIOS COM AS LINHAGENS rS2

Como neste trabalho não foram feitos ensaios experimentais, fez-se um abrangente estudo sobre todos os ensaios gerados ao longo do Projeto FAPESP/Temático realizados com todas as linhagens das células S2. Foram selecionados, além dos ensaios utilizados para a modelagem (Tabela 3), ensaios válidos (Tabela 8) que podem ser utilizados para verificação do comportamento das outras linhagens após a modelagem matemática ser estabelecida (extrapolação do modelo final).

Esse estudo compreendeu 18 dissertações e teses, 22 artigos científicos, 5 relatórios parciais e um relatório final para a FAPESP. Este estudo faz parte de um artigo de revisão intitulado “*Drosophila melanogaster* S2 cells for expression of heterologous genes: From gene cloning to bioprocess development” publicado no *Journal Biotechnology Advances*, em maio de 2012 (MORAES *et al.*, 2012).

Tabela 8 – Ensaios válidos para este trabalho

Linhagem	Meio de Cultura / Sistema de cultivo	OD	$X_{\max}$ (10^7 cel/mL)	$\mu_{\max}$ (d^{-1})	Referência	
modelagem	S2AcGPV	TC 100 AG/ reator	5%	2,2	0,65	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	50%	0,7	0,50	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	80%	0,1	0,08	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	1,8	0,64	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	0,2	0,23	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	1,0	0,47	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	0,4	0,58	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	0,7	0,21	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	1,4	0,63	Aguiar, 2010
	S2AcGPV	TC 100 AG/ reator	30%	0,6	0,55	Aguiar, 2010
S2AcGPV	TC 100 AG/ reator	30%	1,7	0,65	Aguiar, 2010	
S2AcGPV	SF900 II / reator	50%	5,1	1,07	Pamboukian, 2008	
S2AcGPV	SF900 II / reator	50%	2,7	1,02	Batista, 2007	
S2AcGPV	SF900 II / reator	50%	2,2	1,01	Ventini, 2010	
S2AcGPV	SF900 II / reator	50%	3,0	0,68	Ventini, 2010	
S2MtGPV-H	SF900 II / reator	50%	1,8	1,02	Ventini, 2010	
S2MtGPV-H	SF900 II / reator	50%	1,4	0,94	Ventini, 2010	
S2MtGPV-H	SF900 II / reator	10%	1,8	1,22	Ventini, 2010	
S2MtGPV-H	SF900 II / reator	10%	1,7	1,18	Ventini, 2010	
S2AcGPV	IPL-41 supl./ reator	40%	3,7	0,67	Batista, 2007	
S2AcGPV	IPL-41 supl./ reator	5%	3,0	0,72	Batista, 2007	
S2AcGPV	IPL-41 supl./ reator	20%	1,3	0,74	Batista, 2007	
S2AcGPV	IPL-41 supl./ reator	60%	2,1	0,91	Batista, 2007	

De todo o grupo estudado, foram escolhidos os ensaios feitos em biorreatores (com condições controladas) e que tinham uma condição diferente por ensaio.

Com base na análise do comportamento desses ensaios e nas conclusões oriundas dos trabalhos do grupo do Projeto FAPESP/Temático, foi possível elaborar as principais conclusões sobre o comportamento metabólico da linhagem rS2:

- lactato quase não é produzido;
- amônio é produzida em baixas concentrações, porém após o término da fase exponencial sua produção aumenta;

- glicose e glutamina em baixa concentração limitam o crescimento;
- alanina é produzida e no final dos cultivos pode ser consumida;
- concentrações de oxigênio dissolvido acima de 50% tendem a atuar como inibidor do crescimento celular.

5.4 MODELAGEM DO COMPORTAMENTO DA LINHAGEM rS2

Na Tabela 9 tem-se a sequência dos modelos utilizados nesta modelagem.

Tabela 9 – Sequência de modelos

Modelo	Observações
Modelo 1	Modelo preliminar a partir do metabolismo de células de mamíferos
Modelo 2	Introdução da GOGAT
Modelo 3	Vias metabólicas melhor estruturadas com adição das vias PEPCCK, PC e PDH – 8 submodelos
Modelo 4	Adição de respiração, balanço de ATP e coenzimas Com GOGAT – 21 submodelos Sem GOGAT – 14 submodelos
Modelo 5	Reações do núcleo de glutamina e lactato como reversíveis com 15 submodelos até a solução final. <ul style="list-style-type: none"> • só exponencial • todas as fases • divisão por faixa de concentração de substratos • cálculo da purina inicial • nova divisão por faixa de concentração de substratos

5.4.1 Modelo 1

Foi estudado o metabolismo das células rS2 de maneira a formular um modelo matemático representativo de suas reações.

Para essa proposta preliminar metabólica, partiu-se de uma compilação dos estudos do metabolismo básico de células animais e suas diferenças para células de inseto (ALTAMIRANO; GÓDIA; CAIRÓ, 2008) e estudos específicos para a célula Sf9 (DREWS *et al.*, 2000; BERNAL *et al.*, 2009).

Esta proposta foi preliminar e simplificada, pois se desejava ver como o programa Metatool interpretaria essa rede estudando seu comportamento e aprendendo a como trabalhar com essa ferramenta.

Na Figura 15 tem-se o metabolismo básico proposto neste Modelo 1.

Figura 15 - Metabolismo proposto para a célula rS2 – Modelo 1

O metabolismo foi implementado no Metatool (itens 3.7.7 e 4.2.2) empregando-se as diretrizes do artigo de Provost e Bastin (2004) e Provost *et al.* (2006) e foi aplicado ao conjunto de dados do experimento Ma_07 (AGUIAR, 2010). Conforme a Figura 15 as vias metabólicas propostas para a célula S2 possuem:

- substratos: glicose e glutamina;
- produtos extracelulares: lactato, alanina e amônio;

- metabólitos intracelulares terminais: purina e pirimidina;
- metabólitos internos: glicose 6-P; ribose 5-P; frutose 6-P; gliceraldeído 3-P; dihidroxiacetona-P; piruvato; AcCoA; citrato; α -cetoglutarato; malato; oxaloacetato; aspartato; glutamato.

O *software* Metatool é capaz de gerar, a partir do conjunto de reações prováveis da via metabólica (Modelo), diferentes subconjuntos (Modos Elementares) que, além de atenderem aos requisitos de álgebra linear (vetores não negativos), representam os caminhos mais simples capazes de "conectar" os substratos com os produtos finais.

Todas as reações nesse Modelo 1 foram propostas como irreversíveis.

Para o metabolismo proposto, a aplicação do *software* Metatool resultou em 7 modos elementares conforme listado a seguir. Observa-se que cada reação corresponde a uma via (v_n) conforme mostrado na Figura 15.

e_1 : Glicose $\rightarrow$ Glicose 6-P	(v_1)
Glicose 6-P $\rightarrow$ Dihidroxiacetona P + Gliceraldeído 3-P	(v_2)
Dihidroxiacetona P $\rightarrow$ Gliceraldeído 3-P	(v_4)
2 Gliceraldeído 3-P $\rightarrow$ 2 Piruvato	(v_5)
2 Piruvato $\rightarrow$ 2 Lactato	(v_6)
e_2 : Glicose $\rightarrow$ Glicose 6-P	(v_1)
Glicose 6-P $\rightarrow$ Dihidroxiacetona P + Gliceraldeído 3-P	(v_2)
Dihidroxiacetona P $\rightarrow$ Gliceraldeído 3-P	(v_4)
2 Gliceraldeído 3-P $\rightarrow$ 2 Piruvato	(v_5)
2 Piruvato $\rightarrow$ 2 AcetilcoenzimaA + 2 CO_2	(v_8)
2 Oxaloacetato + 2 AcetilcoenzimaA $\rightarrow$ 2 Citrato	(v_9)
2 Citrato $\rightarrow$ 2 α -cetoglutarato + 2 CO_2	(v_{10})
2 α -cetoglutarato $\rightarrow$ 2 Malato + 2 CO_2	(v_{11})
2 Malato $\rightarrow$ 2 Oxaloacetato	(v_{12})
e_3 : Piruvato + Glutamato $\rightarrow$ Alanina + α -cetoglutarato	(v_7)
α -cetoglutarato $\rightarrow$ Malato + CO_2	(v_{11})
Malato $\rightarrow$ Piruvato + CO_2	(v_{13})
Glutamina $\rightarrow$ Glutamato + NH_4^+	(v_{16})
e_4 : Piruvato $\rightarrow$ Lactato	(v_6)
α -cetoglutarato $\rightarrow$ Malato + CO_2	(v_{11})
Malato $\rightarrow$ Piruvato + CO_2	(v_{13})
Glutamato $\rightarrow$ α -cetoglutarato + NH_4^+	(v_{15})
Glutamina $\rightarrow$ Glutamato + NH_4^+	(v_{16})
e_5 : Piruvato $\rightarrow$ AcetilcoenzimaA + CO_2	(v_8)
AcetilcoenzimaA + Oxaloacetato $\rightarrow$ Citrato	(v_9)
Citrato $\rightarrow$ α -cetoglutarato + CO_2	(v_{10})
2 α -cetoglutarato $\rightarrow$ 2 Malato + CO_2	(v_{11})

Eliminando os metabólitos internos entre as reações de cada modo elementar, temos um conjunto de macro reações representadas por:

Esse conjunto de reações e seus coeficientes estequiométricos podem auxiliar na obtenção dos coeficientes de conversão máximos possíveis, o que norteia o modo de condução dos processos. Além disso, a obtenção da matriz estequiométrica global (resultantes dos modos elementares) possibilita analisar os pesos que cada reação tem no sistema todo, além do estabelecimento da modelagem dinâmica representada por:

$$\frac{d\xi(t)}{dt} = Kv(t) + u(t) \quad (12)$$

Onde: ξ é o vetor das concentrações das variáveis de estado

K é a matriz dos coeficientes estequiométricos

$v(t)$ é o vetor das velocidades dos modos elementares

$u(t)$ é a troca efetiva da variável de estado com o exterior do reator

A matriz estequiométrica K foi gerada a partir dos coeficientes estequiométricos presentes nas 7 equações dos modos elementares.

No caso, as variáveis de estado (ξ) são: glicose, glutamina, lactato, amônio, alanina, CO_2 e nucleotídeos (purina e pirimidina). Como não foram feitas medições do CO_2 , esta variável será desprezada. Como $u(t)=0$ para todas as variáveis, excluindo o CO_2 , que compõe o ar de saída do reator, pois tratam-se de ensaios descontínuos, sem entrada ou saída de líquidos, tem-se que:

$$\frac{d\xi(t)}{dt} = K_V(t)$$

E assumindo-se que:

- como se está trabalhando na fase exponencial: $d\xi(t) = cte$
- como se trata de proposição de reações irreversíveis, tem-se a restrição de que o vetor v deve ter elementos não negativos;
- cinética de Michaelis-Menten, onde K_{GLC} e K_{GLN} são as constantes de Michaelis-Menten.

Tem-se:

$$v_h = a_i \frac{GLC.X}{K_{GLC} + GLC} \quad \text{para os modos elementares 1 e 2;} \quad (13)$$

$$v_k = a_i \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} \quad \text{para os modos elementares 3, 4 e 5;} \quad (14)$$

$$v_k = a_i \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} \quad \text{para os modos elementares 6 e 7.} \quad (15)$$

Onde a_i são as velocidades específicas máximas das reações. Aplicando essas equações:

$$\frac{dX}{dt} = \mu X \quad (16)$$

$$\frac{dGLC(t)}{dt} = -a_1 \frac{GLC.X}{K_{GLC} + GLC} - a_2 \frac{GLC.X}{K_{GLC} + GLC} - a_6 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} - a_7 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} \quad (17)$$

$$\frac{dGLN(t)}{dt} = -a_3 \frac{GLN.X}{K_{GLN} + GLN} - a_4 \frac{GLN.X}{K_{GLN} + GLN} - a_5 \frac{GLN.X}{K_{GLN} + GLN} - 2a_6 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} - 2a_7 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} \quad (18)$$

$$\frac{dLAC(t)}{dt} = 2a_1 \frac{GLC.X}{K_{GLC} + GLC} + a_4 \frac{GLN.X}{K_{GLN} + GLN} \quad (19)$$

$$\frac{dNH_4(t)}{dt} = a_3 \frac{GLN.X}{K_{GLC} + GLN} + 2a_4 \frac{GLN.X}{K_{GLN} + GLN} + 2a_5 \frac{GLN.X}{K_{GLN} + GLN} + a_6 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} + a_7 \frac{GLC.GLN.X}{(K_{GLC} + GLC)(K_{GLN} + GLN)} \quad (20)$$

$$\frac{dALA(t)}{dt} = a_3 \frac{GLN \cdot X}{K_{GLN} + GLN} \quad (21)$$

Os valores das constantes de saturação foram assumidos valores pequenos visto que a solução do sistema desses está considerando a fase exponencial de crescimento, mas ao mesmo tempo grandes o suficiente para não comprometer a rigidez do sistema (*stiffness*). Assim, valores como $K_{GLC}=K_{GLN}=0,1\text{mM}$ são coerentes no emprego dessa modelagem e foram utilizados.

Assim tem-se para o modelo:

$$\xi = \begin{bmatrix} \text{Glicose}(GLC) \\ \text{Glutamina}(GLN) \\ \text{Lactato}(LAC) \\ \text{Amonio}(NH_4) \\ \text{Alanina}(ALA) \\ \text{Nucleotídeos}(NUC) \end{bmatrix} \quad K = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 & -3 & -2 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Como o vetor ξ é o perfil das variáveis de estado ao longo do tempo medidas, pode-se admitir que ξ seja o vetor das máximas velocidades específicas de consumo/produção dos metabólitos, considerando-se que na fase exponencial tem-se um estado quase-estacionário.

Portanto, temos:

$$\underbrace{\begin{pmatrix} \mu_{GLC} \\ \mu_{GLN} \\ \mu_{LAC} \\ \mu_{NH_4} \\ \mu_{ALA} \\ \mu_{NucI} \end{pmatrix}}_{\xi} = \underbrace{\begin{pmatrix} -1 & -1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 & -3 & -2 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}}_K \underbrace{\begin{pmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \\ a7 \end{pmatrix}}_v \quad (22)$$

Assim, é possível calcular os valores dos parâmetros a_1 a a_7 . Como se considera a fase exponencial, os valores do vetor ξ são as máximas velocidades específicas de consumo de substrato e formação de produtos, conforme a Tabela 10.

Tabela 10 – Máximas velocidades específicas de consumo de substratos e formação de produtos dos ensaios para modelagem na fase exponencial, calculadas através do TataDados

Ensaio	μ_{Xmax} (h ⁻¹)	μ_{GLCmax} (10 ⁻¹² mmol/ (cel.h))	μ_{GLNmax} (10 ⁻¹² mmol/ (cel.h))	μ_{LACmax} (10 ⁻¹² mmol/ (cel.h))	μ_{NH4max} (10 ⁻¹² mmol/ (cel.h))	μ_{ALAMax} (10 ⁻¹² mmol/ (cel.h))
Ma_07	0,027	45,11	3,97	2,54	3,66	31,55
Ma_08	0,021	167,62	7,31	0,77	19,14	62,21
Ma_09	0,004	27,91	0,51	0	12	104,26
Ma_10	0,027	33,04	0,51	2,52	287,53	40,01
Ma_11	0,009	42,51	3,99	3,77	10,10	48,31
Ma_12	0,020	36,24	0,02	0,62	6,89	46,43
Ma_14	0,024	109,50	4,10	4,02	30,87	80,07
Ma_15	0,009	5,02	0,82	3,98	4,12	16,90
Ma_16	0,026	19,15	4,12	6,19	14,98	49,28
Ma_17	0,023	128,62	0,60	5,00	38,06	59,08

Segundo Provost e Bastin (2004) e Provost *et al.* (2006) a resolução desse sistema de equações ordinárias deveria ser executada através do método de resolução matemática de análise de cone convexo, porém em todos os resultados encontrados existia algum valor negativo e, como se trata de proposição de reações irreversíveis, tem-se a restrição de que o vetor v deve ter elementos não negativos. Então, utilizando a resolução matemática do método do cone convexo não haveria solução possível para esse sistema, significando que o Modelo 1 proposto não representa um sistema possível.

Foi executado, então, um outro método para a obtenção dos parâmetros, o *fmincon* com algoritmo *interior point* através de programação em MATLAB. Como foi utilizada a restrição de que só poderiam ser encontrados valores positivos, encontrou-se um conjunto de parâmetros para cada ensaio, porém com resíduo alto. Na programação, após o *fmincon* encontrar um conjunto dos valores dos parâmetros "a's" é feito um cálculo através da Eq. 22 para ver qual o valor resultante do vetor ξ . O resíduo calculado é a diferença entre esse valor encontrado e o valor correto do vetor ξ . De modo geral, o resíduo variou de 13 a 25%. Sendo assim, foi encontrada uma solução para o sistema de equações, porém, devido ao alto valor de resíduo,

esta solução não representa bem o sistema (Figura 16, 17 e 18). O conjunto de parâmetros a_i está descrito na Tabela 10.

Tabela 11 – Valores dos parâmetros a_i para cada ensaio no Modelo 1

Ensaio	$a_i (10^{-15} / (\text{cel} \cdot \text{h}))$						
	a_1	a_2	a_3	a_4	a_5	a_6	a_7
Ma_07	16,61	59,10	36,02	0,00	0,00	21,05	21,05
Ma_08	21,83	141,00	44,12	0,00	0,00	29,33	29,33
Ma_09	20,38	76,99	71,21	0,00	0,00	12,60	12,60
Ma_10	16,41	66,09	42,45	0,00	0,00	17,64	17,64
Ma_11	17,99	127,71	45,52	0,00	0,00	19,88	19,88
Ma_12	17,35	77,87	45,58	0,00	0,00	18,19	18,19
Ma_14	22,61	91,27	48,75	0,00	0,00	29,64	29,64
Ma_15	7,74	68,62	27,19	0,00	0,00	4,97	4,97
Ma_16	15,51	122,39	51,80	0,00	0,00	11,54	11,54
Ma_17	13,23	173,71	66,48	0,00	0,00	1,24	1,24

Nota-se que, na solução encontrada, os modos elementares 4 e 5 não estariam ativos (Tabela 10 colunas a_4 e a_5).

Figura 16 - Simulações do Modelo 1. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio.

Figura 17 – Simulações do Modelo 1. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio.

Figura 18 – Simulações do Modelo 1. Os dados experimentais são representados por linha e a linha representa os dados simulados para cada ensaio.

Nas Figuras 16, 17 e 18 apresentaram-se os dados simulados para a fase exponencial utilizando-se as equações (16) a (21) e os parâmetros encontrados da Tabela 11. Nota-se que na maioria dos gráficos os valores simulados desviam muito dos valores experimentais, o que mostra que o Modelo 1 proposto não representa satisfatoriamente o conjunto de dados experimentais.

Assim, entende-se, pela inspeção visual, que a solução encontrada sugere que o Modelo 1 não representa o que na realidade acontece nas vias metabólicas das células rS2. Sendo assim, na próxima formulação de modelo será proposta uma via de importante discussão nas células de inseto, que é a GOGAT (DREWS *et al.*, 2000; PICCOLI *et al.*, 2009; AGUIAR, 2010), na direção de confirmar essa possibilidade, embora neste trabalho não se tenha condições de verificar-se experimentalmente.

5.4.2 Modelo 2

Como já citado anteriormente, ainda não foram encontrados na literatura, estudos detalhados relacionados ao metabolismo das células S2 de *Drosophila melanogaster* em cultura.

Para se verificar a influência específica da via metabólica GOGAT, cuja possibilidade de existência é prevista pelo KEGG, alterou-se o Modelo 1 somente com a adição

dessa via, conforme Figura 19. A via GOGAT é uma via de reação entre glutamina e α -cetoglutarato resultando em glutamato, não formando, portanto, o amônio.

Figura 19 –Metabolismo proposto para a célula rS2 com a adição da via GOGAT (v_{19}) – Modelo 2

O genoma da *Drosophila melanogaster* já foi mapeado e já foram identificadas as principais enzimas que podem atuar nos processos metabólicos em geral (KEGG). Cada enzima identificada pode ou não estar ativa, dependendo se o gene, que codifica essa enzima, está expresso ou inibido na célula rS2.

Na Figura 20 apresenta-se, como exemplo do KEGG, o metabolismo da glutamina com a presença da GOGAT (glutamato sintase). As enzimas destacadas em verde estão presentes na *Drosophila melanogaster* e o programa permite obter a descrição de cada enzima, onde ela pode atuar e sua descrição gênica.

Eliminando-se os metabólitos internos dos vários modos elementares indicados na Figura 21, tem-se um conjunto de 9 macro reações representadas por:

- $e_1:$ GLC = 2 LAC
 $e_2:$ GLN = 2 CO₂ + ALA + NH₄⁺
 $e_3:$ GLC + 2 GLN = 2 CO₂ + NH₄⁺ + PURI
 $e_4:$ GLC = 6 CO₂
 $e_5:$ GLC + GLN = ALA + GLU
 $e_6:$ GLN = GLU + NH₄⁺
 $e_7:$ GLN = 2 CO₂ + LAC + 2 NH₄⁺
 $e_8:$ GLC + 2 GLN = 2 CO₂ + NH₄⁺ + PYRI
 $e_9:$ GLN = 5 CO₂ + 2 NH₄⁺

Na Tabela 12 são apresentados os parâmetros a_i do Modelo 2.

Tabela 12 – Valores dos parâmetros a_i para cada ensaio no Modelo 2

Ensaio	$a_i (10^{-15} / (\text{cel} \cdot \text{h}))$								
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
Ma_07	9,99	0,00	20,52	70,00	28,65	0,00	0,00	20,52	0,00
Ma_08	18,79	0,00	38,61	229,68	43,07	0,00	0,00	38,61	0,00
Ma_09	16,81	69,65	0,00	119,72	30,62	0,00	0,00	0,00	0,00
Ma_10	4,14	0,00	9,44	100,81	34,63	0,00	0,00	9,44	0,00
Ma_11	6,09	0,00	13,62	236,33	40,53	0,00	0,00	13,62	0,00
Ma_12	3,85	0,00	8,79	133,63	39,81	0,00	0,00	8,79	0,00
Ma_14	18,59	0,00	39,98	147,36	51,35	0,00	0,00	39,98	0,00
Ma_15	3,91	16,92	0,00	77,99	7,90	0,00	0,00	0,00	0,00
Ma_16	7,20	33,05	0,00	191,29	24,38	0,00	0,00	0,00	0,00
Ma_17	16,62	74,01	0,00	180,64	5,64	0,00	0,00	0,00	0,00

Nas Figuras 22 e 23 apresentam-se os dados simulados para a fase exponencial com os parâmetros encontrados da Tabela 12.

Figura 22 – Simulações do Modelo 2. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio.

Figura 23 – Simulações do Modelo 2. Os dados experimentais são representados por símbolos e a linha representa os dados simulados para cada ensaio.

Nota-se, mais uma vez, que na maioria dos gráficos os valores simulados se desviam muito dos valores experimentais. A solução apresentou um resíduo variando de 10 a 18% entre os experimentos. Isto mostra que o Modelo 2 proposto também não representa adequadamente o conjunto de dados experimentais.

5.4.3 Modelo 3

A partir desta etapa, os trabalhos se concentraram na análise metabólica para definição das vias metabólicas, suas velocidades e direções. Partiu-se então, para uma proposição de metabolismo com alterações de reversibilidade ou subtração de

vias para análise do comportamento das vias metabólicas através do *software* Metatool. Como através do KEGG se pode verificar todas as vias metabólicas que podem existir, para o Modelo 3 levou-se em conta a atividade das vias fosfoenolpiruvato carboxiquinase (PEPCK), piruvato carboxilase (PC) e piruvato desidrogenase (PDH), vias 24, 21 e 14 respectivamente da Figura 24, como forma de explicar a pouca formação de lactato nos experimentos considerados (NEERMANN; WAGNER, 1996).

Figura 24 – Proposição do Modelo 3 para a célula rS2

Como se observa na Figura 24 a via glicolítica também foi melhor estruturada, assim como o ciclo de Krebs e a via das pentoses. Assim, tomando o Modelo 3 como base, formularam-se 8 sub-proposições diferentes entre si no que tange a reversibilidade e/ou supressão de algumas vias, conforme Tabela 13.

Tabela 13 – Proposições metabólicas estudados no Modelo 3

Propo- sição	Condições	Quantidade de Modos Elementares	Observações
Modelo 3.01	Reações 23 e 24 ausentes	7	Ajuste muito ruim
Modelo 3.02	Reação 27 ausente	15	Ajuste muito ruim
Modelo 3.03	Reação 24 ausente	210	Modos Elementares incoerentes
Modelo 3.04	Reações 26 e 29 - ausentes	20	Ajuste muito ruim
Modelo 3.05	Reação 29 - ausente	34	Condição para excesso de GLC e GLN
Modelo 3.06	Reação 29 - ausente; 30 - reversível	39	Ajuste muito ruim
Modelo 3.07	Reação 29 e 27 - ausente; 26 e 30 - reversível	43	Condição para GLC limitante
Modelo 3.08	Reação 29 - presente; 26 e 30 - reversível	50	Condição para GLN limitante

Após a análise dos resultados obtidos nessas 8 propostas metabólicas, cada uma nas 10 condições de ensaios que são a base deste estudo, chegou-se aos resultados descritos a seguir:

- Condição de excesso de glicose e/ou glutamina - Modelo 3.05

Segundo a literatura, nessas condições, as células tendem a utilizar a via GOGAT, Figura 25. Dos dez ensaios, os que têm a condição de excesso de glicose inicial, são os que responderam a essa proposição metabólica, conforme pode ser visto na Figura 26. Esses ensaios que responderam bem a esta proposição foram: Ma-07, Ma-10, Ma-12, Ma-14, Ma-16 e Ma-17. Para os outros ensaios as respostas foram muito ruins.

Figura 25 – Modelo 3.05 para condições de excesso de GLC e/ou GLN para a célula rS2 na fase exponencial.

Figura 26 - Simulação para o Modelo 3.05 para condições de excesso de GLC e GLN para a célula rS2 na fase exponencial.

- Condição de glicose limitante

Nesta proposição (Figura 27) não se observa o uso da GOGAT, pois não existe mais a condição de piruvato e glutamina em excesso. Porém, ainda existe a condição de formação de alanina para evitar que seja excretado amônio. O ensaio Ma-11 foi

realizado nessa condição e a simulação correspondente está na Figura 28, onde se observa a boa representatividade desta proposição para a fase exponencial.

Figura 27 - Modelo 3.07 para condição limitante de GLC para a célula rS2 na fase exponencial.

Ma_11 - GLC = 1 g/L; GLN = 3,5 g/L

Figura 28 - Simulação para o Modelo 3.07 para condição limitante de GLC para a célula rS2 na fase exponencial.

- Condição de glutamina limitante - Modelo 3.08

Nesta proposição (Figura 29), tem-se a atuação da GOGAT e a previsão de reversibilidade das vias 26 e 30 (GDH e G). Para este Modelo Metabólico tem-se o ensaio Ma-15 que descreve essa situação, conforme apresentado na Figura 30, onde se observa uma representatividade razoável, pois glutamina e alanina não respondem corretamente ao modelo.

Figura 29 - Modelo 3.08 para condição limitante de GLN para a célula rS2 na fase exponencial.

Ma_15 - GLC = 10 g/L; GLN = 0,6 g/L

Figura 30 - Simulação para o Modelo 3.08 para condição limitante de GLC para a célula rS2 na fase exponencial.

Com estas Proposições de Modelos Metabólicos ainda não estavam descritas as condições de balanço de ATP e co-enzimas, que foi feito no próximo modelo.

As grandes diferenças desse conjunto de proposições encontram-se no metabolismo da glutamina. A condição inicial deste aminoácido é extremamente importante na ativação ou desativação de diferentes vias metabólicas.

5.4.4 Modelo 4

Como o modelo básico anterior conduziu a respostas satisfatórias, fez-se um aprimoramento desse modelo incorporando-se a respiração, com balanço de ATP e co-enzimas. Para melhor análise, dividiu-se o mesmo em duas condições básicas: contendo ou não a via GOGAT (com ou sem a via 26). O modelo básico com GOGAT foi chamado de Modelo 4A, como na Figura 31 e o modelo básico sem a GOGAT (via 26) foi chamado de Modelo 4B.

Tabela 14 - Apresentação das vias metabólicas, numeração e enzimas do Modelo 4

Via Nº	Enzima	Reação	Nome do conjunto das vias
01	PGM1	$GLC + ATP = G6P + ADP$	glicólise
02	PGI	$G6P = F6P$	glicólise
03	PFK	$F6P + ATP = F16P + ADP$	glicólise
04	FDPA	$F16P = 2\ G3P$	glicólise
05	G3PDH	$G3P + NAD = 13PG + NADH$	glicólise
06	PGK	$13PG + ADP = 3PG + ATP$	glicólise
07	PGM2	$3PG = 2PG$	glicólise
08	ENO	$2PG = PEP$	glicólise
09	PYK	$PEP + ADP = PYR + ATP$	glicólise
10	LDH	$PYR + NADH = LAC + NAD$	glicólise
11	6PGL	$G6P + NADP = 6PG + NADH$	Via das pentoses
12	PGDH	$6PG + NADP = R5P + NADPH + CO_2$	Via das pentoses
13	APRT	$R5P + ASP + 2\ GLN + CO_2 + 5\ ATP = PURI + 5\ ADP$	Via das pentoses
14	ACT	$R5P + ASP + GLN + 2\ ATP = PIRI + 2\ ADP$	Via das pentoses
15	PDH	$PYR + CoA + NAD = AcCoA + CO_2 + NADH$	Ciclo de Krebs
16	CS	$AcCoA + OXA = CIT + COA$	Ciclo de Krebs
17	AC	$CIT = ICIT$	Ciclo de Krebs
18	ICDH	$ICIT + NAD = AKG + CO_2 + NADH$	Ciclo de Krebs
19	AKGDH	$AKG + CoA + NAD = SCoA + CO_2 + NADH$	Ciclo de Krebs
20	SCoAS	$SCoA + ADP = SUC + CoA + ATP$	Ciclo de Krebs
21	SDH	$SUC + FAD = FUM + FADH_2$	Ciclo de Krebs
22	FuDH	$FUM = MAL$	Ciclo de Krebs
23	MDH	$MAL + NAD = OXA + NADH$	Ciclo de Krebs
24	AlaTa	$PYR + GLU = ALA + AKG$	Núcleo PYR/AKG
25	GDH	$GLU + NAD = AKG + NH_4 + NADH$	Núcleo PYR/AKG
26	GOGAT	$AKG + GLN + NADH = 2\ GLU + NAD$	Núcleo PYR/AKG
27	AspTa	$OXA + GLU = ASP + AKG$	Núcleo PYR/AKG
28	G	$GLN + ADP = GLU + ATP$	Núcleo PYR/AKG
29	Comp. I, III e IV	$NADH + 2,5\ ADP + 0,5\ O_2 = NAD + 2,5\ ATP$	Fosforilação Oxidativa
30	Comp. II, III e IV	$FADH_2 + 1,5\ ADP + 0,5\ O_2 = FADH + 1,5\ ATP$	Fosforilação Oxidativa
31		$NAD + NADPH = NADH + NADP$	Trans- hidrogenase
32	PEPCK	$OXA = PEP + CO_2$	Núcleo PYR/AKG
33	PC	$PYR + ATP + CO_2 = OXA + ADP$	Núcleo PYR/AKG

A velocidade de transporte de elétrons e da fosforilação oxidativa são controlados estritamente pelas necessidades energéticas da célula. O controle de ADP é ilustrado pelo fato que a mitocôndria só oxida o NADH e o FADH₂ quando houver disponibilidade de ADP e P_i como substratos para a fosforilação. Os elétrons não fluem para a cadeia transportadora de elétrons até o oxigênio, a menos que o ADP seja simultaneamente fosforilado a ATP. A ATP-sintase é inibida em altas concentrações de ATP e ativada quando as concentrações de ADP e P_i estão elevadas. Estudos recentes confirmam os valores (2,5 e 1,5) para a razão P/O (medida do número de ATP sintetizados por átomos de oxigênio utilizado) e, portanto, não correspondem aos valores anteriormente usados (3 e 2 respectivamente) (MOTTA, 2011). Esses valores foram os utilizados nas vias V29 e V30 do metabolismo proposto (Tabela 14).

Tomando-se como base esses dois modelos (4A-com GOGAT e 4B-sem GOGAT), foram feitos estudos de reversibilidade e irreversibilidade, e também presença ou ausência de algumas reações. Observou-se até então, que a grande sensibilidade acontece em torno das vias de síntese e assimilação da glutamina. Sendo assim, foram propostos vários metabolismos conforme as Tabela 15 e 16, sempre estudando a fase exponencial dos 10 ensaios (Ma-07 a Ma-17).

Outra observação importante é a quantidade de purinas e pirimidinas formadas, pois elas são essenciais para a síntese das células. Ou seja, além de serem utilizadas na síntese de DNA e RNA, as purinas são componentes primordiais de várias biomoléculas, como o ATP, GTP, NADH, e Coenzima-A. Três das bases dos ácidos nucleicos, a citosina, a timina e o uracila, são derivados pirimídicos. Se em uma análise metabólica não há formação de purinas e pirimidinas, portanto, não há formação de biomassa, já que elas são a base do DNA.

No tratamento das respostas com os Modos Elementares, considera-se que as quantidades de purina e pirimidina formadas devem ser iguais, já que elas são utilizadas na mesma quantidade para formarem os pares de base do DNA.

Assim, o equacionamento dos modos elementares considera a equação diferencial referente à formação de purina (ou de pirimidina, pois são sintetizadas na mesma quantidade). O problema é que não se possui o valor inicial de purinas, como se tem o valor inicial de glicose, por exemplo.

Partindo da premissa de que cada célula de S2 tem em seu genoma 180 milhões de pares de base (PIVETTA, 2007), tem-se:

- a) 1 célula - 1 genoma - 180 milhões de pares de base
- b) Quantidades iguais de purinas e pirimidinas
- c) 1 genoma tem $180 \cdot 10^6$ moléculas de purinas que correspondem a X mols
- d) 1 mol possui $6,022 \cdot 10^{23}$ moléculas de purinas

Portanto: $X = 298,9 \cdot 10^{-18}$ mols de purinas no genoma/célula

Assim, através da quantidade inicial de células pode-se calcular a quantidade inicial de purina. Nota-se que este cálculo de purina, refere-se ao total de purinas existentes no genoma, e, portanto não leva em conta a quantidade total de purina, já que elas são também a base para outras biomoléculas. Mas por enquanto, esse valor de purina genômica inicial é o que foi usado para se verificar a formação de purina ao longo do ensaio.

Nestes cálculos o crescimento celular continua tendo como base a velocidade específica de crescimento celular de cada ensaio.

Através das proposições metabólicas que são apresentadas nas Tabelas 15 e 16, quando houve fluxo mensurável das vias de formação de purina, o gráfico resultante dessa formação teve o mesmo perfil do crescimento celular, porém em algumas proposições não houve a formação de purina. Este fato é importante, pois muitas vezes os gráficos correspondentes aos metabólitos indicavam que o modelo até respondia razoavelmente bem, porém se não havia a formação de purina, não deveria existir a formação de biomassa e, então, essa proposição era descartada. Com isso, observou-se a importância de se simular o fluxo metabólico de formação de purina (e pirimidina).

Como foram feitas muitas proposições, as Tabelas 15 e 16, apresentam um resumo de tudo que foi estudado. Em cada proposição estão identificadas as suas condições, ou seja, a existência de reversibilidade ou irreversibilidade de reações e também se o glutamato foi considerado na entrada do Metatool como um metabólito interno ou externo. Está também identificada a quantidade de modos elementares gerados em cada proposição e a principal observação.

Tabela 15 – Modelos metabólicos - Modelos 4A (com GOGAT)

Propo- sição	Condições	Quantidade de Modos Elementares	Observações
4A-01	GLU externo; Reversíveis: V10; Irreversíveis: V25 e V28; Ausente: V32 e V33	9	Não forma amônio
4A-02	GLU interno; Ausente: V32 e V33 Irreversíveis: V10, V25 e V28	11	Não forma amônio
4A-03	GLU externo; Ausente: V32 e V33 Irreversíveis: V10, V25 e V28	7	Não forma amônio
4A-04	GLU externo; Irreversíveis: V10, V25 e V28	14	Não forma amônio
4A-05	GLU interno; Irreversíveis: V10, V25 e V28	18	Não forma amônio
4A-06	GLU externo; Reversíveis: V10; Irreversíveis: V25 e V28	19	Ajuste razoável
4A-07	GLU externo; Reversíveis: V10, 25 e V28; Ausente: V32	23	Não forma amônio
4A-08	GLU externo; Reversíveis: V25 e V28; Irreversíveis: V10	14	Não forma amônio
4A-09	GLU externo; Irreversíveis: V10, V25 e V28	10	Não forma amônio
4A-10	GLU externo; Reversíveis: V10, V25 e V28	11	Não forma amônio
4A-11	GLU externo; Reversíveis: V10 e V25; Irreversíveis: V28	11	Mesmos modos elementares que 4A-10
4A-12	GLU externo; Reversíveis: V10, V25 e V28	41	O amônio é só consumido
4A-14	GLU externo; Reversíveis: V10; Irreversíveis: V25 e V28; Ausente: V33	12	Não ajusta lactato e alanina
4A-15	GLU externo; Reversíveis: V25; Irreversíveis: V10 e V28	18	Não ajusta lactato e alanina
4A-16	GLU externo; Ausente: V33 e V32; Reversível: V25; Irreversível: V10 e V28	14	O amônio é só consumido
4A-17	GLU externo; Reversíveis: V10 e V25; Irreversíveis: V28; Ausente: V33 e V32	11	O amônio é só consumido
4A-18	GLU externo; Reversíveis: V10; Irreversíveis: V25 e V28; Ausente: V33 e V32	9	O amônio é só consumido
4A-19	GLU externo; Reversíveis: V10; Irreversíveis: V25 e V28; Ausente: V32	15	Não forma amônio
4A-20	GLU externo; Reversíveis: V10, V25, V28 e V24	19	Não forma amônio e alanina
4A-21	GLU externo; Reversíveis: V10; Irreversíveis: V24, V25 e V28 (ao contrário)	19	Não forma amônio e alanina

Tabela 16 – Modelos metabólicos - Modelos 4B (sem GOGAT)

Proposição	Condições	Quantidade de Modos Elementares	Observações
4B-01	GLU externo; Reversíveis: V10, V25 e V28	18	Não forma purina
4B-02	GLU externo; Reversíveis: V10 e V25	18	Não forma purina
4B-03	GLU externo; Reversíveis: V10	10	Não forma purina
4B-04	GLU externo; Irreversíveis: V10, V25 e V28	9	Não forma purina
4B-05	GLU externo; Reversíveis: V25 e V28 Irreversíveis: V25 e V28	13	Não forma purina
4B-06	GLU externo; Reversíveis: V10, V25 e V28 Ausentes: V32 e V33	8	O amônio e lactato são só consumidos
4B-07	GLU externo; Reversíveis: V25 e V28 Ausentes: V32 e V33	6	O amônio é só consumido
4B-08	GLU externo; Irreversíveis: V10, 25 e V28; Ausentes: V32 e V33	5	Não forma alanina
4B-09	GLU externo; Reversível: V25; Ausente: V32 e V33	6	O amônio e alanina são só consumidos
4B-10	GLU externo; Reversíveis: V25, Irreversíveis: V10 e V28; Ausente: V32 e V33	6	Não forma alanina
4B-11	GLU externo; Reversíveis: V10, V24, V25 e V28	24	Não forma alanina
4B-12	GLU externo; Reversíveis: V10 e V24	17	Não forma amônio
4B-13	GLU externo; Reversíveis: V10 e V28	16	Não forma amônio e alanina
4B-14	GLU externo; Reversíveis: V10, Irreversíveis: V24 (ao contrário), V25 e V28	16	Não forma amônio e alanina

Das proposições das Tabelas 15 e 16, a única que se mostrou consistente foi a proposição 4A-06, na qual das 33 vias metabólicas (Figura 31) propostas, a única reversível seria a V10. Porém, esta proposição não atendia a todos os ensaios na fase exponencial, como no caso dos ensaios Ma-09, Ma-11 e Ma-15, que não tiveram uma boa resposta do modelo (dados não mostrados).

Com base no conhecimento adquirido do comportamento das vias frente ao modelos trabalhados até aquele momento, procedeu-se a montagem de um novo modelo metabólico.

5.4.5 Modelo 5

Como pôde ser observado durante o trabalho com os modelos já apresentados nos itens anteriores, a grande sensibilidade à resposta metabólica para uma modelagem está nas vias de assimilação e síntese de glutamina. Outro ponto importante é a reversibilidade da via V10 ($\text{PYR} + \text{NADH} = \text{LAC} + \text{NAD}$). Observou-se que com essa reversibilidade proposta, o modelo mostra que à medida que a glicose vai se esgotando, o lactato vai sendo consumido. Diante destas experiências, foi proposto o Modelo 5 conforme a Figura 32.

Figura 32 - Representação das Vias Metabólicas do Modelo 5

Neste Modelo 5 todas as reações mantêm-se como na Tabela 14, porém as reações V10, V24, V25, V26, V27 e V28 são consideradas reversíveis e o glutamato foi considerado um metabólito externo. Com essas especificações foi utilizado o Metatool que resultou nos modos elementares conforme item 5.4.5.1.

5.4.5.1 Modos Elementares do Modelo 5

Os dados de entrada para o Metatool foram:

- vias metabólicas: as mesmas descritas na Tabela 14;
- vias reversíveis: V10, V24, V25, V26, V27 e V28;
- metabólitos externos: GLC, LAC, ALA, GLN, NH₄, GLU, PURI, PIRI, CO₂, ADP, ATP e O₂.

As macro-reações dos 37 Modos Elementares fornecidos pelo Metatool foram:

- 1: $ALA = LAC + NH_4$
- 2: $LAC + GLN = ALA + GLU$
- 3: $GLN + ADP = NH_4 + GLU + ATP$
- 4: $GLC + 2 ADP = 2 LAC + 2 ATP$
- 5: $NH_4 + GLU = GLN$
- 6: $2 LAC + NH_4 + 6.5 ADP + 1.5 O_2 = GLU + CO_2 + 6.5 ATP$
- 7: $GLC + ALA + 2 GLN + CO_2 + 2 ADP + 2 ATP + O_2 = PURI$
- 8: $GLC + 2 GLN + GLU + 10.5 ADP + 2.5 O_2 = PURI + CO_2 + 6.5 ATP$
- 9: $GLC + ALA + GLN + ADP + O_2 = PIRI + ATP$
- 10: $LAC + 15 ADP + 3 O_2 = 3 CO_2 + 15 ATP$
- 11: $LAC + ALA + 6.5 ADP + 1.5 O_2 = GLU + CO_2 + 6.5 ATP$
- 12: $GLC + GLN + GLU + 9.5 ADP + 2.5 O_2 = PIRI + 2 CO_2 + 9.5 ATP$
- 13: reação sem metabólitos externos (só com metabólitos internos)
- 14: $GLU + 8.5 ADP + 1.5 O_2 = ALA + 2 CO_2 + 8.5 ATP$
- 15: $2 LAC + GLN + 6.5 ADP + 1.5 O_2 = 2 GLU + CO_2 + 6.5 ATP$
- 16: $GLU + 8.5 ADP + 1.5 O_2 = LAC + NH_4 + 2 CO_2 + 8.5 ATP$
- 17: $2 GLU + 8.5 ADP + 1.5 O_2 = LAC + GLN + 2 CO_2 + 8.5 ATP$
- 18: $GLC + LAC + GLN + NH_4 + ADP + O_2 = PIRI + ATP$
- 19: $GLC + LAC + 2 GLN + NH_4 + CO_2 + 2 ADP + 2 ATP + O_2 = PURI$
- 20: $GLC + LAC + 2 GLN + ADP + O_2 = GLU + PIRI + ATP$
- 21: $GLC + LAC + 3 GLN + CO_2 + 2 ADP + 2 ATP + O_2 = GLU + PURI$

- 22: $\text{GLC} + 2 \text{NH}_4 + 2 \text{ADP} = 2 \text{ALA} + 2 \text{ATP}$
 23: $\text{ALA} + 15 \text{ADP} + 3 \text{O}_2 = \text{NH}_4 + 3 \text{CO}_2 + 15 \text{ATP}$
 24: $2 \text{ALA} + 6.5 \text{ADP} + 1.5 \text{O}_2 = \text{NH}_4 + \text{GLU} + \text{CO}_2 + 6.5 \text{ATP}$
 25: $\text{GLC} + 2 \text{GLN} + 2 \text{ADP} = 2 \text{ALA} + 2 \text{GLU} + 2 \text{ATP}$
 26: $\text{ALA} + \text{GLU} + 15 \text{ADP} + 3 \text{O}_2 = \text{GLN} + 3 \text{CO}_2 + 15 \text{ATP}$
 27: $2 \text{ALA} + 6.5 \text{ADP} + 1.5 \text{O}_2 = \text{GLN} + \text{CO}_2 + 6.5 \text{ATP}$
 28: $\text{GLU} + 23.5 \text{ADP} + 4.5 \text{O}_2 = \text{NH}_4 + 5 \text{CO}_2 + 23.5 \text{ATP}$
 29: $2 \text{GLU} + 23.5 \text{ADP} + 4.5 \text{O}_2 = \text{GLN} + 5 \text{CO}_2 + 23.5 \text{ATP}$
 30: $\text{GLC} + 32 \text{ADP} + 6 \text{O}_2 = 6 \text{CO}_2 + 32 \text{ATP}$
 31: $\text{GLC} + \text{NH}_4 + 8.5 \text{ADP} + 1.5 \text{O}_2 = \text{GLU} + \text{CO}_2 + 8.5 \text{ATP}$
 32: $\text{GLC} + \text{GLN} + 8.5 \text{ADP} + 1.5 \text{O}_2 = 2 \text{GLU} + \text{CO}_2 + 8.5 \text{ATP}$
 33: $\text{GLC} + 2 \text{ALA} + 15 \text{ADP} + 3 \text{O}_2 = 2 \text{GLU} + 2 \text{CO}_2 + 15 \text{ATP}$
 34: $3 \text{GLC} + 2 \text{GLN} + 2 \text{NH}_4 + 4 \text{ADP} + 2 \text{O}_2 = 2 \text{PIRI} + 4 \text{ATP}$
 35: $3 \text{GLC} + 4 \text{GLN} + 2 \text{NH}_4 + 2 \text{CO}_2 + 6 \text{ADP} + 2 \text{ATP} + 2 \text{O}_2 = 2 \text{PURI}$
 36: $3 \text{GLC} + 4 \text{GLN} + 4 \text{ADP} + 2 \text{O}_2 = 2 \text{GLU} + 2 \text{PIRI} + 4 \text{ATP}$
 37: $3 \text{GLC} + 6 \text{GLN} + 2 \text{CO}_2 + 6 \text{ADP} + 2 \text{ATP} + 2 \text{O}_2 = 2 \text{GLU} + 2 \text{PURI}$

5.4.5.2 Equacionamento dos Modos Elementares do Modelo 5

Abaixo está descrito o equacionamento referente aos modos elementares, onde "a" é a máxima velocidade específica de cada modo elementar.

$$dX/dt = m_i X; \quad (22)$$

$$\begin{aligned} d\text{GLC}/dt = & -a(4) * (\text{GLC} * X) / ((k_{\text{glc}} + \text{GLC})) \dots \\ & -a(7) * (\text{GLC} * \text{GLN} * \text{ALA} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{ala}} + \text{ALA})) \dots \\ & -a(8) * (\text{GLC} * \text{GLN} * \text{GLU} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{glu}} + \text{GLU})) \dots \\ & -a(9) * (\text{GLC} * \text{GLN} * \text{ALA} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{ala}} + \text{ALA})) \dots \\ & -a(12) * (\text{GLC} * \text{GLN} * \text{GLU} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{glu}} + \text{GLU})) \dots \\ & -a(18) * (\text{GLC} * \text{GLN} * \text{LAC} * \text{NH}_4 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{lac}} + \text{LAC}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -a(19) * (\text{GLC} * \text{GLN} * \text{LAC} * \text{NH}_4 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{lac}} + \text{LAC}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -a(20) * (\text{GLC} * \text{GLN} * \text{LAC} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{lac}} + \text{LAC})) \dots \\ & -a(21) * (\text{GLC} * \text{GLN} * \text{LAC} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{lac}} + \text{LAC})) \dots \\ & -a(22) * (\text{GLC} * \text{NH}_3 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -a(25) * (\text{GLC} * \text{GLN} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN})) \dots \\ & -a(30) * (\text{GLC} * X) / (k_{\text{glc}} + \text{GLC}) \dots \\ & -a(31) * (\text{GLC} * \text{NH}_4 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -a(32) * (\text{GLC} * \text{GLN} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN})) \dots \\ & -a(33) * (\text{GLC} * \text{LAC} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{lac}} + \text{LAC})) \dots \\ & -3 * a(34) * (\text{GLC} * \text{GLN} * \text{NH}_3 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -3 * a(35) * (\text{GLC} * \text{GLN} * \text{NH}_3 * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN}) * (k_{\text{nh4}} + \text{NH}_4)) \dots \\ & -3 * a(36) * (\text{GLC} * \text{GLN} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN})) \dots \\ & -3 * a(37) * (\text{GLC} * \text{GLN} * X) / ((k_{\text{glc}} + \text{GLC}) * (k_{\text{gln}} + \text{GLN})); \end{aligned} \quad (23)$$

$$\begin{aligned} d\text{GLN}/dt = & -a(2) * (\text{GLN} * \text{LAC} * X) / ((k_{\text{gln}} + \text{GLN}) * (k_{\text{lac}} + \text{LAC})) \dots \\ & -a(3) * (\text{GLN} * X) / (k_{\text{gln}} + \text{GLN}) \dots \\ & +a(5) * (\text{NH}_4 * \text{GLU} * X) / ((k_{\text{nh4}} + \text{NH}_4) * (k_{\text{glu}} + \text{GLU})) \dots \end{aligned}$$

$$\begin{aligned}
& -2a(7)*(GLC*GLN*ALA*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{ala}+ALA)) \dots \\
& -2a(8)*(GLC*GLN*GLU*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{glu}+GLU)) \dots \\
& -a(9)*(GLC*GLN*ALA*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{ala}+ALA)) \dots \\
& -a(12)*(GLC*GLN*GLU*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{glu}+GLU)) \dots \\
& -a(15)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& +a(17)*(GLU*X)/(k_{glu}+GLU) \dots \\
& -a(18)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -2a(19)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -2a(20)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& -3a(21)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& -2a(25)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\
& +a(26)*(ALA*GLU*X)/((k_{ala}+ALA)*(k_{glu}+GLU)) \dots \\
& +a(27)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +a(29)*(GLU*X)/(k_{glu}+GLU) \dots \\
& -a(32)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\
& -2a(34)*(GLC*GLN*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{nh4}+NH_4)) \dots \\
& -4a(35)*(GLC*GLN*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{nh4}+NH_4)) \dots \\
& -4a(36)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\
& -6a(37)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)); \tag{24}
\end{aligned}$$

$$\begin{aligned}
dLAC/dt = & +a(1)*(ALA*X)/(k_{ala}+ALA) \dots \\
& -a(2)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& +2a(4)*(GLC*X)/(k_{glc}+GLC) \dots \\
& -2a(6)*(LAC*NH_4*X)/((k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -a(10)*(LAC*X)/(k_{lac}+LAC) \dots \\
& -a(11)*(LAC*ALA*X)/((k_{lac}+LAC)*(k_{ala}+ALA)) \dots \\
& -2a(15)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& +a(16)*(GLU*X)/(k_{glu}+GLU) \dots \\
& +a(17)*(GLU*X)/(k_{glu}+GLU) \dots \\
& -a(18)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -a(19)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -a(20)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& -a(21)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)); \tag{25}
\end{aligned}$$

$$\begin{aligned}
dNH_4/dt = & +a(1)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +a(3)*(GLN*X)/(k_{gln}+GLN) \dots \\
& -a(5)*(NH_4*GLU*X)/((k_{nh4}+NH_4)*(k_{glu}+GLU)) \dots \\
& -a(6)*(LAC*NH_4*X)/((k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& +a(16)*(GLU*X)/(k_{glu}+GLU) \dots \\
& -a(18)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -a(19)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\
& -2a(22)*(GLC*NH_4*X)/((k_{glc}+GLC)*(k_{nh4}+NH_4)) \dots \\
& +a(23)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +a(24)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +a(28)*(GLU*X)/(k_{glu}+GLU) \dots \\
& -a(31)*(GLC*NH_4*X)/((k_{glc}+GLC)*(k_{nh4}+NH_4)) \dots \\
& -2a(34)*(GLC*GLN*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{nh4}+NH_4)) \dots \\
& -2a(35)*(GLC*GLN*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{nh4}+NH_4)); \tag{26}
\end{aligned}$$

$$\begin{aligned}
dALA/dt = & -a(1)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +a(2)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\
& -a(7)*(GLC*GLN*ALA*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{ala}+ALA)) \dots \\
& -a(9)*(GLC*GLN*ALA*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{ala}+ALA)) \dots \\
& -a(11)*(LAC*ALA*X)/((k_{lac}+LAC)*(k_{ala}+ALA)) \dots \\
& +a(14)*(GLU*X)/(k_{glu}+GLU) \dots \\
& +2a(22)*(GLC*NH_4*X)/((k_{glc}+GLC)*(k_{nh4}+NH_4)) \dots \\
& -a(23)*(ALA*X)/(k_{ala}+ALA) \dots \\
& -2a(24)*(ALA*X)/(k_{ala}+ALA) \dots \\
& +2a(25)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\
& -a(26)*(ALA*GLU*X)/((k_{ala}+ALA)*(k_{glu}+GLU)) \dots \\
& -2a(27)*(ALA*X)/(k_{ala}+ALA) \dots
\end{aligned}$$

$$-2*a(33)*(GLC*ALA*X)/((k_{glc}+GLC)*(k_{ala}+ALA)); \quad (27)$$

$$\begin{aligned} dPURI/dt = & +a(7)*(GLC*GLN*ALA*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{ala}+ALA)) \dots \\ & +a(8)*(GLC*GLN*GLU*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{glu}+GLU)) \dots \\ & +a(19)*(GLC*GLN*LAC*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\ & +a(21)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\ & +2*a(35)*(GLC*GLN*NH_4*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{nh4}+NH_4)) \dots \\ & +2*a(37)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)); \end{aligned} \quad (28)$$

$$\begin{aligned} dGLU/dt = & +a(2)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\ & +a(3)*(GLN*X)/(k_{gln}+GLN) \dots \\ & -a(5)*(NH_4*GLU*X)/((k_{nh4}+NH_4)*(k_{glu}+GLU)) \dots \\ & +a(6)*(LAC*NH_4*X)/((k_{lac}+LAC)*(k_{nh4}+NH_4)) \dots \\ & -a(8)*(GLC*GLN*GLU*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{glu}+GLU)) \dots \\ & +a(11)*(LAC*ALA*X)/((k_{lac}+LAC)*(k_{ala}+ALA)) \dots \\ & -a(12)*(GLC*GLN*GLU*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{glu}+GLU)) \dots \\ & -a(14)*(GLU*X)/(k_{glu}+GLU) \dots \\ & +2*a(15)*(GLN*LAC*X)/((k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\ & -a(16)*(GLU*X)/(k_{glu}+GLU) \dots \\ & -2*a(17)*(GLU*X)/(k_{glu}+GLU) \dots \\ & +a(20)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\ & +a(21)*(GLC*GLN*LAC*X)/((k_{glc}+GLC)*(k_{gln}+GLN)*(k_{lac}+LAC)) \dots \\ & +a(24)*(ALA*X)/(k_{ala}+ALA) \dots \\ & +2*a(25)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\ & -a(26)*(ALA*GLU*X)/((k_{ala}+ALA)*(k_{glu}+GLU)) \dots \\ & -a(28)*(GLU*X)/(k_{glu}+GLU) \dots \\ & -2*a(29)*(GLU*X)/(k_{glu}+GLU) \dots \\ & +a(31)*(GLC*NH_4*X)/((k_{glc}+GLC)*(k_{nh4}+NH_4)) \dots \\ & +2*a(32)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\ & +2*a(33)*(GLC*ALA*X)/((k_{glc}+GLC)*(k_{ala}+ALA)) \dots \\ & +2*a(36)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)) \dots \\ & +2*a(37)*(GLC*GLN*X)/((k_{glc}+GLC)*(k_{gln}+GLN)); \end{aligned} \quad (29)$$

Esse equacionamento refere-se à variação de cada metabólito ao longo do tempo conforme o crescimento celular. A princípio, nesta modelagem, o valor da velocidade específica de crescimento celular de cada ensaio (analisado no TrataDados) é utilizado para o incremento do crescimento celular.

5.4.5.3 Simulação e Cálculo do Resíduo

Para a simulação e cálculo do resíduo (diferença entre o valor experimental e calculado) foi feita uma programação em MATLAB utilizando os valores das velocidades específicas dos modos elementares encontrados, as concentrações de cada metabólito (GLC, GLN, LAC, NH_4^+ , ALA e GLU), o equacionamento dos modos elementares com a função *ode45* (Runge-Kutta de ordem 4 e 5). O resíduo foi calculado através da Fórmula 6 da Tabela 1. Vale ressaltar que até este momento trabalha-se somente com a fase exponencial.

Nas Figuras 33 a 42, tem-se os gráficos referentes a cada ensaio na fase exponencial e respectivos resíduos encontrados. Antes de calcular o resíduo, os valores foram normalizados dividindo-se os mesmos pelo máximo valor (considerando valores experimentais e calculados). Desta forma, todos os valores ficaram no intervalo de 0 a 1, tornando possível a comparação entre os resíduos das diversas variáveis de estado. Ao final do cálculo, o resultado foi dividido pela quantidade de pontos utilizados. Vale ainda ressaltar que os modelos foram trabalhados utilizando-se os dados alisados oriundos do TrataDados, porém na simulação foram utilizados os dados experimentais e, portanto, os resíduos calculados são relativos aos dados simulados e alisados.

Resíduo X = 0,001813

Resíduo NH4 = 0,150304

Resíduo GLC = 0,006186

Resíduo ALA = 0,025722

Resíduo GLN = 0,063329

Resíduo GLU = 0,003730

Resíduo LAC = 0,005249

Figura 33 - Resíduos e Simulação do Ensaio Ma-07 em fase exponencial

Resíduo X = 0,000134

Resíduo NH4 = 0,024283

Resíduo GLC = 0,015994

Resíduo ALA = 0,064270

Resíduo GLN = 0,013403

Resíduo GLU = 0,007124

Resíduo LAC = 0,826455

Figura 34 - Resíduos e Simulação do Ensaio Ma-08 em fase exponencial

Resíduo X = 0,000010

Resíduo NH4 = 0,000713

Resíduo GLC = 0,000542

Resíduo ALA = 0,047190

Resíduo GLN = 0,001586

Resíduo GLU = 0,000172

Figura 35 - Resíduos e Simulação do Ensaio Ma-09 em fase exponencial

Resíduo X = 0,012329

Resíduo NH4 = 0,080665

Resíduo GLC = 0,036917

Resíduo ALA = 0,028313

Resíduo GLN = 30,646319

Resíduo GLU = 0,041861

Resíduo LAC = 0.035577

Figura 36 - Resíduos e Simulação do Ensaio Ma-10 em fase exponencial

Resíduo X = 0.000044

Resíduo NH4 = 0.015713

Resíduo GLC = 0.535990

Resíduo ALA = 0.000513

Resíduo GLN = 0.000271

Resíduo GLU = 0.054588

Resíduo LAC = 0.519129

Figura 37 - Resíduos e Simulação do Ensaio Ma-11 em fase exponencial

Resíduo X = 0,011531

Resíduo NH4 = 0,063273

Resíduo GLC = 0,037104

Resíduo ALA = 0,020528

Resíduo GLN = 0,117964

Resíduo GLU = 0,022509

Resíduo LAC = 0,006186

Figura 38 - Resíduos e Simulação do Ensaio Ma-12 em fase exponencial

Resíduo X = 0,000311

Resíduo NH₄ = 0,064514

Resíduo GLC = 0,001030

Resíduo ALA = 0,016179

Resíduo GLN = 0,013832

Resíduo GLU = 0,000249

Figura 39 - Resíduos e Simulação do Ensaio Ma-14 em fase exponencial

Resíduo X = 0,000014

Resíduo NH_4 = 0,063936

Resíduo GLC = 0,003767

Resíduo ALA = 0,001142

Resíduo GLN = 2,736026

Resíduo GLU = 1,539863

Resíduo LAC = 0,798581

Figura 40 - Resíduos e Simulação do Ensaio Ma-15 em fase exponencial

Resíduo X = 0.000824

Resíduo NH4 = 0.048309

Resíduo GLC = 0.006987

Resíduo ALA = 0.017320

Resíduo GLN = 1.225950

Resíduo GLU = 0.666675

Resíduo LAC = 0.463307

Figura 41 - Resíduos e Simulação do Ensaio Ma-16 em fase exponencial

Resíduo X = 0.008441

Resíduo NH4 = 0.205841

Resíduo GLC = 0.000200

Resíduo ALA = 0.010026

Resíduo GLN = 0.026065

Resíduo GLU = 0.012597

Resíduo LAC = 0.135863

Figura 42 - Resíduos e Simulação do Ensaio Ma-17 em fase exponencial

Observa-se que, na maioria das Simulações, o Modelo 5 está respondendo bem e com resíduos aceitáveis. É interessante notar ainda, que os gráficos de purina, em sua maioria, possuem a mesma inclinação da curva de crescimento celular.

5.4.5.4 Novas Proposições

A partir destes resultados, havia a necessidade de se expandir o modelo para todo o experimento (e não somente a fase exponencial), porém essa tarefa foi difícil e necessitou de várias etapas, conforme está descrito na Tabela 17. Observa-se que essa Tabela descreve o desenvolvimento do trabalho até se conseguir chegar ao modelo final, sempre utilizando o metabolismo proposto na Figura 31, correspondentes enzimas da Tabela 14 e modos elementares do item 5.4.5.1.

Tabela 17 - Etapas da evolução do trabalho.

Nº	Proposta	Resultado	Conclusão/ Observação
1	Ensaio foram divididos por grupos: a) sem limitação de GLC ou GLN; b) com limitação de GLC e; c) com limitação de GLN). Utilizou-se, ainda a média das velocidades específicas de consumo/produção de metabólitos entre os ensaios de cada grupo e modelou-se o comportamento de todo o ensaio (não apenas da fase exponencial).	Durante a fase exponencial, os resultados foram razoáveis, porém com ajuste inferior aos conseguidos anteriormente. Após a fase exponencial, as respostas foram muito variadas e não se conseguiu um bom ajuste.	Observou-se que os ensaios possuem faixas relativas à concentração de glicose e glutamina, onde o comportamento das células é o mesmo entre os ensaios.
2	Cada ensaio foi dividido em faixas de acordo com as concentrações de GLC e GLN vigentes: Faixa 1: GLC alta e GLN baixa; Faixa 2: GLC alta e GLN alta; Faixa 3: GLC baixa e GLN alta; Faixa 4: GLC baixa e GLN baixa Utilizaram-se a média das velocidades específicas de consumo/produção, em cada faixa, como na proposição 1.	Alguns ensaios (como o Ma-07 e Ma-10) apresentaram resultados razoáveis, mas não se conseguiu um ajuste satisfatório para a maioria dos ensaios.	Até esta proposta os valores das constantes de Michaelis-Menten (k) que se utilizava era de 0,1 mM.
3	Com os parâmetros ajustados na proposição 2, tentou-se ajustar valores de k (constante de Michaelis-Menten) para cada ensaio.	Os ajustes foram melhores, porém os valores de k entre os ensaios eram muito diferentes (não se ajustou um único conjunto de valores de k)	Percebeu-se que, alterando os valores de k, encontrou-se mais respostas para a fase estacionária.
4	Usando-se apenas a faixa com alta GLC e GLN (faixa 2, acima) de cada ensaio e mantendo-se valores de k = 0,1 mM, tentou-se o ajuste dos 37 valores de a (velocidades dos modos elementares).	Resíduos ainda altos, mas com um ajuste visual melhor.	
5	Os valores de k foram ajustados como parâmetros, ficando essa faixa 2 com 43 parâmetros (37 valores de a e um k para cada metabólito)	Resultados bem melhores.	Até este momento o crescimento celular tinha como base a velocidade específica de crescimento celular de cada ensaio pertencente a essa faixa 2 e o gráfico de purina apresentava-se com o mesmo perfil do crescimento celular.
6	Ajuste das outras faixas usando-se os valores de a e k (encontrados para a faixa 2) para todos os ensaios (para todas as faixas)	Não deu certo. Os valores encontrados não tinham significância.	

Continua

Nº	Proposta	Resultado	Observação
7	Ajuste das outras faixas usando-se os valores de a e k para cada faixa, mas em um único ajuste.	Apresentou um resultado razoável.	Com esta proposição ficou claro que cada faixa deveria ter seu conjunto de valores de a e de k.
8	Usando-se o cálculo de concentração inicial de purina (Eq. 35) para relacioná-la ao crescimento celular, fez-se o ajuste dos valores de a e de k das várias faixas.	Ajuste muito bom.	Concluiu-se que o cálculo da purina estava no caminho certo, mas ainda precisava melhorar.
9	Nova proposição de faixas de concentração de GLC e GLN, conforme Tabela 19. Admitiu-se que o valor da purina total era proporcional ao crescimento celular ($PURI = K_p \cdot X_{total}$) Utilizando só a faixa 2, ajustaram-se os 43 parâmetros (37 valores de a e um k para cada metabólito) mais 10 valores de K_p (um para cada ensaio).	Melhor ajuste até esse momento.	
10	O valor de K_p (da faixa 3 Tabela 19) de cada ensaio foi fixado e encontraram-se os 43 parâmetros (37 valores de a e um k para cada metabólito) para cada faixa.	Bom ajuste para as fases exponencial e estacionária.	
11	Fixaram-se os 43 parâmetros de cada faixa e ajustou-se um novo conjunto de K_p (um para cada ensaio). Agora, o K_p era calculado utilizando os valores da faixa a qual correspondia aos valores iniciais de GLC e GLN do ensaio.	O resíduo diminuiu bastante.	
12	Fixaram-se esses valores de K_p para cada ensaio e reajustaram-se os 43 parâmetros de cada uma das 7 faixas (Tabela 18).	O resíduo diminuiu bastante.	
13	Fixaram-se novamente os valores dos 43 parâmetros e reajustou-se os valores de K_p .	O resíduo permaneceu praticamente inalterado.	Chegou-se ao melhor resultado.
14	Com as faixas encontradas, fez-se o cálculo do valor correto de K_p , utilizando-se a Equação 35.	Chegou-se a valores praticamente iguais, com valores de R^2 em torno de 0,98.	Concluiu-se que a maneira de se encontrar o K_p estava correta, o que faltava era um melhor ajuste das faixas.

No apêndice G, tem-se a listagem genérica de um AjustaPar utilizado para o ajuste de parâmetros deste Modelo 5.

Com todo o trabalho desenvolvido e descrito na Tabela 17 conseguiu-se chegar a uma modelagem que respondesse bem aos 10 ensaios que estavam sendo trabalhados. A seguir tem-se o detalhamento dos cálculos e resultados obtidos.

5.4.5.5 Cálculo da concentração inicial de purina no Modelo 5

Para o cálculo da concentração inicial de purina, partiu-se do valor do conteúdo de purina no genoma, conforme valor obtido no item 5.4.4: **298,9.10⁻¹⁵ mM de purinas no genoma/célula**, e fez-se o seguinte cálculo:

Seja:

X_0 (concentração inicial de células) = X no instante $t_0 = 0$;

X_i (concentração de células no instante i) = X no instante t_i (início da exponencial);

X_1 (concentração de células no instante 1) = X no instante $t_1 = t_i + 1$ hora;

P_g = Purina genômica [mM];

P_T = Purina total na célula [mM];

k_{pg} = Relação de purina genômica com a purina total;

K_p = Relação da purina total com a concentração celular [cel/mM].

O valor de P_g pode ser obtido a partir da equação:

$$P_g = X \cdot 298,9 \cdot 10^{-15} \quad (30)$$

$$\text{e } P_g = P_T \cdot k_{pg} \quad (31)$$

Igualando-se as equações (31) e (32), tem-se: $P_T \cdot k_{pg} = X \cdot 298,9 \cdot 10^{-15}$

$$\text{E portanto: } X = \frac{P_T \cdot k_{pg}}{298,9 \cdot 10^{-15}} \quad (32)$$

Aplicando-se a Equação 32 para um ΔX , tem-se:

$$\Delta X = X_1 - X_0 \quad \Rightarrow \quad \Delta X = \frac{P_{T1} \cdot k_{pg}}{298,9 \cdot 10^{-15}} - \frac{P_{T0} \cdot k_{pg}}{298,9 \cdot 10^{-15}}$$

$$\Delta X = (P_{T1} - P_{T0}) \frac{k_{pg}}{298,9 \cdot 10^{-15}} \quad (33)$$

$$\text{E fazendo: } K_p = \frac{k_{pg}}{298,9 \cdot 10^{-15}} \quad (34)$$

$$\text{Tem-se: } \Delta X = \Delta P_T \cdot K_p \quad (35)$$

Para se encontrar o valor do ΔP_T é necessário primeiramente, saber-se em qual faixa de concentração de substrato o ensaio se inicia. Utilizando os parâmetros da respectiva faixa e os valores das concentrações iniciais dos metabólitos, calcula-se o valor do ΔP_T através das equações diferenciais do modelo segundo o Método de Euler para 1 hora de ensaio (passo $h=1$). Depois disso, calcula-se o valor de X_1 ($X_1 = X_0 \cdot e^{\mu}$) e com isso encontra-se o valor do ΔX para se poder aplicar a Eq. 35 para determinar o valor do K_p .

Para tanto, foi formulado um estudo levando em conta a influência da quantidade inicial de glicose, de glutamina e de oxigênio dissolvido, dos dez ensaios estudados, sobre a média das velocidades específicas de crescimento celular. Desse estudo criou-se um fator μ , que define a velocidade específica de crescimento como função do valor inicial de glicose seguida de correções dadas pelas influências da quantidade inicial de glutamina e de oxigênio dissolvido. Esse cálculo é feito através das equações advindas das curvas que relacionam a velocidade específica de crescimento celular e GLC/GLN/DO, conforme as Figuras 43, 44 e 45.

Figura 43 - Fator para o cálculo da máxima velocidade específica de crescimento celular segunda a quantidade inicial de glicose

Figura 44 - Fator para o cálculo da máxima velocidade específica de crescimento celular segunda a quantidade inicial de oxigênio dissolvido

Figura 45 - Fator para o cálculo da máxima velocidade específica de crescimento celular segunda a quantidade inicial de glutamina

Na Tabela 08 foram apresentados os valores das velocidades específicas máxima de crescimento celular de cada um dos dez ensaios utilizados para esta modelagem, calculados através do programa TrataDados. Na Tabela 18 faz-se a comparação entre os valores encontrados pelo TrataDados e os valores encontrados através do método acima e que será utilizado no programa de simulação.

Tabela 18 - Comparação entre os valores obtidos da velocidade específica máxima de crescimento celular

Ensaio	$\mu_{x_{max}}$ TrataDados (h^{-1})	$\mu_{x_{max}}$ Calculado pelos fatores μ (h^{-1})	Erro (%)
Ma-07	0,0266	0,0233	12,41
Ma-08	0,0206	0,0220	6,80
Ma-09	0,0035	0,0036	2,86
Ma-10	0,0268	0,0263	1,87
Ma-11	0,0094	0,0091	3,19
Ma-12	0,0197	0,0206	4,57
Ma-14	0,0243	0,0231	4,94
Ma-15	0,0086	0,0084	2,33
Ma-16	0,0264	0,0247	6,44
Ma-17	0,0231	0,0214	7,36

5.4.5.6 Definição das faixas de concentração do Modelo 5

Conforme visto na Tabela 17, o modelo 5 precisou ser fracionado em faixas, observando-se o comportamento de todos os ensaios analisados, pois não se conseguiu uma formulação única que representasse a situação real, fruto da flexibilidade metabólica. Essa divisão é coerente uma vez que as vias metabólicas alteram sua velocidade conforme a disponibilidade de substrato. Como se fez o estudo de muitos ensaios, pode ser observado que o comportamento dos ensaios tinham similaridades entre si dependendo da faixa de concentração de glicose e glutamina, inclusive na concentração de formação dos subprodutos. Assim, na Tabela 19 tem-se a divisão das faixas utilizada na modelagem.

Tabela 19 - Divisão por faixas de concentração de substrato

Faixa	Glicose	Glutamina
1	GLC $\geq$ 60 mM	
2	6 mM $\leq$ GLC < 60 mM	GLN $\geq$ 30 mM
3	6 mM $\leq$ GLC < 60 mM	7 mM $\leq$ GLN < 30 mM
4	6 mM $\leq$ GLC < 60 mM	2,5 mM $\leq$ GLN < 7 mM
5	6 mM $\leq$ GLC < 60 mM	GLN < 2,5 mM
6	GLC < 6 Mm	GLN $\geq$ 7 mM
7	GLC < 6 Mm	GLN < 7 mM

Barral (2010) também utilizou-se, em sua modelagem, de faixas de concentração de substrato para melhor abranger as diferenças metabólicas: sem limitação de GLC e GLN; sem limitação de GLC e com limitação de GLN; com limitação de GLC e sem limitação de GLN; com limitações de GLC e GLN, alterando em cada faixa as equações diferenciais de glutamina, amônio, alanina, glutamato e GPV.

5.4.5.7 Cálculo da GPV_e no Modelo 5

Como esta linhagem estudada expressa GPV de forma constitutiva, tentou-se obter uma modelagem para a formação dessa proteína. Assim, para o cálculo da GPV específica foi feito um estudo entre os dez ensaios e chegou-se às relações representadas pelas Equações 36 e 37.

Para a condição de GLC até 32 mM e GLN até 7 mM tem-se (Figura 46):

$$GPV_e = -2,19 \cdot 10^{-8} t^2 + 3,16 \cdot 10^{-6} t + 4,29 \cdot 10^{-5} \quad (36)$$

Figura 46 - Cálculo da GPV_e para condição $GLC \geq 32$ mM e $GLN \geq 7$ mM

E para condições de $GLC < 32$ mM e $GLN < 7$ mM, tem-se (Figura 47):

$$GPV_e = -1,01 \cdot 10^{-7} t + 6,09 \cdot 10^{-5} \quad (37)$$

onde x é o tempo.

Figura 47 - Cálculo da GPV_e para condição $GLC < 32$ mM e $GLN < 7$ mM

5.4.5.8 Resultados obtidos para o Modelo 5

Utilizando-se o Modelo 5, a proposição 14 da Tabela 17 e os itens 5.4.5.6 a 5.4.5.7 obtiveram-se os resultados apresentados a seguir.

- **Velocidades Específicas dos Modos Elementares**

Para o ajuste final tem-se os valores das velocidades dos modos elementares por faixa (Tabela 19) conforme a Tabela 20.

Tabela 20 - Valores das velocidades específicas dos modos elementares (a) para as diferentes faixas (Tabela 18) [10^{-15} / (cel*h)]

	Faixa 1	Faixa 2	Faixa 3	Faixa 4	Faixa 5	Faixa 6	Faixa7
a ₁	1,00E-03	3,44E+01	1,00E-03	1,00E-03	1,72E+01	1,00E-03	2,57E+00
a ₂	4,00E+01	4,00E+01	2,31E+01	1,00E-03	4,00E+01	1,00E-03	4,00E+01
a ₃	4,00E+01	4,00E+01	3,15E+01	4,00E+01	4,00E+01	3,70E+01	3,28E+01
a ₄	8,99E+00	5,69E+00	1,00E-03	2,83E+00	1,00E-03	3,71E+00	4,75E+00
a ₅	2,45E+01	4,00E+01	4,00E+01	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₆	2,91E+00	4,00E+01	1,00E-03	1,00E-03	1,28E+01	1,40E+00	4,00E+01
a ₇	1,00E-03	1,00E-03	1,00E-03	1,02E+00	1,00E-03	1,00E-03	1,00E-03
a ₈	1,00E-03	1,58E+01	2,36E+01	1,54E+01	7,05E-01	1,27E+01	7,50E+00
a ₉	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₁₀	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₁₁	4,00E+01	1,64E+01	1,35E+00	2,86E+00	1,00E-03	1,65E+00	1,00E-03
a ₁₂	1,00E-03	9,30E+00	4,00E+01	1,00E-03	1,00E-03	4,00E+01	4,00E+01
a ₁₃	1,63E+01	1,96E+01	1,48E+01	1,82E+01	1,48E+01	1,55E+01	1,55E+01
a ₁₄	4,00E+01	3,03E+01	4,00E+01	1,00E-03	1,75E+01	4,00E+01	1,00E-03
a ₁₅	1,04E+01	1,00E-03	1,00E-03	3,68E+01	1,00E-03	1,00E-03	1,00E-03
a ₁₆	4,00E+01	4,00E+01	4,00E+01	4,00E+01	4,00E+01	3,00E+00	6,28E-01
a ₁₇	1,00E-03	1,58E+01	7,36E+00	1,00E-03	3,72E+01	1,00E-03	2,41E-03
a ₁₈	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₁₉	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₂₀	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₂₁	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₂₂	4,00E+01	4,00E+01	1,97E+01	1,84E+01	7,02E+00	7,51E+00	1,00E-03
a ₂₃	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,63E+00	1,00E-03
a ₂₄	1,00E-03	2,68E-01	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₂₅	4,00E+01	4,00E+01	4,00E+01	1,00E-03	1,00E-03	1,33E+01	1,00E-03
a ₂₆	1,91E+01	2,12E+01	4,00E+01	1,26E+01	1,00E-03	4,00E+01	4,00E+01
a ₂₇	4,00E+01	4,00E+01	3,75E+01	1,64E+01	1,00E-03	5,63E+00	1,00E-03
a ₂₈	4,00E+01	4,35E+00	4,00E+01	4,00E+01	1,00E-03	1,00E-03	1,04E+01
a ₂₉	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	9,17E+00
a ₃₀	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	2,26E+01	1,00E-03
a ₃₁	4,21E+00	4,00E+01	1,00E-03	3,58E+00	5,00E+00	1,00E-03	1,00E-03
a ₃₂	4,00E+01	1,00E-03	1,00E-03	1,00E-03	4,01E+00	1,00E-03	4,95E+00
a ₃₃	8,96E+00	1,00E-03	6,34E+00	1,00E-03	2,06E+01	1,00E-03	1,00E-03
a ₃₄	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	4,00E+01
a ₃₅	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₃₆	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03
a ₃₇	1,07E+01	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03	1,00E-03

Observa-se que alguns valores são muito pequenos, o que significa que o modo elementar está em baixa atividade, porém não é totalmente zerado pois, na programação, especificou-se como limite inferior valor de $1,00 \cdot 10^{-3}$. Com isso, nota-se que os modos a₉, a₁₀, a₁₈, a₁₉, a₂₀, a₂₁, a₃₅ e a₃₆, possuem baixa atividade.

• Constantes de Michaelis-Menten

Os valores das constantes de Michaelis por faixa de concentração (Tabela 19) estão apresentados na Tabela 21.

Tabela 21 - Valores das constantes de Michaelis-Menten por faixa K_i = [mM]

	Faixa 1	Faixa 2	Faixa 3	Faixa 4	Faixa 5	Faixa 6	Faixa 7
K_{GLC}	2,00E+01	2,00E+01	8,08E+00	1,84E+00	4,70E+00	2,85E+00	2,00E+01
K_{GLN}	2,00E+01	2,00E+01	1,63E+01	1,45E+01	3,95E+00	2,00E+01	1,29E+01
K_{LAC}	1,00E-02	1,00E-02	1,00E-02	1,00E-02	8,55E-02	1,00E-02	1,53E+01
K_{NH_4}	1,54E+00	1,68E+01	1,15E+00	8,76E-01	1,00E-02	1,00E-02	1,40E+01
K_{ALA}	2,00E+01	1,87E+01	2,00E+01	2,00E+01	2,00E+01	2,00E+01	1,00E-02
K_{GLU}	8,86E-01	1,00E-02	1,43E+01	1,10E+01	9,23E+00	1,18E+01	2,00E+01

5.5 SIMULAÇÃO DO COMPORTAMENTO CINÉTICO UTILIZANDO O MODELO 5

Utilizando-se a divisão do perfil cinético em faixas de concentração distintas de GLC e GLN, e considerando as equações diferenciais, a estimativa da velocidade específica máxima de crescimento celular, o cálculo da concentração inicial da purina, os valores das velocidades específicas dos modos elementares e as constantes de Michaelis-Menten, fez-se um programa para simulação do comportamento cinético da linhagem rS2. Esse Simulador foi desenvolvido pela autora deste trabalho, em MATLAB. Toda a descrição do *software* e sequência de utilização encontram-se detalhados no Apêndice D e a listagem completa do programa está no Apêndice H.

Este Simulador gera os gráficos referentes ao crescimento celular e consumo/produção de metabólitos e produto, tanto com os dados simulados como com os dados experimentais e ainda calcula os respectivos resíduos.

5.5.1 Gráficos Finais e Resíduos dos dez ensaios modelados

Nas Figuras de 48 a 57 tem-se os gráficos com os dados experimentais (Símbolos) e os dados simulados (linhas), resultantes das simulações (segundo a modelagem implementada) em função do tempo, e ainda os resíduos referentes à comparação entre esses dados. Esses resíduos são normalizados e divididos pela quantidade de pontos.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 5% (AGUIAR, 2010)

Resíduo X = 0,029156	Resíduo NH4 = 0,295154
Resíduo GLC = 0,214053	Resíduo ALA = 0,067527
Resíduo GLN = 0,147580	Resíduo GLU = 0,075338
Resíduo LAC = 0,094273	Resíduo GPV _e = 0,126738

Figura 48 - Gráfico com os dados experimentais e simulados do ensaio Ma-07 e correspondentes resíduos

Observa-se que nesta condição o modelo respondeu bem, apresentando valores baixos de resíduos entre os dados experimentais e simulados. Pequenas diferenças entre os valores experimentais e simulados ocorreram no final do cultivo, onde já se tem na simulação um maior acúmulo de erros, devido às mudanças de faixas, que inclusive ocasionaram as ondulações observadas nos gráficos. Apesar desta condição trabalhar com 5% de OD isso foi bem interpretado pelo modelo.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 50% (AGUIAR, 2010)

Resíduo X = 0,036740	Resíduo NH ₄ = 0,079795
Resíduo GLC = 0,046376	Resíduo ALA = 0,019777
Resíduo GLN = 0,033886	Resíduo GLU = 0,026611
Resíduo LAC = 1,074433	Resíduo GPV _e = 0,187919

Figura 49 - Gráfico com os dados experimentais e simulados do ensaio Ma-08 e correspondentes resíduos

Nota-se valores de resíduos baixos. Nesta condição trabalha-se com 50% de OD o que também foi bem interpretado pelo modelo. Pequenas diferenças entre os valores experimentais e simulados ocorreram no final do cultivo do perfil de amônio. Apenas o lactato apresentou um resíduo alto e teve uma discrepância maior de perfil a partir da metade do tempo da simulação.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 80% (AGUIAR, 2010)

Resíduo X = 0,001954 Resíduo NH₄ = 0,010522
 Resíduo GLC = 0,001898 Resíduo ALA = 0,042248
 Resíduo GLN = 0,030760 Resíduo GLU = 0,044708
 Resíduo LAC = dados experimentais não detectados
 Resíduo GPV_e = 0,768030

Figura 50 - Gráfico com os dados experimentais e simulados do ensaio Ma-09 e correspondentes resíduos

Apesar desta condição trabalhar com 80% de OD, isto foi muito bem interpretado pelo modelo. Infelizmente não se consegue comparar os valores de lactato, pois os dados experimentais não puderam ser detectados em sua análise.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,008452	Resíduo NH4 = 0,143972
Resíduo GLC = 0,422713	Resíduo ALA = 0,120806
Resíduo GLN = 0,007720	Resíduo GLU = 0,036975
Resíduo LAC = 0,018144	Resíduo GPV _e = 0,076625

Figura 51 - Gráfico com os dados experimentais e simulados do ensaio Ma-10 e correspondentes resíduos

O modelo respondeu muito bem obtendo-se resíduos baixos de maneira geral. Pequenas diferenças entre os valores experimentais e simulados ocorreram no final do cultivo do perfil de alanina. Esta condição é a dita "padrão" (30% OD, 10 g/L de GLC e 3,5 g/L de GLN) o que reforça a expectativa do bom funcionamento do modelo.

- S2AcGPV meio TC100AG, GLC = 1 g/L, GLN = 3,5 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,004376	Resíduo NH ₄ = 0,065571
Resíduo GLC = 0,117621	Resíduo LAC = 0,011775
Resíduo GLN = 0,007709	Resíduo ALA = 0,067749
Resíduo GLU = 0,004231	Resíduo GPV _e = 0,034568

Figura 52 - Gráfico com os dados experimentais e simulados do ensaio Ma-11 e correspondentes resíduos

Pequenas diferenças entre os valores experimentais e simulados ocorreram no final do cultivo do perfil de amônio, glutamato e glutamina. É importante notar que o mesmo perfil de consumo de lactato a partir da metade do ensaio também foi verificado na simulação.

- S2AcGPV meio TC100AG, GLC = 5 g/L, GLN = 3,5 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,007078	Resíduo NH ₄ = 0,059442
Resíduo GLC = 0,487497	Resíduo ALA = 0,065896
Resíduo GLN = 0,006920	Resíduo GLU = 0,096885
Resíduo LAC = 0,182299	Resíduo GPV _e = 0,068705

Figura 53 - Gráfico com os dados experimentais e simulados do ensaio Ma-12 e correspondentes resíduos

Mais uma vez se pode notar o consumo de lactato e também de alanina sendo mostrado também pela simulação.

- S2AcGPV meio TC100AG, GLC = 15 g/L, GLN = 3,5 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,207885 Resíduo NH₄ = 0,177436
 Resíduo GLC = 0,008382 Resíduo ALA = 0,151443
 Resíduo GLN = 0,073421 Resíduo GLU = 0,085729
 Resíduo LAC = 0,240475 Resíduo GPV_e = 259,344370

Figura 54 - Gráfico com os dados experimentais e simulados do ensaio Ma-14 e correspondentes resíduos

Esta condição é a que apresentou resíduos maiores e apenas os perfis de glicose e glutamina estão respondendo muito bem. Os outros perfis mostram um bom desempenho do modelo até o fim da fase exponencial.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 0,6 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,041831	Resíduo NH4 = 0,198560
Resíduo GLC = 0,036277	Resíduo ALA = 0,034023
Resíduo GLN = 0,109640	Resíduo GLU = 0,010995
Resíduo LAC = 0,143242	Resíduo GPV _e = 0,071937

Figura 55 - Gráfico com os dados experimentais e simulados do ensaio Ma-15 e correspondentes resíduos

Esta condição foi bem interpretada pelo modelo. É importante notar o perfil de amônio que sugere a formação e depois seu consumo, e no final do cultivo novamente a sua formação, o que também foi apontado pela simulação.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 1,75 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,004348	Resíduo NH4 = 0,872861
Resíduo GLC = 0,581064	Resíduo ALA = 0,102067
Resíduo GLN = 0,646505	Resíduo GLU = 0,058172
Resíduo LAC = 0,293996	Resíduo GPV _e = 0,352776

Figura 56 - Gráfico com os dados experimentais e simulados do ensaio Ma-16 e correspondentes resíduos

Nesta condição o modelo respondeu bem, obtendo-se resíduos baixos, apenas com pequenas diferenças entre os valores experimentais e simulados que ocorreram no final do cultivo nos perfis de glutamina, lactato, amônio e alanina.

- S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 7 g/L e DO = 30% (AGUIAR, 2010)

Resíduo X = 0,085122 Resíduo NH₄ = 0,114115
 Resíduo GLC = 0,120863 Resíduo ALA = 0,034165
 Resíduo GLN = 0,139252 Resíduo GLU = 0,353443
 Resíduo LAC = 0,477964 Resíduo GPV_e = 0,434574

Figura 57 - Gráfico com os dados experimentais e simulados do ensaio Ma-17 e correspondentes resíduos

Esta condição que trabalha com excesso de glutamina também foi bem interpretada pelo modelo com desvio do perfil apenas de amônio a partir da metade da simulação.

Conforme já apresentado no item 3.7.8.4, os resíduos foram calculados segundo a Fórmula 6 da Tabela 1, com $\mathcal{E} = 10^{-6}$. Dessa maneira, quando se tem valores muito pequenos, como no caso do final do consumo de glicose no gráfico do ensaio Ma-07 (Figura 48), os valores dos resíduos são grandes.

Esse problema é relatado na literatura: “resíduos elevados para valores muito pequenos e diferentes das variáveis calculada e experimental” (BONOMI;

SCHMIDELL, 2001). Por esse motivo, apesar de visualmente alguns gráficos estarem bons, seus resíduos são mais altos. Uma demonstração desse fato é a Figura 56, na qual o gráfico simulado de consumo de glicose representa bem os dados experimentais, enquanto o gráfico simulado do crescimento celular desvia-se um pouco dos dados experimentais, porém, o resíduo de X é menor que o resíduo de GLC.

5.6 COMPARAÇÃO ENTRE MODELOS

Anteriormente a este trabalho, o mesmo conjunto de dados foi utilizado para uma modelagem com equações cinéticas na forma tradicional. Essa modelagem foi implementada por Barral (2010) e está brevemente descrita no item 3.7.1.

Esse modelo não conseguiu prever o consumo de amônio para a síntese de glutamina como o observado no ensaio Ma-15. E, apesar das tentativas, as estruturas do modelo fenomenológico proposto não foram capazes de contemplar respostas para os ensaios Ma-14 e Ma-17.

Nas Figuras 58 a 64 faz-se um comparativo entre os gráficos com os valores simulados e experimentais da modelagem implementada por Barral (2010) e a deste trabalho.

- **Ma-07**

Modelo Barral:

Modelo Pamboukian:

Figura 58 - Simulações do ensaio Ma-07 para as duas modelagens

- **Ma-08**

Modelo Barral:

Modelo Pamboukian:

Figura 59 - Simulações do ensaio Ma-08 para as duas modelagens

- **Ma-09**

Modelo Barral:

Modelo Pamboukian:

Figura 60 - Simulações do ensaio Ma-09 para as duas modelagens

- **Ma-10**

Modelo Barral:

Modelo Pamboukian:

Figura 61 - Simulações do ensaio Ma-10 para as duas modelagens

- **Ma-11**

Modelo Barral:

Modelo Pamboukian:

Figura 62 - Simulações do ensaio Ma-11 para as duas modelagens

- **Ma-12**

Modelo Barral:

Modelo Pamboukian:

Figura 63 - Simulações do ensaio Ma-12 para as duas modelagens

- **Ma-16**

Modelo Barral:

Modelo Pamboukian:

Figura 64 - Simulações do ensaio Ma-16 para as duas modelagens

As modelagens apresentam semelhanças, porém a modelagem implementada neste trabalho consegue acompanhar melhor a maioria dos perfis, principalmente da concentração de células viáveis. É importante notar também que alguns perfis simulados da modelagem implementada neste trabalho, consegue mostrar a produção e posterior consumo de alguns metabólitos conforme o que acontece com os dados experimentais.

De maneira geral, as simulações da modelagem utilizando os fluxos metabólicos aproximaram-se muito mais dos valores experimentais, pois além de simular os dez ensaios do conjunto, responderam ao perfil de lactato, amônio e alanina satisfatoriamente e ainda foram capazes de simular outros 13 ensaios conforme será apresentado no item 5.9. Isso reafirma que a flexibilidade metabólica existente nessa linhagem rS2 consegue ser melhor interpretada quando se analisa mais detalhadamente a maneira como a célula se organiza em relação aos nutrientes que tem a disposição através de alterações nos fluxos metabólitos.

5.7 FLUXOS DAS VIAS METABÓLICAS

Utilizando-se as velocidades específicas dos modos elementares e analisando quais vias metabólicas pertencem a cada modo elementar, pôde-se encontrar os fluxos das vias metabólicas correspondentes a cada faixa de concentração de substratos utilizada no modelo ajustado. Assim, as Figuras 65 a 71 representam os fluxos das vias metabólicas correspondente a cada faixa (1 a 7) de concentração de substrato.

Esses fluxos das vias metabólicas são encontradas fazendo-se a soma da velocidade cada vez que uma via metabólica participa de um modo elementar, assim, tem-se a soma total do fluxo dessa via. Com isso, pode-se observar qual via é mais ativada, se uma reação está ocorrendo preferencialmente como reversível ou em seu sentido original.

Nas Figuras 65 a 71 o fluxo de cada via é representada pela cor vermelha e é expressa em 10^{-12} mmol/(cel.h). As setas preenchidas representam o sentido normal da reação e a seta vazada o sentido de reversibilidade. Se uma reação for negativa, então ela representa que a reação está ocorrendo preferencialmente no sentido da

reversibilidade. Quando se fala em preferencialmente no sentido da reversibilidade, se quer dizer que a reação está ocorrendo nos dois sentidos, porém, na soma geral o que prevalece é o sentido de reversibilidade.

Através das Figuras 65 a 71 pode ser observada a diferença entre os fluxos das vias metabólicas a cada faixa de concentração de substrato. O fluxo das vias de degradação de glicose, por exemplo, vai diminuindo à medida que a disponibilidade de glutamina vai diminuindo (faixas 2, 3 e 4 - Figuras 66, 67 e 68) e aumenta quando a oferta de glutamina é maior (faixas 6 e 7 - Figuras 70 e 71)). Na faixa 5, Figura 69, na qual a disponibilidade de glutamina é baixa a célula mostra preferência a se manter do que crescer.

Deve-se observar que a reação para formação de lactato, na faixa 7, apresenta-se como negativa, mostrando que a reação ocorre preferencialmente no sentido da reversibilidade e realmente nessas condições do cultivo o perfil do lactato apresenta-se praticamente estável ou sendo consumido. O mesmo acontece com a reação de formação da alanina.

5.7.1 Faixa 1 - GLC ≥ 60 mM

Legenda:

→ Sentido normal da reação

⇌ Sentido de reversibilidade

[a] = 10E-12 mmol/(cel.h)

Figura 65 - Velocidades das Vias Metabólicas da Faixa 1 (GLC ≥ 60 mM)

5.7.2 Faixa 2 - $6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $\text{GLN} \geq 30 \text{ mM}$

Figura 66 - Velocidades das Vias Metabólicas da Faixa 2 ($6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $\text{GLN} \geq 30 \text{ mM}$)

5.7.3 Faixa 3 - $6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $7 \text{ mM} \leq \text{GLN} < 30 \text{ mM}$

Legenda:

→ Sentido normal da reação

⇌ Sentido de reversibilidade

[a] = $10\text{E}-12 \text{ mmol}/(\text{cel.h})$

Figura 67 - Velocidades das Vias Metabólicas da Faixa 3 ($6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $7 \text{ mM} \leq \text{GLN} < 30 \text{ mM}$)

5.7.5 Faixa 5 - $6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $\text{GLN} < 2,5 \text{ mM}$

Legenda:

→ Sentido normal da reação

⇌ Sentido de reversibilidade

[a] = $10\text{E-}12 \text{ mmol}/(\text{cel.h})$ Figura 69 - Velocidades das Vias Metabólicas da Faixa 5 ($6 \text{ mM} \leq \text{GLC} < 60 \text{ mM}$ e $\text{GLN} < 2,5 \text{ mM}$)

5.7.6 Faixa 6 - GLC < 6 mM e 7 mM ≤ GLN

Legenda:

→ Sentido normal da reação

⇌ Sentido de reversibilidade

[a] = 10E-12 mmol/(cel.h)

Figura 70 - Velocidades das Vias Metabólicas da Faixa 6 (GLC < 6 mM e GLN ≤ 7 mM)

5.7.7 Faixa 7 - GLC < 6 mM e 7 mM < GLN

Legenda:

→ Sentido normal da reação

⇌ Sentido de reversibilidade

[a] = 10-12 mmol/(cel.h)

Figura 71 - Velocidades das Vias Metabólicas da Faixa 7 (GLC < 6 mM e GLN < 7 mM)

5.8 VALIDAÇÃO DO MODELO 5

O ajuste dos parâmetros do modelo proposto a um conjunto de ensaios experimentais é normalmente avaliado e considerado satisfatório ou não, por simples inspeção visual do conjunto de ensaios, além da análise dos resíduos (BONOMI; SCHMIDELL, 2001). Apesar dessa constatação, é importante submeter os ajustes obtidos a uma análise estatística específica.

Em um primeiro momento, calcula-se o coeficiente angular (CA) e o coeficiente de correlação (R^2), através de uma regressão pela origem, para a curva experimental e ajustada, construída com todos os pontos usados no ajuste. Os valores obtidos são comparados com os valores ótimos (CA=1 e $R^2=1$) (BARRAL, 2010). Neste trabalho ambos os coeficiente foram calculados em relação a curva de crescimento celular e também metabólitos (glicose, glutamina, glutamato, alanina, amônio e lactato). Para o cálculo, cada variável foi normalizada por ensaio. Nas Figuras 72 a 78, têm-se os gráficos e respectivos coeficientes.

Coeficiente Angular X_v = 0,987 Coeficiente Correlação X_v = 0,934

Figura 72 - Gráfico com os todos os dados experimentais e simulados de crescimento celular e correspondentes coeficientes

Coeficiente Angular GLC = 1,049 Coeficiente Correlação GLC = 0,958

Figura 73 - Gráfico com os todos os dados experimentais e simulados de glicose e correspondentes coeficientes

Coeficiente Angular GLN = 1,082 Coeficiente Correlação GLN = 0,935

Figura 74 - Gráfico com os todos os dados experimentais e simulados de glutamina e correspondentes coeficientes

Coeficiente Angular LAC = 0,940 Coeficiente Correlação LAC = 0,717

Figura 75 - Gráfico com os todos os dados experimentais e simulados de lactato e correspondentes coeficientes

Coeficiente Angular amônio = 0,768 Coeficiente Correlação amônio = 0,815

Figura 76 - Gráfico com os todos os dados experimentais e simulados de amônio e correspondentes coeficientes

Coefficiente Angular ALA = 0,792 Coeficiente Correlação ALA = 0,856

Figura 77 - Gráfico com os todos os dados experimentais e simulados de alanina e correspondentes coeficientes

Coefficiente Angular GLU = 1,007 Coeficiente Correlação GLU = 0,369

Figura 78 - Gráfico com os todos os dados experimentais e simulados de glutamato e correspondentes coeficientes

Coefficiente Angular $X_v + GLC + GLN = 1,030$ Coeficiente Correlação $X_v + GLC + GLN = 0,939$

Figura 79 - Gráfico com os todos os dados experimentais e simulados de células viáveis, glicose e glutamina, e correspondentes coeficientes

Coefficiente Angular global = 0,958 Coeficiente Correlação global = 0,876

Figura 80 - Gráfico com os todos os dados experimentais e simulados de todas as variáveis e correspondentes coeficientes

Para todas as variáveis o Coeficiente Angular é de 0,958 e o Coeficiente de Correlação é de 0,876, o que faz o modelo ser considerado válido. Porém, se a média desses coeficientes forem calculadas apenas para o crescimento celular e os dois principais metabólitos (glicose e glutamina), a média do Coeficiente Angular é de 1,030 e do Coeficiente de Correlação é de 0,939, o que se aproxima muito mais do ideal que é a unidade.

Como esta modelagem não levou em conta a presença (consumo e formação) de outros aminoácidos, além de glutamina e glutamato, isso pode influenciar os perfis dos subprodutos (lactato, alanina, amônio e glutamato), fazendo com que os valores dos coeficientes sejam menores.

Para se efetuar uma análise estatística coerente, deve-se efetuar o Teste F conforme visto no item 3.7.9, que foi aplicado levando-se em conta o crescimento celular e o perfil de glicose, glutamina, glutamato, lactato, alanina, amônio e GPV_e.

Os valores calculados foram:

$$s_c^2 \text{ (estimativa da variância do erro do modelo)} = 0,1987$$

$$(nv)_c \text{ é o número de pontos ajustados (todos os 10 ensaios e variáveis)} = 1264;$$

$$p \text{ é o número de parâmetros do Modelo} = 43.$$

Como na Eq. 13 (item 3.7.9) necessita-se que todas variáveis estejam na mesma unidade e como todas as variáveis estão na unidade [g/L], foi necessário transformar o número de células. Assim, foi utilizado para o cálculo de peso das células o valor de massa seca (VIEIRA, 2010) = $1,06 \cdot 10^{-10}$ g/cel

$$v \text{ é número de variáveis de estado} = 8$$

$$\text{Para que } s_c^2 < s_e^2, \text{ deve-se ter um } \mathcal{E} > 0,1586$$

Portanto, o resultado obtido indica que, se o erro experimental dos dados disponíveis for maior do que 16 %, o modelo ajustado é adequado, segundo o Teste F modificado. Segundo Aguiar (2010), os erros das medidas de concentração celular, assim como das medidas analíticas de glicose e glutamina estão de maneira geral na ordem de 15-20%. Portanto, 16% é um erro experimental razoável para medidas em processos biotecnológicos (principalmente se for levada em conta a pouca reprodutibilidade dos processos com as células S2) e conclui-se que o modelo ajustado é adequado. Barral (2010) ajustou um modelo matemático para

sete dos dez ensaios do conjunto com as células S2 com um nível de significância de 15%.

Para análise dos valores dos resíduos obtidos na simulação, fez-se a soma dos resíduos de todos os pontos de todas as variáveis de todos os ensaios modelados ao longo do trabalho, dividida pelo número de pontos experimentais subtraído pelo número de parâmetros (Figura 81). Para coerente comparação com o resíduo final do trabalho de BARRAL (2010), levou-se em consideração apenas os ensaios em comum da simulação final de ambos os trabalhos.

Figura 81 - Gráfico com a evolução dos resíduos ao longo do trabalho

Cada modelo reflete um Modelo Metabólico principal (com as suas variações de reversibilidade), porém até o Modelo 4 cada ensaio (de cada um dos dez utilizados nesta modelagem) tinha um conjunto de valores de parâmetros e utiliza o seu próprio valor de velocidade específica de crescimento. A partir do Modelo 5 foi utilizado um único conjunto de parâmetros para cada faixa de concentração de glicose e glutamina e a velocidade específica de crescimento foi estimada pelo cálculo dos fatores conforme item 5.4.5.5.

5.9 MODELO APLICADO A OUTROS ENSAIOS DO GRUPO

Utilizando-se o Modelo encontrado para o comportamento cinético de células rS2, fez-se uma simulação com outros ensaios conforme citado na Tabela 8 no item 5.3. Observa-se que esses experimentos não foram usados na modelagem para o cálculo dos parâmetros, apenas suas condições iniciais foram usadas em simulações com o modelo desenvolvido e o resultado das simulações foram comparadas com seus valores experimentais.

Considerando-se que o modelo foi desenvolvido com o meio TC-100AG (GALESI *et al.*, 2008) e construção da linhagem celular S2AcGPV, e que os demais ensaios da Tabela 8 usaram outros meios e construção, acredita-se que essas alterações influenciam o valor da velocidade específica máxima de crescimento celular.

Por isso, antes de se fazer a simulação, fez-se um estudo (com os vários ensaios com cada meio e construção diferentes dos usados no modelo) dessas velocidades em cada situação. Esse estudo gerou um valor médio dessas velocidades e fez-se uma relação com a velocidade específica calculada pelo modelo para o meio TC-100AG e construção S2AcGPV. Essa relação gerou um fator multiplicador para a máxima velocidade específica de crescimento celular conforme Tabela 22.

Tabela 22 - Fator multiplicador da máxima velocidade específica de crescimento, baseado na construção S2AcGPV em meio TC-100AG

Condição	Fator multiplicador
Meio SF900 II	1,41
Meio IPL-41	1,44
Construção S2MtGPVH	1,36

Nas Figuras 82 a 94, tem-se a simulação dos vários ensaios e respectivos resíduos obtidos. Também é mostrado o cálculo dos coeficientes angular e de correlação referentes à comparação entre os dados experimentais e simulados.

Nota-se que a produção de GPV foi modelada para o meio TC100AG e apresenta diferenças para os outros meios, devido à presença de diferentes aminoácidos nos

meios. E ainda que, as medições de GPV nos trabalhos de Batista (2007) e Pamboukian (2008) apresentavam problemas e são diferentes do protocolo atual de medição.

Na Tabela 23 apresenta-se os valores das velocidades específicas máximas de crescimento celular dos ensaios simulados. São mostrados os valores obtidos pelo TrataDados e os valores calculados pelo simulador do comportamento cinético.

Tabela 23 - Comparação entre os valores da velocidade específica máxima de crescimento celular

Simulação	$\mu_{x_{max}}$ TrataDados (h^{-1})	$\mu_{x_{max}}$ Calculado (h^{-1})	Erro %
1	0,027	0,027	0,8
2	0,045	0,035	21,27
3	0,043	0,030	28,71
4	0,042	0,030	27,79
5	0,028	0,022	22,89
6	0,043	0,038	10,80
7	0,039	0,037	4,34
8	0,051	0,048	6,86
9	0,049	0,048	2,85
10	0,028	0,027	3,21
11	0,030	0,029	3,33
12	0,031	0,030	1,73
13	0,038	0,017	55,69

Pode-se notar que apesar de alguns valores de erros serem grandes na velocidade específica máxima de crescimento celular, os perfis de glicose e glutamina, principalmente, estão respondendo razoavelmente bem nesses casos.

Simulação 1 - S2AcGPV meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 30%
(repetição do Ma-10) (AGUIAR, 2010)

Resíduo X = 0.013971
 Resíduo GLC = 0.016776
 Resíduo GLN = 0.253729
 Resíduo LAC = 0.061104
 Xv: Coeficiente Angular = 1.017
 GLC: Coeficiente Angular = 0.797
 GLN: Coeficiente Angular = 0.821

Resíduo NH4 = 0.434641
 Resíduo ALA = 0.002232
 Resíduo GLU = 0.034553
 Resíduo GPVe = 0.228457
 Coeficiente de Correlação = 0.978
 Coeficiente de Correlação = 0.991
 Coeficiente de Correlação = 0.987

Figura 82 - Simulação S2AcGPV com meio TC100AG, GLC = 10 g/L, GLN = 3,5 g/L e DO = 30%

Simulação 2 - S2AcGPV meio SF900 II e DO = 40% (PAMBOUKIAN, 2008)

Resíduo X = 0.085485

Resíduo GLC = 0.092333

Resíduo GLN = 0.710974

Xv: Coeficiente Angular = 0.646

GLC: Coeficiente Angular = 0.895

GLN: Coeficiente Angular = 1.417

Resíduo LAC = 2.426424

Resíduo GPV_e = 18.898593

Coeficiente de Correlação = 0.873

Coeficiente de Correlação = 0.995

Coeficiente de Correlação = 0.967

Figura 83 - Simulação S2AcGPV com meio SF900 II e DO = 40%

Simulação 3 - S2AcGPV meio SF900 II e DO = 50% (BATISTA, 2007)

Resíduo X = 0.036423

Resíduo GLC = 0.060993

Resíduo LAC = 1.451022

Xv: Coeficiente Angular = 0.798

GLC: Coeficiente Angular = 1.061

Resíduo NH4 = 0.661047

Resíduo GPVe = 3.817876

Coeficiente de Correlação = 0.991

Coeficiente de Correlação = 0.992

Figura 84 - Simulação S2AcGPV com meio SF900 II e DO = 50%

Simulação 4 - S2AcGPV meio SF900 II e DO = 50% - Reator: Inceltech Discovery
 100 - Temperatura: 28 a 22°C (VENTINI, 2010)

Resíduo X = 0.042133

Resíduo GLC = 0.049859

Resíduo GLN = 0.003700

Xv: Coeficiente Angular = 0.788

GLC: Coeficiente Angular = 0.537

GLN: Coeficiente Angular = 1.079

Resíduo LAC = 0.634681

Resíduo GLU = 0.026706

Resíduo GPV_e = 0.787606

Coeficiente de Correlação = 0.994

Coeficiente de Correlação = 0.954

Coeficiente de Correlação = 0.983

Figura 85 - Simulação S2AcGPV com meio SF900 II, DO = 50% e temperatura = 28 - 22°C

Simulação 5 - S2AcGPV meio SF900 II e DO = 50% - Reator: Bioflo 110 -
Temperatura: 28°C (VENTINI, 2010)

Resíduo X = 0.496197
 Resíduo GLC = 0.015529
 Resíduo GLN = 0.027275
 Xv: Coeficiente Angular = 0.229
 GLC: Coeficiente Angular = 0.469
 GLN: Coeficiente Angular = 0.885

Resíduo LAC = 1.589325
 Resíduo GLU = 0.082728
 Resíduo GPV_e = 0.381899
 Coeficiente de Correlação = 0.962
 Coeficiente de Correlação = 0.963
 Coeficiente de Correlação = 0.983

Figura 86 - Simulação S2AcGPV com meio SF900 II, DO = 50% e temperatura = 28°C

Simulação 6 - S2MtGPV-H meio SF900 II e DO = 50% - Temperatura: 28°C
(VENTINI, 2010)

Resíduo X = 0.344583
 Resíduo GLC = 0.010614
 Resíduo GLN = 0.036063
 Xv: Coeficiente Angular = 0.377
 GLC: Coeficiente Angular = 0.570
 GLN: Coeficiente Angular = 0.769

Resíduo LAC = 1.700195
 Resíduo GLU = 0.190736
 Resíduo GPV_e = 0.680361
 Coeficiente de Correlação = 0.992
 Coeficiente de Correlação = 0.996
 Coeficiente de Correlação = 0.999

Figura 87 - Simulação S2MtGPV-H com meio SF900 II, DO = 50% e temperatura = 28°C

Simulação 7 - S2MtGPV-H meio SF900 II, DO = 50% e Temperatura: 28 - 22°C
(VENTINI, 2010)

Resíduo X = 0.172944

Resíduo GLC = 0.012829

Resíduo GLN = 0.171984

Xv: Coeficiente Angular = 1.486

GLC: Coeficiente Angular = 1.401

GLN: Coeficiente Angular = 1.400

Resíduo LAC = 2.548941

Resíduo GLU = 0.297963

Resíduo GPV_e = 0.798496

Coeficiente de Correlação = 0.981

Coeficiente de Correlação = 0.984

Coeficiente de Correlação = 0.978

Figura 88 - Simulação S2MtGPV-H com meio SF900 II, DO = 50% e temperatura = 28 - 22°C

Simulação 8 - S2MtGPV-H meio SF900 II, DO = 50 - 10% e Temperatura: 28°C
(VENTINI, 2010)

Resíduo X = 0.263624

Resíduo GLC = 0.002497

Resíduo GLN = 0.017312

Xv: Coeficiente Angular = 0.428

GLC: Coeficiente Angular = 0.713

GLN: Coeficiente Angular = 0.820

Resíduo LAC = 0.741858

Resíduo GLU = 0.107560

Resíduo GPV_e = 0.686380

Coeficiente de Correlação = 0.995

Coeficiente de Correlação = 0.991

Coeficiente de Correlação = 0.997

Figura 89 - Simulação S2MtGPV-H com meio SF900 II, DO = 50 - 10% e temperatura = 28°C

Simulação 9 - S2MtGPV-H meio SF900 II, DO = 50 - 10% e Temperatura: 28°C
(VENTINI, 2010)

Resíduo X = 0.166175

Resíduo GLC = 0.032687

Resíduo GLN = 0.220936

Xv: Coeficiente Angular = 0.510

GLC: Coeficiente Angular = 0.673

GLN: Coeficiente Angular = 1.302

Resíduo LAC = 1.938156

Resíduo GLU = 0.071496

Resíduo GPV_e = 0.714590

Coeficiente de Correlação = 0.991

Coeficiente de Correlação = 0.995

Coeficiente de Correlação = 0.946

Figura 90 - Simulação S2MtGPV-H com meio SF900 II, DO = 50 - 10% e temperatura = 28°C

Simulação 10 - S2AcGPV meio IPL-41 suplementado, DO = 40% (BATISTA, 2007)

Resíduo X = 0.141804

Resíduo GLC = 0.598381

Resíduo GLN = 1.878206

Resíduo LAC = 0.140238

Xv: Coeficiente Angular = 0.675

GLC: Coeficiente Angular = 0.977

GLN: Coeficiente Angular = 0.675

Resíduo NH4 = 0.108044

Resíduo ALA = 0.143956

Resíduo GPV_e = 4.457299

Coeficiente de Correlação = 0.937

Coeficiente de Correlação = 0.999

Coeficiente de Correlação = 1.000

Figura 91 - Simulação S2AcGPV com meio IPL-41, DO = 40%

Simulação 11 - S2AcGPV meio IPL-41 suplementado, DO = 5% (BATISTA, 2007)

Resíduo X = 0.075259

Resíduo GLC = 1.582109

Resíduo LAC = 1.801742

Xv: Coeficiente Angular = 1.013

GLC: Coeficiente Angular = 0.912

GLN: Coeficiente Angular = 1.000

Resíduo NH4 = 0.047597

Resíduo GPV_e = 4.391574

Coeficiente de Correlação = 0.978

Coeficiente de Correlação = 0.949

Coeficiente de Correlação = 1.000

Figura 92 - Simulação S2AcGPV com meio IPL-41, DO = 5%

Simulação 12 - S2AcGPV meio IPL-41 suplementado, DO = 20% (BATISTA, 2007)

Resíduo X = 0.082851
 Resíduo GLC = 0.897776
 Resíduo LAC = 2.717322
 x_v : Coeficiente Angular = 0.670
 GLC: Coeficiente Angular = 0.940

Resíduo NH₄ = 0.031684
 Resíduo GPV_e = 4.129689
 Coeficiente de Correlação = 0.931
 Coeficiente de Correlação = 0.905

Figura 93 - Simulação S2AcGPV com meio IPL-41, DO = 20%

Simulação 13 - S2AcGPV meio IPL-41 suplementado, DO = 60% (BATISTA, 2007)

Resíduo X = 0.230142

Resíduo GLC = 1.214372

Resíduo LAC = 3.154836

Xv: Coeficiente Angular = 0.685

GLC: Coeficiente Angular = 0.937

Resíduo NH4 = 0.045359

Resíduo GPV_e = 2.860087

Coeficiente de Correlação = 0.810

Coeficiente de Correlação = 0.962

Figura 94 - Simulação S2AcGPV com meio IPL-41, DO = 60%

Observa-se que a simulação para outros meios e construção mostra que o modelo obtido apresenta uma boa representatividade, pois tem-se um coeficiente de correlação médio entre esses treze ensaios de 0,96 em relação a previsão do comportamento do crescimento celular. Com relação a glicose e glutamina também tem-se uma boa correlação. A maior diferença foi observada em relação a formação de lactato, porém essa diferença pode ser influenciada pela presença de diferentes nutrientes nos diversos meios de cultura, uma vez que a modelagem foi executada para o meio TC100AG.

Outra diferença é pertinente à análise de GPV, pois o meio de cultura também interfere na formação da GPV.

Observa-se que para o ensaio com TC-100AG a resposta da simulação foi a melhor, apresentando os menores resíduos, pois o modelo foi feito com dados advindos de ensaios com esse meio. Já para os outros meios houve diferenças nas respostas metabólicas, mas nota-se que em meio SF 900 II, mesmo com construção diferente, apenas o lactato é que apresenta grandes diferenças. Com o meio IPL-41 suplementado o lactato também é o que possui maiores resíduos, porém o perfil de sua diferença é menor em comparação quando comparado com o SF 900 II. É importante observar-se que as diferenças de concentração de oxigênio dissolvido estão sendo bem respondidas pelo modelo.

Nessa extrapolação da atuação do modelo pode-se notar que foi melhor interpretada pelo modelo a diferença entre construção do que propriamente em relação à diferença de meios. Isso tem explicação devido a presença de diferentes metabólitos nos meios de cultura (meios mais ricos) o que influencia diretamente a dinâmica dos fluxos metabólicos.

Toda essa dinâmica executada neste trabalho pode ser melhorada e ampliada com a implantação do estudo do comportamento dos aminoácidos. Como neste trabalho não foi levada em conta a presença dos aminoácidos de maneira geral, um estudo mais aprofundado inserindo esses metabólitos no metabolismo poderá gerar uma modelagem mais completa, ou seja, baseando-se na presença de todos os metabólitos ter-se-á uma modelagem realmente independente do meio de cultura. Inclusive, levando-se em conta a presença dos aminoácidos, poder-se-á eventualmente modelar mais corretamente a quantificação da GPV.

5.10 MODELO METABÓLICO FINAL

Na Figura 95 é mostrado o modelo metabólico proposto, com as enzimas pertinentes, para a célula rS2, conforme discutido anteriormente.

Figura 95 - Proposição das vias metabólicas básicas para as células rS2.

6. CONCLUSÃO

Os estudos realizados durante este trabalho permitiram obter as conclusões abaixo descritas:

1. Foi proposto um modelo, com as seguintes características:

- Equacionamento através da utilização de modos elementares utilizando-se o Metatool;
- Metabolismo – glicólise, via das pentoses, ciclo de Krebs, fosforilação oxidativa, núcleo piruvato- α -cetoglutarato.
- Utilização de equações cinéticas de Michaelis-Menten;
- Parâmetros – velocidades específicas dos modos elementares e constantes de Michaelis-Menten – 43 parâmetros;
- Utilização de 7 faixas de concentração de glicose e glutamina. Cada faixa possui um conjunto de parâmetros;
- Validação – Teste F (para experimento sem repetição) = 16%;
- Coeficiente Angular = 0,95; Coeficiente de Correlação = 0,88;
- Extrapolação do modelo encontrado para outros 13 ensaios do grupo, com meios e construção diferentes dos ensaios utilizados para a modelagem. Essa extrapolação conseguiu resultados satisfatórios com baixos valores de resíduos.

2. Metabolismo:

- o crescimento celular sugere ser limitado a partir de valores inferiores a 1 g/L de glicose e também de glutamina;
- sugere-se que a via GOGAT é ativa nesta linhagem nas condições em que GLC está acima de 1 g/L;
- nas vias de degradação e síntese de glutamina, observa-se, em especial, a flexibilidade metabólica da célula S2 como uma importante área de mudança de reversibilidade dos fluxos metabólicos;

- proposição das principais vias metabólicas envolvidas do metabolismo das células rS2.
3. A sistemática de correlação entre quantidade de purinas formadas e crescimento celular mostrou-se um método inovador e interessante, pois conseguiu uma relação com resposta satisfatória para o perfil do crescimento celular.
 4. Contribuições tecnológicas geradas:
 - Software TrataDados: programa escrito em MATLAB que facilita a padronização, organização, armazenamento e o tratamento de dados experimentais e ainda a análise de dados que podem ser utilizados por pesquisadores de várias áreas de formação;
 - Software AjustaPar: programa escrito em MATLAB que apresenta uma gama diversa de possibilidades de otimização e também pode ser utilizado por vários pesquisadores a fim de facilitar a técnica de ajuste de parâmetros;

Ambos os programas são apresentados em formato executável e possuem interação com o usuário, inclusive para a inserção de seus próprios dados através de planilhas do Excel.
 5. O programa de simulação do comportamento cinético conseguiu abranger uma variada gama de ensaios (diferentes dos utilizados na modelagem) em condições distintas de meios e construção da célula, obtendo-se valores baixos de resíduos entre os dados experimentais e simulados.
 6. A utilização dos modos elementares como ferramenta para a análise metabólica mostrou-se eficiente para a implementação de uma modelagem.

7. TRABALHOS FUTUROS

Sugere-se que estudos complementares sejam realizados abordando basicamente:

- a) a introdução dos aminoácidos na modelagem metabólica;
- b) modelar mais corretamente a formação da GPV levando em conta os aminoácidos presentes no meio e também a sua formação e consumo dentro do metabolismo e também a disponibilidade de metabólitos;
- c) ampliação da atuação do TrataDados para a inserção dos aminoácidos;
- d) melhoria do Simulador do comportamento cinético para abranger os aminoácidos;
- e) verificação da existência da via GOGAT (experimentalmente).

REFERÊNCIAS*

- AGATHOS, S. N. Production scale insect cell culture. **Biotechnology Advances**, v.9, p. 51-68, 1991.
- AGUIAR, M. A. **Estudo Cinético de células de *Drosophila melanogaster* transfetadas para a produção da glicoproteína da raiva em biorreator**. Dissertação (Mestrado em Biotecnologia) - Instituto de Ciências Biomédicas, Universidade de São Paulo, São Paulo, 2010.
- AGUIAR, M. A.; BARRAL, M. F., PICCOLI, R. A. M.; LEO, P.; ASTRAY, R. M.; ANDRADE, R. J. Influência da Concentração de Oxigênio Dissolvido no Crescimento e Produção de Células de *Drosophila melanogaster* Transfetadas para a Expressão da Glicoproteína do Vírus da Raiva. In: SIMPOSIO NACIONAL DE BIOPROCESSOS – SINAFER. 2009. Natal. **Anais do XVII Simpósio Nacional de Bioprocessos**. Natal: UFRN. 1-5p. 2009.
- AIBA, S; MATSUOKA, M, **Identification of metabolic model: citrate production from glucose by *Candida lipolytica***. *Biotechnology and Bioengineering*, v.21, p.1373-1386, 1979.
- ALTAMIRANO, C.; GODIA, F.; CAIRÓ, J.J. Metabolismo de células de mamíferos cultivadas in vitro. In.: MORAES, A.M.; AUGUSTO, E.F.P.; CASTILHO, L.R.; **Tecnologia do cultivo de células animais: de biofármacos a terapia gênica**. São Paulo: Ed. Roca, p. 105-119, 2008.
- ALVES, P.M.M; CARRONDO, M.J.T.; CRUZ, P.E., Introdução à tecnologia de cultivo de células animais. In: In.: MORAES, A.M.; AUGUSTO, E.F.P.; CASTILHO, L.R.; BUTLER, M.; **Animal cell technology. From biopharmaceuticals to gene therapy**. London: Taylor & Francis, p. 2-14, 2008.
- AMABLE, P.; BUTLER, M. Cell metabolism and its control in culture. In.: MORAES, A.M.; AUGUSTO, E.F.P.; CASTILHO, L.R.; BUTLER, M.; **Animal cell technology. From biopharmaceuticals to gene therapy**. London: Taylor & Francis, p. 75-110, 2008.
- ANDREW, G. **Overview of Quasi-Newton optimization methods**. Acesso em: 10 mar. 2011. Disponível em: < <https://www.cs.washington.edu/homes/galen/files/quasi-newton-notes.pdf> >, 2008.

* De acordo com:

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO. Divisão de Biblioteca. **Diretrizes para apresentação de dissertação e teses**. 3ª ed. São Paulo: 2006. 103p. Disponível em: <<http://www.poli.usp.br/media/Biblioteca/Diretrizes3.pdf>>. Acesso em: 10 jan. 2011

ASTRAY, R. M.; AUGUSTO, E. F. P.; YOKOMIZO, A. Y.; PEREIRA, C. A. Analytical approach for the extraction of recombinant membrane viral glycoprotein from stably transfected *Drosophila melanogaster* cells. **Biotechnology Journal**, v. 3, p. 98-103, 2008.

AUGUSTO, E. F. P., BARRAL, M. F., PICCOLI, R. A. M. Modelos de crescimento e formação de produtos no cultivo de células animais. In: MORAES, A. M., AUGUSTO, E. F. P., CASTILHO, L. R. (Eds.), **Tecnologia do Cultivo de Células Animais: de Biofármacos a Terapia Gênica**. São Paulo: Roca, p. 170-215, 2008.

AUGUSTO, E. F. P.; BONOMI, A.; GIUDICI R. Estratégias para ajuste de parâmetros em modelos de processos fermentativos inibidos pelo substrato e produto. Estudo de casos. In: **Congresso Brasileiro de Engenharia Química**, 10. São Paulo. Anais, v.2, p. 1252-1257, 1994.

BACHMANN, A. S.; CORPUZ, G.; HARELD, W. P.; WANG, G.; COLLIER, B. A. A simple method for the rapid purification of copia virus-like particles from *Drosophila* Schneider 2 cells. **Journal of Virological Methods**, v.115, p.159-165, 2004.

BAILEY, J. E.; OLLIS, D. F. **Biochemical engineering fundamentals**, New York: McGraw-Hill International editions, 1986.

BAILEY, J. E. Mathematical modeling and analysis in biochemical engineering: past accomplishments and future opportunities. **Biotechnology Progress**, v.14 (1), p.8-20, 1998.

BANKS, D. J.; HUA, G.; ADANG, M. J. Cloning of a *Heliothis virescens* 110 kDa aminopeptidase N and expression in *Drosophila* S2 cells. **Insect Biochemistry and Molecular Biology**, v. 33, p. 499-508, 2003.

BARRAL, M. F. **Modelo cinético não-estruturado para crescimento e produção de glicoproteína recombinante do vírus da raiva em linhagem S2 de célula de *Drosophila melanogaster***. Tese (Doutorado). 205p. Programa de pós-graduação interunidades em Biotecnologia USP/Instituto Butantan/IPT. 2010.

BATISTA, F. R. X. **Formulação de meio de cultura livre de proteína animal para células de *Drosophila melanogaster* produtoras da glicoproteína G do vírus da raiva**. Tese (Doutorado em Engenharia Química). Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas. 2007.

BATISTA, F. R. X.; MORAES, A. M.; BÜNTEMEYER, H.; NOLL, T. Influence of culture conditions on recombinant *Drosophila melanogaster* S2 cells producing rabies virus glycoprotein cultivated in serum-free medium. **Biologicals**, v. 37, p. 108-118, 2009.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear programming: Theory and Algorithms**. 3rd ed. New Jersey: John Wiley & Sons, 1979. Acesso em:

18 ago. 2012. Disponível para consulta on-line em: < <http://books.google.com.br> >, 1979.

BENSLIMANE, C.; ELIAS, C. B.; HAWARI, J.; KAMEN, A. Insights into the Central Metabolism of *Spodoptera frugiperda* (Sf-9) and *Trichoplusia ni* BTI-Tn-5B1-4(Tn-5) Insect Cells by Radiolabeling Studies. **Biotechnology Progress**, v. 21, p. 78-86, 2005.

BENTING, J.; LECAT, S.; ZACCHETTI, D.E.; SIMONS, K. Protein expression in *Drosophila* Schneider cells. **Analytical biochemistry**. v.278, p.59-68, 2000.

BÉRDARD, C.; TOM, R.; KAMEM, A. Growth, nutrient consumption, and end product accumulation in Sf9 and BTI-EAA insect cell cultures: insights into growth limitation and metabolism. **Biotechnology progress**. v. 9, p. 615-624, 1993.

BERNAL, V.; CARINHAS, N.; YOKOMIZO, A. Y.; CARRONDO, M. J.; ALVES, P. M. Cell density effect in the baculovirus-insect cells system: A quantitative analysis of energetic metabolism. **Biotechnology and Bioengineering**, v.104, p. 162-180, 2009.

BEVERIDGE, G.; SCHECHTER, R. S. **Optimization: Theory and Practice**. New York: McGraw-Hill, 1970.

BHATIA, R.; JESIONOWSKI, G.; FERRANCE, J.; ATTAI, M.M. Insect cell physiology. **Cytotechnology**. v.24, p.1-9, 1997.

BOGGS P. T.; TOLLE, J.W. Sequential Quadratic Programming. **Acta Numerica**, v. 4, p. 1-51, 1995.

BONOMI, A.; FLEURY, A. T.; PICCOLI, R. A.; QUEIROZ, L. H.C.; PEREIRALIMA, P.S.; TANNURI, E. A.; MATTOS, M. N. Modelagem matemática, otimização e controle avançado do processo de produção de plástico biodegradável a partir do açúcar por via fermentativa. In: **Congresso de Equipamentos e Automação da Indústria Química**, 4. São Paulo, Anais, São Paulo, 1999.

BONOMI, A.; SCHMIDELL, W., **Modelagem Matemática e Simulação de Processos Fermentativos**. In: SCHMIDELL, W.; LIMA, U. A.; AQUARONE, E.; BORZANI, W. (Org.). **Biotecnologia Industrial**. 1 ed. São Paulo: Edgard Blücher, v. 2, p. 123-178, 2001.

BORASH, D. J.; PIERCE, V. A.; GIBBS, A. G.; MUELLER, L. D. Evolution of ammonia and urea tolerance in *Drosophila melanogaster*: resistance and cross-tolerance. **Journal of Insect Physiology**, v. 46, p. 763-769, 2000.

BOUTILIER, R.G.. Mechanisms of cell survival in hypoxia and hypothermia. **Journal of Experimental Biology**, v. 204, p. 3171-3178, 2001.

BOVO, R.; GALESI, A. L. L.; JORGE, S. A. C.; PICCOLI, R. A. M.; MORAES, A. M.; PEREIRA, C. A.; AUGUSTO, E. F. P. Kinetic response of a *Drosophila melanogaster* cell line to different medium formulations and culture conditions. **Cytotechnology**, v. 57, p. 23-35, 2008.

BRIGHTY, D.W.; ROSENBERG, M.; CHEN, I.S.Y.; IVEY-HOYLE, M. Envelope proteins from isolates of human immunodeficiency virus type-I that are refractory to neutralizations by soluble CD4 receptor. In: National academy of sciences. 1991. USA. **Proc. natl. academy science USA**. v. 88, p. 7802-7805, 1991.

BROYDEN, C.G., The Convergence of a Class of Double-rank Minimization Algorithms. **Journal Institute Mathematics. Applications**, v. 6, p. 76–90, 1970.

BUCKINGHAM, S.D.; MATSUDA K.; HOISE, A.M.; BAYLIS, H.A.; SQUIRE, M.D.; LANSDELL, S.J.; MILLAR, N.S.; SATELLE, D.B. Wild-type and insecticide resistant homo-oligomeric GABA receptors of *Drosophila melanogaster* stably expressed in a *Drosophila* cell line. **Neuropharmacology**. v. 35, p. 1393-1401, 1996.

BYRD, R. H., SCHNABEL, R. B.; SHULTZ, G. A., Approximate Solution of the Trust Region Problem by Minimization over Two-Dimensional Subspaces. **Mathematical Programming**, v. 40, p. 247–263, 1988.

BYRD, R. H.; GILBERT, J. C.; TOINT, N. A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming. **Mathematical Programming**, v. 89, n. 1, p. 149–185, 2000.

BYRD, R. H.; HRIBAR, M. E.; NOCEDAL, J. An Interior Point Algorithm for Large-Scale Nonlinear Programming. **SIAM Journal on Optimization**, v. 9, n. 4, p. 877–900, 1999.

CASTILLO, I. D. S. **Estudo do processo de cultivo de células de hibridoma: modelagem matemática e otimização preliminar do processo**. Tese (Doutorado em Engenharia Química). Escola Politécnica da Universidade de São Paulo. 2002.

CCAUFES – **Centro de ciência agrária da universidade federal do Espírito Santo**. Disponível em: <[http:// www.cca.ufes.br](http://www.cca.ufes.br)>. Acesso em: 18 ago. 2012..

CHA, H. J.; SHIN, H. S.; LIM, H. J.; CHO, H. S.; DALAL, N. N.; PHAM, M. Q.; BENTLEY, W. E. Comparative production of human interleukin-2 fused with green fluorescent protein in several recombinant expression systems. **Biochemical Engineering Journal**, v. 24, p. 225-233, 2005.

CHERBAS, L.; MOSS, R.; CHERBAS, P. Transformation techniques for *Drosophila* cell lines. **Methods in cell biology**. v.44, p.161-179, 1994.

CHIERICHETTI, F.; LACROIX, V.; MARCHETTI-SPACCAMELA, A.; SAGOT, M-T; STOUGIE, L., **Modes and cuts in metabolics networks: Complexity and algorithms**. Biosystems, 2009.

COELHO, S. Método permite encontrar seqüências ocultas do genoma. In: **Ciência hoje**. São Paulo: 19.02.2002.

COLEMAN, T.F.; LI, Y. An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. **SIAM Journal on Optimization**, v. 6, p. 418–445, 1996.

CONN, N. R.; GOULD, N. I. M.; TOINT, P. L. Trust-Region Methods, MPS/SIAM Series on Optimization, **SIAM and MPS**, 2000.

DAVIDON, W.C., Variable Metric Method for Minimization. **A.E.C. Research and Development Report**, ANL-5990, 1959.

DELM, L.; WOLF, H.; WAGNER, R.. Hight level expression of hepatitis B virus superface antigen in stably transfected *Drosophila* Schneider 2 cells. **Journal virology methods**. v.79, p.191-203, 1999.

DEML L.; SCHIRMBECK, R.; REIMANN, J.; WOLF, H.; WAGNER, R. Purification and characterization of hepatitis B virus surface antigen particles produced in *Drosophila* Schneider-2 cells. **Journal of Virological Methods**, v.79, p.205-217, 1999.

DOVERSKOG, M.; HAN, L.; HÄGGSTRÖM, L. Cystine/cysteine metabolism in cultured Sf9 cells: Influence of cell physiology on biosynthesis, amino acid uptake and growth. **Cytotechnology**, v. 26, p. 91-102, 1998.

DOVERSKOG, M.; JACOBSON, U.; CHAPMAN, B. E.; KUCHEL, P. W.; HÄGGSTRÖM, L. Determination of NADH-dependent glutamate synthase (GOGAT) in *Spodoptera frugiperda* (Sf9) insect cells by a selective ¹H/¹⁵N NMR in vitro assay. **Journal of Biotechnology**, v. 79, p. 87-97, 2000.

DREWS, M.; DOVERSKOG, M.; ÖHMAN, L.; CHAPMAN, B. E.; JACOBSSON, U.; KUCHEL, P.W.;HAGGSTROM, L. Pathways of glutamine metabolism in *Spodoptera frugiperda* (Sf9) insect cells: evidence for the presence of the nitrogen assimilation system, and a metabolic switch by ¹H/¹⁵N NMR. **Journal of Biotechnology**, v. 78, p. 23-37, 2000.

DREWS, M.; PAALME, T.; VILU, R. The growth and nutrient utilization of the insect cell line *Spodoptera frugiperda* Sf9 in batch and continuous culture. **Journal of biotechnology**. v. 40, p. 187-198, 1995.

EDGAR, T.F.; HIMMELBLAU, D.M. **Optimization of Chemical Processes**. New York: McGraw – Hill, 1988.

ENGASSER, J.M., Bioreactor engineering the design and optimization of reactors with living cells. **Chemical Engineering Science**, v. 43(8), p.1739-1748, 1988.

FEALA, J. D.; COQUIN, L.; ZHOU, D.; HADDAD, G. G.; PATERNOSTRO, G.; McCULLOCH, A. D. Metabolism as means for hypoxia adaptation: metabolic profiling and flux balance analysis. **BMC Systems Biology**, v. 3, p. 91-106, 2009.

FERRANCE J.P.; GOEL A.; ATAAL M.M. Utilization of glucose and amino acids in insect cell cultures: quantifying the metabolic flows within the primary pathways and medium development. **Biotechnology Progress**. v. 42, p.697-707, 1993.

FERRAZ, L. P. C. **Proposição de um modelo fenomenológico estruturado cibernético para o processo de produção de copolímeros de polihidroxialcanoatos por via fermentativa**. Dissertação (Mestrado em Engenharia Química). Escola Politécnica da Universidade de São Paulo. 1999.

FLETCHER, R. **Practical Methods of Optimization**. John Wiley and Sons, 1987.

FLETCHER, R., A New Approach to Variable Metric Algorithms. **Computer Journal**, v. 13, p. 317–322, 1970.

FLETCHER, R.; POWELL, J. D., A Rapidly Convergent Descent Method for Minimization. **Computer Journal**, v. 6, p. 163–168, 1963.

FOLLISTAD, B.D.; BALCAREL, R.R.; STEPHANOPOULOS, G.; WANG, D.I.; **Metabolic flux analysis of hibridoma continuous culture steady state multiplicity**. *Biotechnology Bioengineering*, v. 63, p. 675-683, 1999.

FREDRICKSON, A.G.; TSUCHIYA, H.M. Microbial kinetics and dynamics. em: LAPIDUS, L.; AMUNDSON, N.R., **Chemical reactor theory: review**. Englewood Cliffs: Prentice – Hall, 1997.

FRESHNEY, R.I. **Culture of Animal Cells: A Manual of Basic Technique**, 3^a ed, Wiley-Liss, 1994.

GALES, A. L. L. **Cultivo de células de *Drosophila melanogaster* em diferentes formulações de meios de cultura livres de soro visando a produção da glicoproteína do vírus da raiva**. 177 f. Tese (Doutorado em Engenharia Química). Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, 2007.

GALES, A. L. L.; AGUIAR, M. A.; ASTRAY, R. M.; AUGUSTO, E. F. P.; MORAES, A. M. Growth of recombinant *Drosophila melanogaster* Schneider 2 cells producing rabies virus glycoprotein in bioreactor employing serum-free medium. **Cytotechnology**. v. 57, p. 73-81, 2008.

GARRET, R.; GRISHAM, C., **Biochemistry**. Saunders College Publishing, 1999.

GEMANO, P.M.L. Avanços na pesquisa da raiva. **Revista de saúde pública**, v.28, p.38-48, 1994.

GIBSON, K. R.; VANEK, P. G.; KALOSS, W. D.; COLLIER, B.; CONNAUGHTON, J. F.; MONICA ANGELICHIO, M.; LIVI, G. P.; FLEMING, P. J. Expression of Dopamine β -hydroxylase in *Drosophila* Schneider 2 cells. Evidence for a mechanism of membrane binding other than uncleaved signal peptide. **The Journal of Biological Chemistry**, v. 268, p. 9490-9495, 1993.

GILL, P.E.; MURRAY, W; WRIGHT, M. H. **Practical Optimization**, London, Academic Press, 1981.

GOCKENBACH, M. S. **Introduction to sequential quadratic programming**. Acesso em: 10 mar. 2011. Disponível em: < <http://www.math.mtu.edu/~msgocken/ma5630spring2003/lectures/sqp1/sqp1.pdf> >, 2003.

GOLDFARB, D., A Family of Variable Metric Updates Derived by Variational Means. **Mathematics of Computing**, v. 24, p. 23–26, 1970.

GOMBERT, A. K.; NIELSEN, J. Mathematical modeling of metabolism. **Current Opinion in Biotechnology**, v.11, p.180-186. 2000.

GOTOH, T.; MIYAZAKI, Y.; CHIBA, K.; KIKUCHI, K. Significant increase in recombinant protein production of a virus-infected Sf9 insect cell culture of low MOI under low dissolved oxygen conditions. **Journal of Bioscience and Bioengineering**, v. 94, p. 426-433, 2002.

GRACE, T.D.C. Analysis of the aminoacids and sugar in an insect cell medium during cell growth. **Journal insect physiology**. v.12, p.625-633, 1966.

GROSAN, C.; ABRAHAM, A.; Modified Line Search Method for Global Optimization. Modelling & Simulation, AMS '07. **First Asia International Conference**, p. 415-420, 2007.

GUY, E. *Drosophila* cell in culture. San Diego, **Academy press**. p. 702, 1997.

HEINZLE, E.; SANER, U. Methodology for process control in research and development. In: Pons, M.N. (Ed) **Bioprocess Monitoring and Control**, Hanser, Munique, 1991.

HILL, R. M., BRENNAN, S. O., BIRCH, N. P. Expression, purification, and functional characterization of serine protease inhibitor neuroserpin expressed in *Drosophila* S2 cells. **Protein Expression and Purification**, v. 22, p. 406-413, 2001.

HIMMELBLAU, D. M. **Process Analysis by Statistical Methods**. New York: John Wiley & Sons, 1970.

HIMMELBLAU, D. M. **Applied Nonlinear Programming**. New York: McGraw-Hill, 1972.

HISS, H. Cinética de Processos Fermentativos. In: SCHMIDELL, W.; LIMA, U. A.; AQUARONE, E.; BORZANI, W. (Org.). **Biotecnologia Industrial**. 1 ed. São Paulo: Edgard Blücher, v. 2, p. 93-122, 2001.

HOCK, W.; SCHITTKOWSKI, K. A Comparative Performance Evaluation of 27 Nonlinear Programming Codes. **Computing**, v. 30, p. 335, 1983.

HOLMS, N.H., **The central metabolic pathways of *Escherichia coli*: relationship between flux and control at a branch point, efficiency of conversion to biomass and excretion of acetate**. Current Topics in Cellular Regulation, v.28, p.69-105, 1986.

HOOKE, R.; JEEVES, T., **Direct Search Solution of numeral and Statistical Problems**, **Journal of the ACM**, v. 8, n. 2, p. 212-229, 1961.

IKONOMOU, L.; SCHNEIDER, Y-J; AGATHOS, S.N. Insect cell culture for industrial production of recombinant proteins. **Applied microbiology and biotechnology**. v.62, p.1-20, 2003.

JARVIS, D. L., KAWAR, Z. S., HOLLISTER, J. R. Engeneering N-glycosylation pathways in the baculovirus-insect cell system. **Current Opinion in Biotechnology**, v. 9, p. 528-533, 1998.

JEON, H. K.; CHANG, K. H.; KIM, K. I.; CHUNG, I. K. Functional expression of recombinant tumstatin in stably transformed *Drosophila melanogaster* S2 cells. **Biotechnology Letters**, v.25, p.185-189, 2003.

KAHANER, D. **Numerical Methods and Software**. New Jersey: Prentice Hall, 1988.

KAMP, A. V.; SHUSTER, S. Metatool 5.0: fast and flexible elementary modes analysis. **Bioinformatics Applications Note**, v. 22, n. 15, p. 1930-1931, 2006.

KEGG - **Kyoto Encyclopedia of Genes and Genomes**. Disponível em: < http://www.genome.jp/kegg-bin/show_pathway?query=&map=dme01100&scale=0.5&show_description=show&multi_query= > Acesso em: 18 ago. 2012..

KIOUKIA, N.; NIENOW, A. W.; EMERY, A. N.; ALBUREAI, M. Physiological and environmental-factors affecting the growth of insect cells and infection with baculovirus. **Journal of biotechnology**. v.38, p.243-251, 1995.

KIRKPATRICK, R. B.; GANGULY, S.; ANGELICHIO, M.; GRIEGO, S.; SHATZMAN, A.; SILVERMAN, C.; ROSENBERG, M. Heavy chain dimmers as well as complete antibodies are efficiently formed and secreted from *Drosophila* via a BiP-mediated pathway. **Journal of Biological Chemistry**, v. 270, p. 19800-19805, 1995.

KLAMT, S.; GILLES, E. D. Minimal cut sets in biochemical reaction networks. **Bioinformatics**, v. 20, p. 226-234, 2004.

KLAMT, S.; STELLING, J.; GINKEL, M.; GILLES, E. D. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. **Bioinformatics**, v. 19, p. 261-269, 2003.

KUNAS, K. T.; PAPOUTSAKIS, E. T. Damage mechanisms of suspended animal cells in agitated bioreactors with and without bubble entrainment. **Biotechnology and Bioengineering**, v. 36, p. 476-483, 1990.

LAGARIAS, J. C.; REEDS, J. A.; WRIGHT, M. H.; WRIGHT, P. E. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. **SIAM Journal on Optimization**, v. 9, n. 1, p. 112-147, 1998.

LARNER, J.; **Metabolismo Intermediário e sua Regulação**. Editora Edgard Blücher, 1974.

LE DUY, A.; ZAJIC, J.E. A geometrical approach for differentiation of an experimental function at a point: applied to growth and product formation. **Biotechnology and Bioengineering**. vol. 15, n. 4, p. 805-810, 1973.

LEE, J.M., PARK, J.H.; PARK, J. O.; CHANG, K. H.; CHUNG, I. S. Expression of recombinant erythropoietin in stably transformed *Drosophila melanogaster* S2 cell. **In vitro cell dev. biology**. v.3, p.348-350, 2000.

LEHNINGER, A.; NELSON, D.; FOX, M., **Principles of Biochemistry**. Sarvier, ed.4, 2004.

LEMONS, M.A.N, **Construção e transfeção de vetores plasmidiais contendo o gene da glicoproteína do vírus da raiva (GPV) em células de *Drosophila melanogaster***. Tese de mestrado no Programa Interunidades em Biotecnologia. USP/Instituto Butantan/IPT, 2009.

LEYFFER, S.; MAHAJAN, A. **Nonlinear Constrained Optimization: Methods and Software**. Disponível em: < <http://wiki.mcs.anl.gov/leyffer/images/7/75/NLP-Solvers.pdf> > Acesso em: 18 ago. 2012..

LI, B.; TSING, S.; KOSAKA, A.H.; NGUYEN, B.; OSEN, E.G.; BACH, C.; CHAN, H.; BARNETT, J. Expression of human dopamine β hydroxylase in drosophila Schneider 2 cells. **Biochemical journal**. v.313, p.57-64, 1996.

LIM, H. J., CHA, H. J. Observation and modeling of induction effect on human transferring production from stably transfected *Drosophila* S2 cell culture. **Enzyme and Microbial Technology**, v. 39, p. 208-214, 2006.

MARQUARDT, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. **SIAM Journal Applied. Mathematics**, v. 11, n. 2, p. 431-441, 1963.

MARTINELLE, K.; DOVERSKOG, M.; JACOBSSON, U.; CHAPMAN, B. E.; KUCHEL, P. W.; HÄGGSTRÖM, L. Elevated glutamate dehydrogenase flux in glucose-deprived hybridoma and myeloma cells: Evidence from $^1\text{H}/^{15}\text{N}$ NMR. **Biotechnology and Bioengineering**, v. 60, p. 508-517, 1998.

MARZZOCO, A.; TORRES, B. B. **Bioquímica Básica**. Editora Guanabara, Koogan S.A., RJ, 1990.

MATHWORKS – Disponível em: < <http://www.mathworks.com/products/matlab> > 2009, acesso em: 18 ago. 2012.

MATHWORKS. **Curve Fitting Toolbox 3 – User's Guide**. Massachusetts: The MathWorks, inc., 2010. Acesso em: 18 ago. 2012. Disponível em: < http://www.mathworks.com/help/pdf_doc/curvefit/curvefit.pdf >, 2010a.

MATHWORKS. **MATLAB Function Reference**. Massachusetts: The MathWorks, inc., 2001. Acesso em: 18 ago. 2012. Disponível em: < <http://maxwell.me.gu.edu.au/spl/matlab-page> >, 2001.

MATHWORKS. **Optimization Toolbox 5 – User's Guide**. Massachusetts: The MathWorks, inc., 2010. Acesso em: 18 ago. 2012. Disponível em: < http://www.mathworks.com/access/helpdesk/help/pdf_doc/optim/optim_tb.pdf >, 2010b.

MCCARROLL, L.; KING, L. Stable insect cell cultures for recombinant protein production. **Current opinion in biotechnology**. v. 8, p. 590-594, 1997.

MEHROTRA, S. On the Implementation of a Primal-Dual Interior Point Method. **SIAM Journal on Optimization**, v. 2, p. 575–601, 1992.

MENDES, P. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. **Trends Biochem Sci**, v. 22, p. 361-363, 1997.

MENDONÇA, R.Z.; PALOMARES, L.A.; RAMÍREZ, O.T. An Insight into Insect Cell Metabolism Through Selective Nutrient Manipulation. **Journal of biotechnology**. v. 72, p. 61-75, 1999.

MITSUHASHI, J. In: **Invertebrate Cell System Applications**, Mitsuhashi J. ed., p. 3-20, CRC Press, 1989.

MOLER, C., Numerical Computing with MATLAB, Edição eletrônica: **The MathWorks**, Inc., Natick, MA, A. Disponível em: < <http://www.mathworks.com/moler> > Acesso em: 16 fev. 2009. Edição impressa: SIAM, Philadelphia, 2004.

MONOD, J. The growth of bacterial cultures. **Ann. Review of Microbiology**, v. 3, p.371-394, 1949.

MORAES, A.M.; JORGE, S.A.C.; ASTRAY, R.M.; SUAZO, C.A.T.; RIQUELME, C.E.C.; AUGUSTO, E.F.P.; TONSO, A.; PAMBOUKIAN, M.M.; PICCOLI, R.A.M.; BARRAL, M.F.; PEREIRA, C.A., **Drosophila melanogaster S2 cells for expression of heterologous gene cloning to bioprocess development**, Biotechnology Advances, v. 30, p. 613-628, 2012.

MORÉ, J. J.; SORENSEN, D. C. Computing a Trust Region Step. **SIAM Journal on Scientific and Statistical Computing**, v. 3, p. 553–572, 1983.

MOSER, A. Kinetics of batch fermentation. In: REHM, H. J.; REED, G. Volume Editor: Brauer, H. **Biotechnology – A comprehensive treatise in 8 volumes**. Weinheim, V. C. H. Verlagsgesellschaft, v. 2, p: 243-283, 1985.

MOTTA, V.T., **Bioquímica**, ed. 2, Medbook, 2011.

NC-IUBMB - **Nomenclature Committee of the International Union of Biochemistry and Molecular Biology** - <http://www.chemqmul.ac.uk/iubmb/enzyme/rules.html>. Acessado em: 18/08/2012.

NEERMANN, J.; WAGNER, R. Comparative analysis of glucose and glutamine metabolism in transformed mammalian cell lines, insect and primary liver cells. **Journal of Cellular Physiology**, v. 166, p. 152-169, 1996.

NELDER, J. A.; MEAD, R. A simplex method for function minimization. **The Computer Journal**, v. 7, n. 4, p. 308-313, 1965.

NIELSEN, J., **Metabolic engineering techniques for analysis of target for genetic manipulation**. Biotechnology and Bioengineering, v58, p.125-132, 1998.

NOCEDAL, J.; WRIGHT, S. J. **Numerical Optimization**, Second Edition. Springer Series in Operations Research, Springer Verlag, 2006.

OEUVRES. Mémoire sur les approximations des formules qui sont fonctions de très-grands nombres, et sur leur application aux probabilités (suite). **Mémoires l'Institut** 353-415, 559-565. 12 p.301-345, p.349-353, 1810.

ÖHMAN, L.; ALARCON, M.; LJUNGGREN, J.; RAMQVIST, A.K; HAGGSTROM, L. Glutamine is not an essential amino acid for Sf9 Insect Cell. **Biotechnology letters**. v. 18, p. 756-770, 1996.

ÖHMAN, L; LJUNGGREN, J.; HAGGSTROM, L. Induction of a metabolic switch in insect cell by substrate-limited fed batch cultures. **Applied microbiology and biotechnology**. v.43, p.1006-1013, 1995.

OLSSON, D. M.; NELSON, L. S. The Nelder-Mead Simplex Procedure for Function Minimization. **Technometrics**, v. 17, n. 1, p. 45-51, 1975.

OZTURK, S. S.; PALSSOM, B. O. Effects of dissolved oxygen on hybridoma cell growth, metabolism and antibody production kinetics on continuous culture. **Biotechnology Progress**, v. 6, p. 437-446, 1990.

PAMBOUKIAN, M. M.; JORGE, S. A. C.; SANTOS, M. G.; YOKOMIZO, A. Y.; PEREIRA, C. A.; TONSO, A. Insect cells respiratory activity in bioreactor. **Cytotechnology**, v. 57, p. 37-44, 2008.

PAMBOUKIAN, M.M.; PEREIRA, C.A.; TONSO, A.; **Metodologia de cálculo de C_{crit} em células animais**, SINAFERM 2009, Curitiba. Anais. CD-Rom, 2009.

PAPOUTSAKIS, E.T.; MEYER, C.L., **Equations and calculations of products yeilds and preferred pathways for butanediol and mixed-acid fermentations**. Biotechnology and Bioengineering, v.27, p.50-66, 1985.

PÉREZ, O.; PAOLAZZI, C.C. Production methods for rabies vaccine. **Journal of industrial microbiology and biotechnology**. v.18, p.340-347, 1997.

PFEIFER, T.A.; HEGEDUS, D.D.; GRIGLIATTI, T.A.; THEILMANN, D.A. Baculovirus immediate-early promoter-mediated expression of the Zeocin resistance gene for use as a dominant selectable marker in dipteran and lepidopteran insect cell lines. **Gene**. v. 188, p. 183-190, 1997.

PFEIFFER, T.; SANCHEZ-VALDENEBRO, I.; NUNO, J. C.; MONTERO, F.; SCHUSTER, S. METATOOL: for studying metabolic networks. **Bioinformatics**. v. 15, n. 3, p. 251-257, 1999.

PICCOLI, R. A. M. **Otimização do processo de produção de copolímeros de polihidroxialcanoatos por via fermentativa, baseada num modelo matemático estruturado** Tesse (Doutorado em Engenharia Química). Escola Politécnica da Universidade de São Paulo. 2000.

PICCOLI, R. A. M.; BATISTA, F. R. X.; MORAES, A. M.; BARRAL, M. F.; AGUIAR, M. A.; LEO, P.; AUGUSTO, E. F. P. Modelagem matemática para células de *Drosophila melanogaster* sob condições de crescimento balanceado. SINAFERM 2009. Curitiba. Anais. CD-ROM. 2009.

PICCOLI, R. A. M.; VALDEZ, W. P.; AUGUSTO, E. F. P.; BONOMI A. Otimização de técnica de poliedros flexíveis aplicada ao ajuste de parâmetros de modelos matemáticos de processos fermentativos. In: **Encontro de Áreas Técnicas da Divisão de Química**, IPT, 1995.

PIVETTA, M., Sequência inchada. Pesquisa on line. FAPESP, ed. 136, junho, 2007.

PORTAL DA SAÚDE, http://portal.saude.gov.br/portal/arquivos/pdf/mapa_raiva_final_2011.pdf. Acessado em: 18/08/2012

POWELL, M. J. D. Variable Metric Methods for Constrained Optimization. **Mathematical Programming: The State of the Art**, (A. Bachem, M. Grottschel and B. Korte, eds.) Springer Verlag, p. 288–311, 1983.

PROVOST, A.; BASTIN, G. Dynamic metabolic modeling under the balanced growth condition, **Journal of Process Control**, v. 14, p. 717-728, 2004.

PROVOST, A.; BASTIN, G.; AGATHOS, S. N.; SCHNEIDER, Y. J.; Metabolic design of macroscopic bioreaction models: application to Chinese hamster ovary cells. **Bioprocess Biosystem Engineer**, v. 29, p. 349-366, 2006.

QUINTANA, E.; ALEIXANDRE, J. G.; ARTIME, J.; PEIRE, J. A Derivative-free method to minimize torque pulsations in P.W.M. Inverter Drives. **Industrial Electronics Society**. IECON '88. Proceedings, 14 Annual Conference of, v. 3, p. 798-803, 1988.

RAICHER, G., **Análise Econômica da Produção de Polímeros Biodegradáveis no contexto de uma Biorrefinaria a partir de cana-de-açúcar**. Tese de doutorado no Programa Interunidades em Biotecnologia. USP/Instituto Butantan/IPT, 2011.

RATTENBURY, N., Almost Runge-Kutta Methods for Stiff and Non-Stiff Problems. Tese de Doutorado, The University of Auckland, 2005.

RHIEL, M.; MITCHELL-LOEGEAN, C. M.; MURHAMMER, D. W. Comparison of *Trichoplusia ni* BTI-Tn-5B1-4 (High Five) and *Spodoptera frugiperda* Sf9 insect cell line metabolism in suspension cultures. **Biotechnology and Bioengineering**, v. 55, p. 909-920, 1997.

ROCHA, I; MAIA, P.; EVANGELISTA, P.; VILAÇA, P.; SOARES, S.; PINTO, J. P.; NIELSEN, J. OptFlux: na open-source software platform for *in silico* metabolic engineering. **BMC Systems Biology**, v. 4, p. 45, 2010.

SAGOT, M-F. **An introduction to metabolic networks and their structural analysis**. Journal Transactions and Computational Biology and Bioinformatics, v.5 issue 4, p. 594-617, 2008.

SANTOS, M. G. **Expressão do gene da proteína verde fluorescente (EGFP) em células de *Drosophila melanogaster* como instrumento de otimização de bioprocesso para a produção de proteínas heterólogas**. 2006. 59p. Dissertação (Mestrado). Programa de pós-graduação interunidades em Biotecnologia USP/Instituto Butantan/IPT. 2006.

SCHATZMAYR, H.G. **Emerging and reemerging viral diseases**. Caderno de Saúde Publica. v. 17, Suppl., 2001.

SCHITTKOWSKI, K. NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems. **Annals of Operations Research**, v. 5, p. 485-500, 1985.

SCHITTKOWSKI, K. Sequential **Quadratic Programming Methods with Distributed and Non-Monotone Line Search**. Disponível em: < http://www.dynardo.de/fileadmin/Material_Dynardo/bibliothek/WOST_1.0/WOST_1_SequentialQuadraticProgrammingMethods_En.pdf > Acesso em: 18 ago. 2012.

SCHNEIDER, I. Cell lines derived from late embryonic stages of *Drosophila melanogaster*. **Journal embryologic exp. morphologic**, v.27, p.363-365, 1972.

SCHOENBERG, R. Optimization **with the Quasi-Newton Method**. 2001. Disponível em: < <http://www.aptech.com/papers/qnewton.pdf> > Acesso em: 10 jan. 2011.

SCHUSTER, S.; FELL, D.A.; DANDEKAR, T. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. **Nature Biotechnology**, v. 18, p. 326–332, 2000.

SCHUSTER, S.; PFEIFFER, T.; MOLDENHAUER, F.; KOCH, I.; DANDEKAR, T. Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. **Bioinformatics**, v. 18, p. 351-361, 2002.

SCHWARZ, R.; MUSCH, P.; KAMP, A. B.; ENGELS, B.; SCHIRMER, H.; SCHUSTER, S.; DANDEKAR, T. YANA – a software tool for analyzing flux modes, gene-expression and enzyme activities. **Bioinformatics**, v. 6, p. 135, 2005.

SCOTT, T. R. I.; BLANCHARD, J. H.; FERGUNSON, C. H. R. Effects of oxygen on recombinant protein production by suspension cultures of *Spodoptera frugiperda* (Sf9) insect cells. **Enzyme Microbiology and Technology**, v. 14, p. 798-804, 1992.

SVS/MS, Secretaria de vigilância em saúde, Ministério da Saúde, http://portal.saude.gov.br/portal/arquivos/pdf/tab_casos_raiva_2012.pdf, Acesso em: 18 ago. 2012.

SERMUTLU, E. Comparison of Runge-Kutta methods of order 4 and 5 on Lorenz equation. **Journal of Arts and Sciences**, v. 1, p. 61-69, 2004.

SESHACHALAM, R. V.; SUBRAMANYAM, V. V.; KRISHNAMOORTHY, R. V. New pathway of utilization of ammonia nitrogen for the synthesis of amino acids through NADH dependent transaminases in *Bombyx mori*. **Physiological Entomology**, v. 17, p. 281-287, 1992.

SHANNO, D. F., Conditioning of Quasi-Newton Methods for Function Minimization. **Mathematics of Computing**, v. 24, p. 647–656, 1970.

SHIN, H. S.; CHA, H. J. Statistical optimization for immobilized metal affinity purification of secreted human erythropoietin from *Drosophila* S2 cell. **Protein Expression and Purification**, v. 28, p. 331-339, 2003.

SONDERGAARD, L. *Drosophila* cells can grown to high cell densities in a biorreator. **Biotechnology Techniques**, v. 10, p. 161-166, 1996.

SPENDLEY, N.; HEXT, G. R.; HIMSWORTH, F. R. Sequential Application of Simplex Designs in Optimization and Evolutionary Operation. **Technometrics**, v. 4, p. 441-461, 1962.

STEihaug, T., The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. **SIAM Journal on Numerical Analysis**, v. 20, p. 626-637, 1983.

STRAVOULAKIS, D. A.; KALOGERAKIS, N.; BEHIE, L. A. Growth characteristics of *Bombyx mori*. Insect cell line in stationary and suspension culture. **Canadian journal of chemical engineering**, v. 69, p. 457-464, 1991.

STREET, J. C.; DELORT, A. M.; BRADDOCK, P. S. H.; BRINDLE, K. M. A ¹H/¹⁵N n.m.r. study of nitrogen metabolism in cultured mammalian cells. **Biochemical Journal**, v. 291, p. 485-492, 1993.

SUGIURA, T.; AMANN, E. Properties of two insect cell lines useful for the baculovirus expression system in serum-free culture. **Biotechnology and Bioengineering**, v. 51, p. 494-499, 1996.

SUMMERS, M.D.; SMITH, G. E. A manual of methods for baculovirus vectors and insect cell culture procedure. **Texas agric. experiment station**. v.9, p.1-56, 1987.

SUZUKI, K.; SMOLENSKI, R. T.; JAYAKUMAR, J. F. R. C. S.; MURTUZA, B. M. A.; BRAND, N. J.; YACOUB, M. F. R. S. Heat shock treatment enhances graft cell survival in skeletal myoblast transplantation to the heart. **Circulation**, v. 102, p. 216-221, 2000.

SWIECH, K. **Identificação de aspectos cinéticos e fisiológicos chave no cultivo de células de *Drosophila melanogaster* S2 para a produção de glicoproteína do vírus da raiva (GPV) recombinante**. Tese (Doutorado em Engenharia Química). Departamento de Engenharia Química, Universidade Federal de São Carlos, São Carlos. 2007.

SWIECH, K.; ROSSI, N.; SILVA, B. G.; JORGE, S. A. C.; ASTRAY, R. M.; SUAZO, C. A. T. Bioreactor culture of recombinant *Drosophila melanogaster* S2 cells: characterization of metabolic features related to cell growth and production of the rabies virus glycoprotein. **Cytotechnology**, v. 57, p. 61-66, 2008a.

SWIECH, K.; SILVA, C. S.; ARANTES, M. K.; SANTOS, A. S.; ASTRAY, R. M.; PEREIRA, C. A.; SUAZO, C. A. T. Characterization of growth and metabolism of

Drosophila melanogaster cells transfected with the rabies-virus glycoprotein gene. **Biotechnology and Applied Biochemistry**, v. 49, p. 41-49, 2008b.

TACIRO, M.K, **Processo contínuo de produção de polihidroxialcanoatos de cadeia média (PHA_{ncl}) sob limitação múltipla de nutrientes**. Tese de doutorado no Programa Interunidades em Biotecnologia. USP/Instituto Butantan/IPT, 2008.

TAVARES, L. Z. **Otimização de processo de produção de etanol por levedura amilolítica recombinante utilizando técnicas de modelagem matemática**. Tese (Doutorado em Biotecnologia). Programa Interunidades em Biotecnologia USP/Instituto Butantan/IPT. 2005.

TORRES, B.B., Elementos de enzimologia. In: SCHMIDELL, W.; LIMA, U. A.; AQUARONE, E.; BORZANI, W. (Org.). **Biotecnologia Industrial**. 1 ed. São Paulo: Edgard Blücher, v. 1, p. 151-176, 2001.

TRAMPER, J.; VLAK, J. M.; GOOIJER, C. D. Scale-up Aspects os Sparged Insect Cells Bioreactors. **Cytotechnology**, v. 20, p. 221, 1996.

URBANCZIK, R.; WAGNER, C. An improves algorithm for stoichiometric network analysis: theory and applications. **Bioinformatics**. v. 21, Issue: 7, p. 1203-1210, 2005.

VALLE, M. A.; KESTER, M. B.; BURNS, A. L.; MARX, S. J.; SPIEGEL, A. M.; SHILOACH, J. Production and purification of human menin from *Drosophila melanogaster* S2 cells using stirred tank reactor. **Cytotechnology**, v.35, p. 127-135, 2001.

VALLINO, J.J.; STEPHANOPOULOS, G, **Metabolic flux distributions in *corynebacterium glutamicum* during growth and lysine over production**. **Biotechnology and Bioengineering**, v. 41, p. 633-646, 1993.

VENTINI, D.C., **Desenvolvimento de bioprocessos com células S2 e CHO em biorreatores para produção de proteínas recombinantes**. Tese de mestrado no Programa Interunidades em Biotecnologia. USP/Instituto Butantan/IPT, 2010.

VIEIRA, P.B., Cultivo de células de *Drosophila melanogaster* (S2) em processo contínuo. . Tese de mestrado, Universidade São Paulo, 2010

VOLESKY, B.; VOTRUBA, J. **Modeling and optimization of fermentation processes**. Amsterdam e New York: Elsevier. 1992.

WALTZ, R. A.; MORALES, J. L.; NOCEDAL, J.; ORBAN, D. An interior algorithm for nonlinear optimization that combines line search and trust region steps. **Mathematical Programming**, v. 107, n. 3, p. 391–408, 2006.

WANG, M-Y.; KWONG, S.; BENTLEY, W.E. Effects of oxygen/glucose/glutamine feeding on insect cell baculovirus protein expression: Study on epoxide hydrolase production. **Biotechnology Progress**, v. 9, p.355-361, 1993.

WILDE, D. J.; BEIGHTLER, C. S. **Foundations of Optimization**. New Jersey: Prentice-Hall, 1967.

YOKOMIZO A.Y.; JORGE S.A.C.; ASTRAY R.M.; FERNANDES I.; RIBEIRO O.G.; HORTON D.S.P.; TONSO A.; TORDO N.; PEREIRA C.A. Rabies virus glycoprotein expression in *Drosophila* S2 cells I. Functional recombinant protein in stable co-transfected cell line. **Biotechnology Journal** v.2, p.102-109, 2007.

ZAMBONI, L. C.; MONEZZI JR, O. **Cálculo Numérico para Universitários**. 1. ed. São Paulo: Páginas e Letras, 343 p., 2002.

ZHANG, Y. **Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment**. Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD, Technical Report TR96-01, July, 1995.

ZHONG, N.; ZHANG, Y.; FANG, Q-Z.; ZHOU, Z-N. Intermittent hypoxia exposure-induced heat-shock protein 70 expression increases resistance of rat heart to ischemic injury. **Acta Pharmacologica Sinica**, v. 5, p. 467-472, 2000.

ZHOU, D.; XUE, J.; LAI, J. C. K.; SCHORK, N. J.; WHITE, K. P.; HADDAD, G. G. Mechanisms Underlying Hypoxia Tolerance in *Drosophila melanogaster*. hairy as a Metabolic Switch. **Plos Genet**, v. 4, Issue 10, p. 1-12, 2008.

ZHU, Y.-S.; CHEN, W.-K. Computer-aided design of communication networks. **World Scientific**, 2000.

Apêndices

APÊNDICE A – MANUAL DO SOFTWARE TrataDados

1. Introdução

O *software* TrataDados foi desenvolvido pelo autor desta tese, em MATLAB versão 7.4 (R2007a), para ser utilizado como ferramenta básica para o tratamento dos dados utilizados na modelagem matemática proposta. Sua versão mais recente utiliza os novos recursos do MATLAB 7.10.0 (R2010a).

Este aplicativo permite:

- efetuar a leitura de dados experimentais diretamente de uma planilha Excel;
- efetuar o ajuste da curva de cada variável de estado utilizando diferentes métodos: *spline*, *smoothing spline* e Hermite;
- determinar graficamente os tempos que são importantes durante o crescimento celular como início da fase exponencial, término da fase exponencial, tempo final de crescimento, tempo de início da morte e outros;
- elaborar o gráfico de $\ln X_v$, calcular o coeficiente de correlação para a reta ajustada e determinar o valor de μ máximo;
- remover pontos experimentais (que foram lidos de forma errada durante o experimento, por exemplo);
- inserir novos pontos experimentais para melhorar o ajuste da curva;
- mover pontos experimentais;
- efetuar um pré-ajuste dos pontos da fase exponencial, movendo automaticamente os mesmos para que o ajuste da reta de $\ln X_v$ seja mais preciso;
- durante a edição da curva referente a uma variável de estado, permitir a visualização do gráfico normalizado de outras variáveis;
- calcular as velocidades de consumo de substratos;
- efetuar o cálculo das conversões, com possibilidade de ajuste de pontos e elaboração de gráficos;
- salvar os dados referentes ao alisamento em uma planilha Excel;
- salvar os gráficos gerados no formato bmp.

2. Instalação do software

Para utilizar o *software* TrataDados é necessário instalar a versão 7.13 da biblioteca MATLAB Compiler Runtime (MCR) de acesso gratuito (não é necessário ter uma cópia completa do MATLAB). Para isto basta executar o arquivo MCRInstaller.exe distribuído junto com o *software* e seguir as instruções apresentadas na tela.

3. Arquivo de dados experimentais

O primeiro passo para a utilização deste *software* é a preparação do arquivo Excel que contém as condições do experimento e os dados experimentais. Isto pode ser feito utilizando-se como base o arquivo modelo (Dados.xls) que acompanha o *software*. Neste arquivo tem-se duas planilhas: “Condições de Ensaio” e “Dados Originais”.

A planilha “Condições de Ensaio”, vista na Figura 96, tem por finalidade registrar resumidamente todas as condições do ensaio. Esta planilha possui texto livre e pode ser modificada aleatoriamente pelo usuário.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2		Esta planilha deve ser utilizada para conter as condições de ensaio, seu texto é livre.																	
3																			
4		Dados do Ensaio:																	
5																			
6		Nome do ensaio:												Responsável:					
7																			
8		Data de início do ensaio:												Tipo de célula:					
9																			
10		Meio:												Temperatura:					
11																			
12		Sistema (Schott? Reator?....)												Vol. total (mL):					
13																			
14		Outros detalhes :												Vol. útil (de meio) (mL):					
15																			
16		Aeração: (Membrana, Borbulhamento, Superfície)												Agitação:					
17																			
18		Tipo de pá:												Controla OD?					
19																			
20		Set Point OD:												Automático?					
21																			
22		Controla pH?								Set Point:				Tipo de Alkali:					
23																			
24		Conc. Alkali:								Tipo de Ácido:				Conc. Ácido:					
25																			
26		Controla Espuma?								Forma (Automático/Manual):									
27																			
28		Conc. Antiespumante								Tipo de Antiespumante									
29																			
30																			
31		Dados do Inóculo:																	
32																			
33		X _{FINAL} (cel/m)												Nº de passagens:					
34																			
35		Tipo de frasco																	

Figura 96 - Planilha Excel com as condições de ensaio.

A planilha “Dados Originais”, vista na Figura 97, contém os dados experimentais que serão lidos e manipulados pelo TrataDados. As células em amarelo são protegidas para que o usuário não modifique dados importantes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	INSTRUÇÕES													
2		1) Inserir comentários apenas nas linhas de 1 a 10. Não inserir células com valores numéricos.												
3		2) Não alterar a ordem das colunas												
4		3) Indicar dados inexistentes com o valor -1 (pontos serão ignorados no cálculo)												
5		4) Não alterar células que possuem fórmulas												
6		5) Colocar os dados apenas nesta planilha												
7		6) Não trocar o nome das planilhas (guias)												
8		7) Não alterar as planilhas geradas automaticamente pelo MATLAB												
9		8) O NOME DO ENSAIO deve estar obrigatoriamente na célula C12												
10		9) O NOME DO PRODUTO deve estar obrigatoriamente na célula G12												
11														
12	NOME DO ENSAIO		XPTO		NOME DO PRODUTO		GPV							
13														
14			Concentração celular (cel/mL)											
15	Amostra	Tempo	X (cel/mL)			Viab	GLC	GLN	GLU	LAC	NH₄	Elisa (GPV)	GPV_E	GPV
16		(d)	X_{VIÁVEL}	X_{MORTA}	X_{TOTAL}	(%)	(g/L)	(mg/L)	(mg/L)	(g/L)	(mg/L)	(ng/10⁶ cel)	(ng/10⁷ cel)	(ng/mL)
17	0	0,00	7,93E+05	6,25E+03	7,99E+05	99,2	9,81	3012,00	-1,00	0,01943	12,50	48,34	483,40	38,31
18	1	1,06	1,36E+06	3,25E+04	1,39E+06	97,7	9,86	2744,00	-1,00	0,01943	23,90	69,33	693,30	93,94
19	2	1,84	2,44E+06	6,00E+04	2,50E+06	97,6	-1,00	-1,00	-1,00	-1,00000	42,14	-1,00	-1,00	-1,00
20	3	2,13	3,19E+06	1,20E+05	3,31E+06	96,4	8,94	2140,00	-1,00	0,01943	40,47	117,34	1173,40	374,31
21	4	2,72	4,91E+06	1,65E+05	5,08E+06	96,8	-1,00	-1,00	-1,00	-1,00000	60,69	-1,00	-1,00	-1,00
22	5	3,10	7,17E+06	2,60E+05	7,43E+06	96,5	7,41	986,00	-1,00	0,08650	53,74	118,03	1180,30	846,28
23	6	3,88	1,09E+07	2,06E+05	1,11E+07	98,1	7,17	690,00	-1,00	0,14870	77,40	91,66	916,60	1000,70
24	7	4,09	1,12E+07	3,19E+05	1,15E+07	97,2	6,77	634,00	-1,00	0,15716	71,37	112,17	1121,70	1251,40

	O	P	Q	R	S	T	U	V	W	X	Y	Z
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14			Concentração celular (cel/mL)									
15	Tempo	X (cel/mL)			Viab	GLC	GLN	GLU	LAC	NH₄	GPV	GPV_E
16	(d)	X_{VIÁVEL}	X_{MORTA}	X_{TOTAL}	(%)	(g/L)	(g/L)	(g/L)	(g/L)	(g/L)	(ug/L)	(ug/10⁷ cel)
17	0,00	7,93E+05	6,25E+03	7,99E+05	99,2	9,81	3,01	-1,00	0,01943	0,01250	38,31	483,40
18	1,06	1,36E+06	3,25E+04	1,39E+06	97,7	9,86	2,74	-1,00	0,01943	0,02390	93,94	693,30
19	1,84	2,44E+06	6,00E+04	2,50E+06	97,6	-1,00	-1,00	-1,00	-1,00000	0,04214	-1,00	-1,00
20	2,13	3,19E+06	1,20E+05	3,31E+06	96,4	8,94	2,14	-1,00	0,01943	0,04047	374,31	1173,40
21	2,72	4,91E+06	1,65E+05	5,08E+06	96,8	-1,00	-1,00	-1,00	-1,00000	0,06069	-1,00	-1,00
22	3,10	7,17E+06	2,60E+05	7,43E+06	96,5	7,41	0,99	-1,00	0,08650	0,05374	846,28	1180,30
23	3,88	1,09E+07	2,06E+05	1,11E+07	98,1	7,17	0,69	-1,00	0,14870	0,07740	1000,70	916,60
24	4,09	1,12E+07	3,19E+05	1,15E+07	97,2	6,77	0,63	-1,00	0,15716	0,07137	1251,40	1121,70

Figura 97 - Planilha de Excel com o dados experimentais.

Esta planilha deve ser manipulada com muita cautela pelo usuário, seguindo as seguintes recomendações:

- inserir comentários apenas nas linhas de 1 a 10. Não inserir células com valores numéricos nestas linhas;
- não alterar a ordem das colunas;

- indicar dados inexistentes com o valor -1 (pontos serão ignorados no cálculo);
- utilizar a unidade adequada para cada valor;
- não alterar células que possuem fórmulas;
- colocar os dados apenas nesta planilha;
- não trocar o nome das planilhas (guias);
- não alterar as planilhas geradas automaticamente pelo MATLAB;
- o NOME DO ENSAIO deve estar obrigatoriamente na célula C12. Este nome será utilizado nos títulos dos gráficos;
- o NOME DO PRODUTO deve estar obrigatoriamente na célula G12.

Os dados experimentais que podem ser tratados por este programa são: Concentração Celular, Viabilidade, Glicose (GLC), Glutamina (GLN), Glutamato (GLU), Lactato (LAC), Amônia (NH₄), Alanina (ALA), GPV e GPV_e.

4. Tratamento de dados

Ao dar um duplo clique sobre o arquivo executável “TrataDados.exe”, o *software* de Tratamento de Dados será acionado. A Janela Principal possui fundo preto, semelhante ao “Prompt de Comando” do Windows, e servirá apenas para exibir ao usuário mensagens de progresso das atividades que estão sendo executadas.

A primeira ação do *software* é abrir uma caixa de diálogo para que o usuário possa escolher o arquivo Excel que contém seus dados experimentais (Figura 98).

Após a leitura das informações do arquivo Excel, cujos detalhes são mostrados na janela principal, será perguntado ao usuário se ele deseja efetuar o ajuste de curvas (Figura 99). O TrataDados permite fazer o ajuste de curvas e também o cálculo das conversões. Caso o ajuste já tenha sido feito, o usuário pode pular esta etapa e ir direto para as conversões.

Figura 98 - Caixa de Diálogo – Selecionar arquivos de dados.

Figura 99 - Caixa de Diálogo – Efetuar Ajuste de Curvas.

O ambiente de trabalho do TrataDados utiliza, além da Janela Principal, duas janelas adicionais: a Janela de Gráficos e o Painel de Controle (Figura 100). A Janela Principal, muitas vezes é sobreposta pelas outras janelas, porém não deve ser fechada, pois desta forma o *software* seria encerrado. Se quiser torná-la visível, utilize os recursos do seu gerenciador de janelas (ALT-TAB, por exemplo).

Figura 100 - Painel de controle e janela de gráficos.

4.1 Definição da fase exponencial e cálculo de μ_{max}

Para iniciar o tratamento de dados devemos inicialmente definir no Painel de Controle os tempos de início e fim da fase exponencial. Estes valores devem ser estimados visualmente e digitados nas caixas de texto correspondentes. A caixa de verificação (quadrado) ao lado de cada tempo indica se o *software* deve desenhar no gráfico uma linha vertical destacando este tempo. É importante salientar que, sempre que as informações do Painel de controle forem alteradas, devemos clicar no botão “Atualizar” para que as modificações sejam exibidas no gráfico.

O Painel de Controle também permite definir se o usuário admite ou não uma velocidade específica de crescimento μ constante durante a fase exponencial. Se μ for admitido constante, o software pode reposicionar automaticamente os pontos da fase exponencial para melhor ajustar o valor de μ , como será explicado mais adiante. Toda vez que esta opção for modificada devemos clicar no botão “Reiniciar” disponível no gráfico de X_v da Janela de Gráficos para que a opção entre em vigor.

Durante o ajuste de pontos da fase exponencial é importante visualizarmos a curva de μ para verificar o seu comportamento. Para isto, basta clicar na caixa de verificação μ_x existente na Janela de Gráficos.

A Figura 101 mostra uma curva onde os tempos de início e fim da fase exponencial foram definidos como 1.1 e 4.1, respectivamente, e o usuário admite μ constante nesta fase. No gráfico, os círculos vermelhos indicam a posição original dos pontos experimentais, os quadrados pretos indicam pontos experimentais que foram movidos para novas posições e a linha contínua indica o ajuste de curva feito pelo método *smoothing spline*.

Figura 101 - Ajuste da fase exponencial.

Importante: ao digitar os valores dos tempos, lembre-se de utilizar ponto decimal ao invés de vírgula.

O TrataDados utiliza um processo muito simples, mas engenhoso, para reposicionar os pontos da fase exponencial a fim de tornar o valor de μ o mais próximo de uma constante na fase exponencial:

- criar um gráfico com os valores do logaritmo de X_v em função do tempo. Devemos observar que este gráfico, teoricamente deveria ser uma reta, mas possui alguma variação devido à característica experimental dos dados;

- efetuar uma regressão linear para ajustar uma reta referente ao logaritmo de X_v , como ilustra a Figura 102;
- uma vez encontrados os coeficientes da reta, recalculer os valores de X_v para todos os pontos da fase exponencial. O coeficiente angular desta reta indica o valor de $\mu_{\max}$.

Após fazer o reposicionamento dos pontos experimentais, o TrataDados informa no Painel de Controle os coeficientes da regressão linear ($\ln X$ e $\mu_{\max}$) e o seu coeficiente de correlação.

Figura 102 - Ajuste de reta para a curva do $\ln$ e X_v .

Se o resultado deste ajuste prévio ainda não for satisfatório, podemos mover manualmente alguns pontos ou utilizar um método diferente para o alisamento da curva para melhorar o ajuste.

4.2 Tratamento de Dados das demais variáveis de estado

Neste *software*, o usuário tem a liberdade de trabalhar como ele desejar. Todos os gráficos são dinâmicos. Na barra inferior da área de trabalho, o gráfico assinalado pela “bolinha” é o gráfico ativo e os gráficos assinalados pelas caixas de verificação (quadrados) são gráficos normalizados que são utilizados como co-adjuvantes. Apenas o gráfico ativo pode ser editado pelo usuário que pode, por exemplo, mover,

inserir e remover pontos. Os gráficos co-adjuvantes servem apenas para o usuário ter uma noção do que acontece com as outras variáveis enquanto edita uma delas. Na Figura 103 podemos ver o gráfico de X_v sendo editado e os gráficos normalizados de GLC, GLN, LAC e μ_x sendo utilizados como co-adjuvantes.

Figura 103 - Gráfico com X_v ativo e gráficos de GLC, GLN e LAC normalizados.

Os gráficos disponíveis no TrataDados são: células viáveis, viabilidade, produto, produto específico, glicose, glutamina, glutamato, lactato, amônia, alanina, velocidade específica de crescimento, logaritmo das células viáveis e proteína/células totais.

A tela de edição de um gráfico oferece os recursos para:

- escolher o método de alisamento (ajuste de curva). Os métodos escolhidos podem ser: *smoothing spline*, *spline* e Hermite. A opção *smoothing spline* permite a escolha do coeficiente p de suavização da curva;
- inserir novos pontos experimentais. Este recurso é utilizado principalmente para a obtenção de uma melhor curva de μ_x . Para inserir um ponto, basta clicar na região do gráfico onde o ponto deve ser inserido e, em seguida, clicar no botão "Inserir". O ponto inserido será identificado por um asterisco verde dentro de um círculo vermelho;

- remover pontos experimentais. Este recurso é utilizado quando se tem um ponto que está muito fora da curva. Para remover um ponto, basta clicar sobre o mesmo e, em seguida, clicar no botão “Remover”. O ponto removido será identificado por um "x" azul;
- desfazer uma ação. O botão “Desfazer” funciona como um *undo*, desfazendo a última operação;
- retornar um ponto movido para sua posição original. Para isto, basta clicar no ponto e, em seguida, clicar no botão “Original”;
- reiniciar o gráfico desconsiderando todas as alterações feitas. Para isto, basta clicar no botão “Reiniciar”;
- encerrar a edição dos gráficos.

Após o encerramento da edição, o usuário tem a opção de gravar os dados alterados na planilha Excel. As informações gravadas são:

- dados alisados (total de 300 pontos);
- dados alisados que serão utilizados para o ajuste de parâmetros (1 ponto a cada 20 pontos alisados, pois 300 pontos é uma quantidade inviável para a execução do ajuste, pois o torna muito lento);
- velocidades específicas (de crescimento e de todas as demais variáveis de estado envolvidas);
- tempos definidos no Painel de Controle;
- dados modificados, removidos, inseridos, etc. Dados removidos aparecem com o valor -2;
- método utilizado para o ajuste de curva (1=*Smoothing spline*, 2=*spline* e 3=Hermite) e o coeficiente de suavização p utilizado pela *smoothing spline*.

Como este é um processo demorado, se o usuário quiser acompanhar o andamento da gravação, basta tornar visível a janela principal onde as mensagens serão exibidas (Figura 104).

Figura 104 - Acompanhando a gravação de dados.

Em seguida, o TrataDados gera uma versão final dos gráficos e permite a gravação dos mesmos em disco no formato *bitmap* (bmp). A Figura 105 ilustra um desses gráficos. Note os pontos inseridos, removidos e modificados durante o ajuste.

Figura 105 - Gráficos de X_v e μ_x .

4.3. Cálculo das Conversões

A próxima etapa do TrataDados é o Cálculo das Conversões. São calculadas todas as conversões possíveis para os dados experimentais. Durante a edição de um gráfico de conversão, o usuário pode escolher para cada ponto experimental três opções de uso:

- usar o ponto considerando a sua posição original. Estes pontos podem ser identificados por um asterisco verde dentro de um círculo vermelho;
- usar o ponto considerando a sua posição modificada na etapa anterior do tratamento de dados. Estes pontos podem ser identificados por um asterisco verde dentro de um quadrado preto;
- não usar o ponto. Os pontos não utilizados não possuem o asterisco verde.

Na Figura 106, por exemplo, podemos observar que o terceiro ponto não está sendo utilizado, o quarto ponto está sendo utilizado em sua posição modificada e o sexto ponto está sendo utilizado em sua posição original.

Figura 106 - Conversão $Y_{LAC/X}$.

Nesta etapa também são calculados os valores das conversões e os respectivos coeficientes de correlação da regressão linear. Após o encerramento das conversões, os dados modificados podem ser gravados na planilha Excel.

Como este também é um processo demorado, se o usuário quiser acompanhar o andamento da gravação, basta tornar visível a janela principal onde as mensagens serão exibidas (Figura 107).

Figura 107 - Acompanhando a gravação de dados de conversão.

Em seguida, são gerados os gráficos finais de conversão que também podem ser salvos no formato *bitmap* (bmp).

Após finalizar todas as etapas do *software*, as janelas dos gráficos ficarão disponíveis para consulta. Para encerrar o *software*, basta fechar todas as janelas de gráficos. Após fechar a última janela, a janela principal será fechada automaticamente.

4.4. Retornando ao TrataDados

A qualquer momento pode-se retornar ao TrataDados. Os dados já gravados serão lidos da planilha Excel e o *software* retoma a mesma situação de quando foi encerrado. Isso pode acontecer quantas vezes o usuário necessitar.

As informações gravadas na planilha Excel, além de serem utilizadas para restaurar as informações em uma nova seção do TrataDados, podem ser utilizados para gerar novos gráficos no Excel ou em outros aplicativos e também podem ser exportados para outros *softwares* como os de ajuste de parâmetros.

4.5. Inserindo automaticamente no Excel os gráficos gerados

Foi desenvolvida uma *macro* em Excel que permite a inserção automática dos gráficos (*bitmaps*) gerados pelo TrataDados. Basta pressionar *ctrl-i* e novas planilhas serão criadas para alojar os gráficos inseridos.

Para que esta *macro* funcione corretamente, habilite a execução de macros no Excel, não renomeie os arquivos *bmp* gerados automaticamente pelo TrataDados e mantenha todos os arquivos na mesma pasta.

APÊNDICE B – MANUAL DO SOFTWARE AjustaPar

1. Introdução

O *software* AjustaPar foi desenvolvido pelo autor desta tese, em MATLAB versão 7.4 (R2007a), para ser utilizado como ferramenta básica para o ajuste de parâmetros utilizados na modelagem matemática de bioprocessos. O ajuste pode ser feito utilizando diversos métodos que serão descritos mais adiante. Sua versão mais recente utiliza os novos recursos do MATLAB 7.10.0 (R2010a).

2. Instalação do software

Para utilizar o *software* AjustaPar é necessário instalar a versão 7.13 da biblioteca MATLAB *Compiler Runtime* (MCR) de acesso gratuito (não é necessário ter uma cópia completa do MATLAB). Para isto basta executar o arquivo "MCRInstaller.exe" distribuído junto com o *software* e seguir as instruções apresentadas na tela.

3. Arquivos de dados e de equações cinéticas

Para utilizar este *software* é necessário criar um arquivo Excel (obrigatoriamente com o nome Dados.xls) com as informações sobre cada um dos ensaios que serão utilizados para o ajuste de parâmetros. Estas informações são:

- os valores alisados de cada variável de estado em cada ensaio (Figura 108);
- os valores experimentais de cada variável de estado em cada ensaio (Figura 109);
- os valores iniciais de cada parâmetro a ser ajustado (Figura 110);
- os limites inferior e superior utilizados no ajuste dos parâmetros (Figura 110);
- a informação de quais ensaios que serão utilizados no ajuste (Figura 110);

- as equações cinéticas que descrevem o comportamento do sistema (Figura 111).

	A	B	C	D	E	F	G
1	Tempo (h)	1E-06.Xv (cel/mL)	GLC (g/L)	GLN (mg/L)	P (ug/mL)	LAC (g/L)	NH4 (mg/L)
2	0,1	1,01E-01	4,50E+00	6,00E+02	1,51E+00	1,51E-01	5,03E+00
3	10,1	1,65E-01	4,34E+00	5,46E+02	2,57E+00	2,24E-01	8,38E+00
4	20,1	2,64E-01	4,09E+00	4,63E+02	4,25E+00	3,36E-01	1,35E+01
5	30,1	4,07E-01	3,73E+00	3,43E+02	6,79E+00	4,99E-01	2,10E+01
6	40,1	6,02E-01	3,24E+00	1,81E+02	1,04E+01	7,21E-01	3,11E+01
7	50,1	8,18E-01	2,70E+00	6,08E-03	1,51E+01	9,67E-01	4,23E+01
8	60,1	8,18E-01	2,70E+00	1,00E-07	1,83E+01	9,70E-01	4,23E+01
9	70,1	8,18E-01	2,70E+00	1,00E-07	2,14E+01	9,72E-01	4,24E+01

Figura 108 - Dados alisados de cada ensaio.

	A	B	C	D	E	F	G
1	Tempo (h)	1E-06.Xv (cel/mL)	GLC (g/L)	GLN (mg/L)	P (ug/mL)	LAC (g/L)	NH4 (mg/L)
2	0,1	1,01E-01	4,50E+00	6,00E+02	1,51E+00	1,51E-01	5,03E+00
3	10,1	1,65E-01	4,34E+00	5,46E+02	2,57E+00	2,24E-01	8,38E+00
4	20,1	2,64E-01	4,09E+00	4,63E+02	4,25E+00	3,36E-01	1,35E+01
5	30,1	4,07E-01	3,73E+00	3,43E+02	6,79E+00	4,99E-01	2,10E+01
6	40,1	6,02E-01	3,24E+00	1,81E+02	1,04E+01	7,21E-01	3,11E+01
7	50,1	8,18E-01	2,70E+00	6,08E-03	1,51E+01	9,67E-01	4,23E+01
8	60,1	8,18E-01	2,70E+00	1,00E-07	1,83E+01	9,70E-01	4,23E+01
9	70,1	8,18E-01	2,70E+00	1,00E-07	2,14E+01	9,72E-01	4,24E+01

Figura 109 - Dados experimentais de cada ensaio.

	A
1	% Pontos inicial e final de cada ensaio - pontosEnsaio = [i1 f1; i2 f2; i3 f3; ...];
2	pontosEnsaio = [1 8];
3	
4	% Quantidade de ensaios - ensaiosAjuste = [e1 e2 e3 ...];
5	ensaiosAjuste = [1];
6	
7	% Valores iniciais dos parâmetros
8	inicial = [0.035 2.55 150 2.9 0.45 0.65 0.0019 0.020 8.57 0.379 1.03 0.302E-03 51 0.49E-03 0.51 6.95]';
9	
10	% Limite inferior dos parâmetros
11	inferior = [0.035 2.55 150 2.9 0.45 0.65 0.0019 0.020 8.57 0.379 1.03 0.302E-03 51 0.49E-03 0.51 6.95]';
12	
13	% Limite superior dos parâmetros
14	superior = [0.085 2.75 170 3.1 0.65 0.85 0.0021 0.022 8.77 0.399 1.23 0.322E-03 53 0.51E-03 0.53 7.15]';
15	

Figura 110 - Informações de ajuste.

A1	f_x	MIX = k(1)*(y(3)/(y(3)+k(2)))*(k(3)/(k(3)+y(6)))*(k(4)/(k(4)+y(5)));
		A
1		MIX = k(1)*(y(3)/(y(3)+k(2)))*(k(3)/(k(3)+y(6)))*(k(4)/(k(4)+y(5)));
2		resp(1) = MIX*y(1);
3		resp(2) = -(1/k(5))*MIX*y(1) - (1/k(6))*k(15)*MIX*y(1);
4		resp(3) = -(1/k(7))*MIX*y(1) - (1/k(8))*k(16)*MIX*y(1);
5		resp(4) = k(9)*MIX*y(1)+k(10)*y(1);
6		resp(5) = k(11)*MIX*y(1)+k(12)*y(1);
7		resp(6) = k(13)*MIX*y(1)+k(14)*y(1);
8		

Figura 111 - Equações cinéticas.

3.1 Informações para o ajuste

Na planilha “Ajuste”, que pode ser vista na Figura 107, a expressão “pontosEnsaio = [1 8]”, indica que estamos utilizando apenas um ensaio e que as informações referentes a este ensaio (dados alisados e experimentais) correspondem às oito primeiras linhas de dados.

Caso o usuário queira utilizar mais de um ensaio no ajuste, basta colocar os dados alisados e experimentais dos novos ensaios em sequência aos dos ensaios anteriores. Por exemplo, se o segundo ensaio tiver 10 pontos, basta colocar estas informações em seguida aos 8 pontos do primeiro ensaio e ajustar a expressão para “pontosEnsaio = [1 8; 9 18]”, indicando que o primeiro ensaio utiliza os pontos de 1 a 8 e o segundo utiliza os pontos de 9 a 18.

Nesta mesma planilha, a expressão “ensaiosAjuste = [1]” indica que apenas o primeiro ensaio será utilizado no ajuste. Por exemplo, se temos 5 ensaios e queremos utilizar apenas o ensaio 2 e o ensaio 4 no ajuste, podemos escrever “ensaiosAjuste = [2 4]”.

Nesta planilha, os nomes das variáveis (pontosEnsaio, ensaiosAjuste, inicial, inferior e superior) e a sintaxe dos comandos não podem ser modificados.

3.2 Equações Cinéticas

Na planilha “Equacoes”, vista na figura 112, são descritas as equações cinéticas do processo, onde os valores de y correspondem aos valores das variáveis de estado

na mesma ordem em que elas aparecem nas planilhas de dados alisados e experimentais, ou seja, $y(1) = X_v$, $y(2) = \text{GLC}$, $y(3) = \text{GLN}$, $y(4) = \text{pO}_2$, $y(5) = \text{LAC}$ e $y(6) = \text{NH}_4$, neste exemplo.

Os valores de k correspondem aos parâmetros que serão ajustados. Neste exemplo:

- $k(1) = \text{MIX}_{\text{max}}$
- $k(2) = K_{\text{gln}}$
- $k(3) = K_{\text{nh4}}$
- $k(4) = K_{\text{lac}}$
- $k(5) = Y_{\text{x/glc}}$
- $k(6) = Y_{\text{lac/glc}}$
- $k(7) = Y_{\text{x/gln}}$
- $k(8) = Y_{\text{nh4/gln}}$
- $k(9) = \text{Alfa}$
- $k(10) = \text{Beta}$
- $k(11) = \text{Alfa}_1$
- $k(12) = \text{Beta}_1$
- $k(13) = \text{Alfa}_2$
- $k(14) = \text{Beta}_2$
- $k(15) = \text{Gama}$
- $k(16) = \text{Gama}_1$

Desta forma, as equações cinéticas inseridas no arquivo de dados correspondem a:

- $\text{MIX} = \text{MIX}_{\text{Max}} * (\text{GLN} / (\text{GLN} + K_{\text{gln}})) * (K_{\text{nh4}} / (K_{\text{nh4}} + \text{NH}_4)) * (K_{\text{lac}} / (K_{\text{lac}} + \text{LAC}));$
- $dX/dt = \text{MIX} * X_v;$
- $d\text{GLC}/dt = -(1 / Y_{\text{x/glc}}) * \text{MIX} * X_v - (1 / Y_{\text{lac/glc}}) * \text{Gama} * \text{MIX} * X_v;$
- $d\text{GLN}/dt = -(1 / Y_{\text{x/gln}}) * \text{MIX} * X_v - (1 / Y_{\text{nh4/gln}}) * \text{Gama}_1 * \text{MIX} * X_v;$
- $d\text{AcMo}/dt = \text{Alfa} * \text{MIX} * X_v + \text{Beta} * X_v;$
- $d\text{LAC}/dt = \text{Alfa}_1 * \text{MIX} * X_v + \text{Beta}_1 * X_v;$
- $d\text{NH}_4/dt = \text{Alfa}_2 * \text{MIX} * X_v + \text{Beta}_2 * X_v;$

É importante salientar que as equações, as variáveis de estado e os parâmetros devem estar na mesma ordem definida no arquivo de dados.

4. Execução do software

Ao dar um duplo clique sobre o arquivo executável “AjustaPar.exe”, o *software* de Ajuste de Parâmetros será acionado. A Janela Principal possui fundo preto, semelhante ao “Prompt de Comando” do Windows, e servirá como interface para o usuário do sistema.

Para iniciar o ajuste de parâmetros, o usuário precisa escolher o método de otimização que deseja usar. As opções disponíveis são:

- minimização com restrição (*fmincon*);
- poliedros flexíveis sem restrição (*fminsearch*);
- poliedros flexíveis com restrição (versão implementada pelo autor desta tese);
- minimização sem restrição (*fminunc*).

Se a opção *fmincon* for escolhida, temos três opções de método de ajuste: *active-set*, *interior-point* e *sqp*.

Se a opção escolhida for poliedros flexíveis com restrição, serão solicitados os coeficientes de Reflexão (α), Expansão (γ), Contração (β) e Redução (δ).

Em seguida, devemos escolher a quantidade máxima de iterações, a quantidade máxima de avaliações da função objetivo e a precisão desejada. Também é necessária a escolha da fórmula utilizada para o cálculo de resíduos na função objetivo e a definição de normalização ou não das variáveis de estado.

O usuário também tem a liberdade de escolher o método utilizado para a resolução do sistema de equações diferenciais ordinárias (Runge-Kutta, Adams-Bashforth-Moulton, NDF/BDF(Gear) e outros)

Todo este processo pode ser visto no Figura 108.

Após estas definições iniciais, o software começa a fazer o ajuste, mostrando informações relativas a cada iteração como mostra o Figura 113.

```

Escolha o método de otimização:
1 = Minimização com restrição (fmincon)
2 = Poliedros Flexíveis sem restrição (fminsearch)
3 = Poliedros Flexíveis com restrição (pamboukian)
4 = Minimização sem restrição (fminunc)
Qual sua opção? (ENTER = 1)

Tipo (1 = active-set, 2 = interior-point, 3 = SQP)? (ENTER = interior-
point)
Quantidade máxima de iterações? (ENTER = 10000)
Quantidade máxima de avaliações da função objetivo? (ENTER = 100000)
Precisão? (ENTER = 1E-6)

Escolha o método para o cálculo dos resíduos:
1 = Fórmula 1
2 = Fórmula 2
3 = Fórmula 3
4 = Fórmula 4
5 = Fórmula 5
6 = Fórmula 6
7 = Fórmula 7
8 = Fórmula 8
Qual sua opção? (ENTER = Fórmula 6)

Valor de epsilon? (ENTER = 1E-6)

Escolha o método para resolução das EDOs:
1 = Runge-Kutta de 4ª e 5ª ordem (ode45)
2 = Runge-Kutta de 2ª e 3ª ordem (ode23)
3 = Adams-Bashforth-Moulton (ode113)
4 = NDF e BDF(Gear) (ode15s)
5 = Rosenbrock modificado de ordem 2 (ode23s)
6 = Regra Trapezoidal (ode23t)
7 = Runge-Kutta TR-BDF2 (ode23tb)
Qual sua opção? (ENTER = ode45)

Escolha da normalização no cálculo do resíduo:
1 = Sem Normalização
2 = Com Normalização
Qual sua opção? (ENTER = Com normalização)

```

Figura 112 - Escolha do método de otimização.

Iter	F-count	f(x)	Feasibility	First-order optimality	Norm of step
0	17	4.079395e+006	0.000e+000	4.126e+009	
1	34	3.979808e+006	0.000e+000	1.189e+009	2.444e-003
2	51	3.972867e+006	0.000e+000	1.357e+009	8.068e-002
3	68	3.968467e+006	0.000e+000	1.571e+009	8.986e-005
4	86	3.848332e+006	0.000e+000	6.921e+008	4.576e-002
5	103	3.847033e+006	0.000e+000	1.035e+010	1.766e-001
6	120	3.753354e+006	0.000e+000	1.013e+010	7.862e-002
7	137	2.727224e+006	0.000e+000	7.775e+010	7.631e-001
8	154	2.703188e+006	0.000e+000	7.848e+010	1.914e-002
9	171	2.329322e+006	0.000e+000	8.227e+010	4.306e-005
10	188	2.304238e+006	0.000e+000	8.313e+010	6.625e-001
...					

Figura 113 - Informações de cada iteração.

O processo de ajuste é muito parecido para todos os métodos: o método de minimização, a cada iteração, estima um conjunto de parâmetros e faz diversas avaliações da função objetivo até que seu valor (resíduo) esteja dentro do critério imposto. A função objetivo, por sua vez, resolve o sistema de equações diferenciais para os parâmetros correntes e calcula os resíduos através da fórmula escolhida pelo usuário.

Ao término do ajuste, são exibidas as informações relativas ao ajuste e os parâmetros encontrados. O Figura 114 mostra em detalhes as informações exibidas.

```
Método:
Minimização com restrição (fmincon - interior-point)
Quantidade máxima de iterações: 10000
Quantidade máxima de avaliações da função objetivo: 100000
Precisão: 1.000000e-006
Cálculo dos resíduos: Fórmula 6
Valor de epsilon: 0.000001
Variáveis de Estado Normalizadas

Resultados:
Quantidade de iterações: 61
Quantidade de avaliações da função objetivo: 1072
Método matemático utilizado: interior-point
Precisão = 1.623090e-008
Resíduo = 0.000849
Tempo = 17 segundos

Parâmetros calculados:
k = [ 5.5536e-002 2.7500e+002 1.5000e+002 2.9000e+000 5.3349e-001 8.3552e-001 ...
      1.9149e-003 2.2000e-002 8.5701e+000 3.8812e-001 1.1289e+000 3.2198e-004 ...
      5.2078e+001 5.0531e-004 5.2979e-001 6.9500e+000 ]';
```

Figura 114 - Resultados do processamento.

A seguir, o programa também exibe os gráficos com os dados experimentais e os dados calculados através do ajuste de parâmetros (Figura 115):

Após finalizar o ajuste, a janela principal e as janelas dos gráficos ficarão disponíveis para consulta. Para encerrar o *software*, basta fechar todas as janelas de gráficos. Após fechar a última janela, a janela principal será fechada automaticamente.

Figura 115 - Gráficos com pontos experimentais e pontos calculados pelo modelo ajustado.

APÊNDICE C – ESTUDO PARA A ESCOLHA DOS MÉTODOS DO AjustaPar

1. Método de minimização do AjustaPar

No AjustaPar estão disponibilizados ao usuário 4 diferentes métodos de otimização com minimização da função objetivo, sendo eles:

- minimização com restrição (*fmincon*)
- poliedros flexíveis sem restrição (*fminsearch*)
- minimização sem restrição (*fminunc*)
- poliedros flexíveis com restrição

Sendo que a função *fmincon* permite a escolha de três algoritmos diferentes: *Active-Set*, *SQP* e *Interior-Point*.

Será verificada a resposta de cada método utilizando as fórmulas para cálculo do resíduo na Função Objetivo 6 (Tabela 1), as mesmas estimativas iniciais e o mesmo limite superior e inferior. Para essa análise foi utilizado o ensaio Ma_07 (Capítulo 4 item 4.2.3) de Aguiar (2010).

1. *fmincon* – *actvide set* - Não funcionou (NAN)

2. *fmincon* – *interior point*

Método:

Minimização com restrição (*fmincon* - *interior-point*)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 6

Valor de epsilon: 0,1000001

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 125

Quantidade de avaliações da função objetivo: 1972

Método matemático utilizado: *interior-point*

Precisão = 1,088245e-008

Residuo = 11,591379

Tempo = 8 segundos

Parâmetros calculados:

k = [5,1966e-002 2,1075e-003 7,9602e-004 5,2547e-002 2,5390e+002 ...
4,1957e+002 3,0653e+002 5,3725e+002 8,2803e+002 1,3728e+002 ...
1,0000e+000 6,4308e-002 1,5734e-003 7,4165e-003];

Figura 116 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando *fmincon – interior point*.

3. fmincon – sqp

- Trava na Iteração 40 se usar fórmula 6
- Funciona só se usar fórmula 1

Método:

Minimização com restrição (fmincon - sqp)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 1

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 43

Quantidade de avaliações da função objetivo: 914

Método matemático utilizado: sequential quadratic programming

Precisão = 4,287634e-007

Residuo = 2,474961

Tempo = 5 segundos

Parâmetros calculados:

k = [1,7803e-001 4,0639e-002 4,0550e-007 9,5411e-002 9,1149e+002 ...

4,8101e+002 2,8726e+002 2,7313e+002 2,8963e+002 5,4992e+002 ...
 1,0000e+000 1,0208e-001 5,9684e-003 7,8558e-003];

Figura 117 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando *fmincon – sqp*.

4. Poliedros Flexíveis sem restrição (fminsearch)

- Parou após atingir limite de iterações (10000) com fórmula 6

Método:

Poliedros Flexíveis sem restrição (fminsearch)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 6

Valor de epsilon: 0,000001

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 10000

Quantidade de avaliações da função objetivo: 13236

Método matemático utilizado: Nelder-Mead simplex direct search

Precisão = 8,572856e-005

Residuo = 11,654465

Tempo = 71 segundos

Parâmetros calculados:

k = [9,9739e-002 6,2516e-004 9,0072e-003 1,4191e-001 3,5656e+002 ...
 4,3807e+002 1,4597e+002 5,1989e+002 1,5004e+000 2,1878e+001 ...
 1,2177e+000 6,5729e-002 1,6417e-003 8,6518e-003];

Figura 118 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando *fminsearch*.

5. Poliedros Flexíveis com restrição (pamboukian)

- Parou após atingir limite de iterações (9000) com fórmula 6

Método:

Poliédros Flexíveis com restrição (pamboukian)

Quantidade máxima de iterações: 9000

Precisão: 1.000000e-008

Passo: 0.300000

Alfa: 1,000000 Gama: 1,500000 Beta: 0,800000 Delta: 0,500000

Cálculo dos resíduos: Fórmula 6

Valor de epsilon: 0.000001

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 9000

Quantidade de avaliações da função objetivo: 31030

Método matemático utilizado: Poliédros Flexíveis com restrição (pamboukian)

Precisão = 7,372440e+003

Residuo = 36,301062

Tempo = 120 segundos

Parâmetros calculados:

k = [1,2010e-001 1,8149e-003 5,8796e-003 2,7927e-002 5,6379e+002 ...
3,3195e+002 5,9968e+001 6,1540e+002 1,2836e+002 1,4903e+001 ...
8,5929e-001 7,3898e-002 2,2615e-003 5,9607e-005];

Figura 119 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando *pamboukian*.

6. Minimização sem restrição (fminunc)

Método:

Minimização sem restrição (fminunc)

Quantidade máxima de iterações: 130

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 6

Valor de epsilon: 0,000001

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 131

Quantidade de avaliações da função objetivo: 2475

Método matemático utilizado: medium-scale: Quasi-Newton line search

Precisão = 6,370943e-006

Resíduo = 36,119735

Tempo = 90 segundos

Parâmetros calculados:

$k = [2,1058e+000 \ 6,3577e-001 \ 8,2502e-004 \ 1,4504e+001 \ 2,8298e+002 \dots$
 $9,7637e+001 \ 9,7222e+001 \ 3,1294e+002 \ 4,3000e+002 \ 2,1000e+002 \dots$
 $7,2917e+000 \ 1,9043e-001 \ 5,9592e-003 \ 1,0435e-002];$

Figura 120 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando *fminunc*.

Tabela 24 – Resultados do estudo com cada método de minimização

Método	Precisão	Tempo de processo	Valor do Resíduo	Nº de Iterações	Fórmula (Tabela 6)	μ_x (h ⁻¹)	
fmincon	ActiveSet	O método não conseguiu processar o cálculo					
	Interiorp	1,0.10 ⁻⁸	8 s	11,59	125	6	5,1.10 ⁻²
	SQP	4,2.10 ⁻⁷	5 s	2,47	43	1	1,7.10 ⁻¹
fminsearch	8,5.10 ⁻⁵	71 s	11,65	10000	6	9,9.10 ⁻²	
pamboukian	7,3.10 ³	120 s	36,30	9000	6	1,2.10 ⁻¹	
fminunc	6,.3.10 ⁻⁶	90 s	36,11	130	6	2,1	

2. Fórmulas para cálculo de resíduo na função objetivo

Estão à disposição do usuário 8 fórmulas de cálculo de resíduo na Função Objetivo, descritas no Capítulo 3 (Tabela 1). Utilizando-se o ensaio Ma_07 (AGUIAR, 2010) e trabalhando-se com a função *fmincon* com o algoritmo *Interior Point*, verificou-se o resultado apresentado através da utilização de cada uma das fórmulas.

Fórmula 1

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 1

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 6

Quantidade de avaliações da função objetivo: 134

Método matemático utilizado: interior-point

Precisão = 1,582438e-004

Resíduo = 9,188641

Tempo = 2 segundos

Parâmetros calculados:

$k = [4,2602e-002 \quad 1,5191e-002 \quad 1,1779e-005 \quad 3,4635e-001 \quad 2,8294e+002 \quad \dots$
 $9,8646e+001 \quad 9,7690e+001 \quad 3,1296e+002 \quad 4,3000e+002 \quad 2,1000e+002 \quad \dots$
 $8,2039e-001 \quad 2,1768e-001 \quad 2,9804e-003 \quad 5,0038e-005]$;

Figura 121 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 1.

Fórmula 2

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 2

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 43

Quantidade de avaliações da função objetivo: 728

Método matemático utilizado: interior-point

Precisão = 4,250104e-007

Residuo = 3,486199

Tempo = 4 segundos

Parâmetros calculados:

k = [1,4400e-001 3,1910e-002 6,6669e-006 1,3462e-001 3,7019e+002 ...
 5,0000e+002 5,0000e+002 8,2100e+002 4,8591e+002 2,6039e+002 ...
 1,0000e+000 1,2103e-001 8,4678e-004 3,2254e-002];

Figura 122 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 2.

Fórmula 3

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 3

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 6663

Quantidade de avaliações da função objetivo: 100000

Método matemático utilizado: interior-point

Precisão = 3,929551e-005

Residuo = 125,638424

Tempo = 364 segundos

Parâmetros calculados:

k = [4,3886e-001 6,6107e-005 9,7472e-002 9,9689e-001 2,8299e+002 ...
 1,1614e+002 1,0602e+002 3,1299e+002 4,3000e+002 2,1013e+002 ...
 9,9983e-001 1,9424e-001 5,4163e-003 1,6135e-005];

Figura 123 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 3.

Fórmula 4

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 4

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 166

Quantidade de avaliações da função objetivo: 2605

Método matemático utilizado: interior-point

Precisão = 2,270932e-009

Residuo = 9,686932

Tempo = 11 segundos

Parâmetros calculados:

k = [5,9992e-002 4,8572e-003 1,9105e-003 9,9949e-001 2,5581e+002 ...
 3,0779e+002 5,0000e+002 2,6990e+002 1,0000e+003 2,5720e+002 ...
 1,0000e+000 4,7917e-002 1,7757e-003 5,1391e-003];

Figura 124 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 4.

Fórmula 5

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 5

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 140

Quantidade de avaliações da função objetivo: 2209

Método matemático utilizado: interior-point

Precisão = 7,486634e-011

Residuo = 11,591379

Tempo = 9 segundos

Parâmetros calculados:

k = [5,1966e-002 2,1079e-003 7,9584e-004 5,2546e-002 2,3765e+002 ...
4,2376e+002 2,8351e+002 6,2798e+002 9,9561e+002 8,7568e+001 ...
1,0000e+000 6,4308e-002 1,5733e-003 7,4166e-003];

Figura 125 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 5.

Fórmula 6

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 6

Valor de epsilon: 0,000001

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 140

Quantidade de avaliações da função objetivo: 2209

Método matemático utilizado: interior-point

Precisão = 7,486634e-011

Residuo = 11,591379

Tempo = 9 segundos

Parâmetros calculados:

k = [5,1966e-002 2,1079e-003 7,9584e-004 5,2546e-002 2,3765e+002 ...
 4,2376e+002 2,8351e+002 6,2798e+002 9,9561e+002 8,7568e+001 ...
 1,0000e+000 6,4308e-002 1,5733e-003 7,4166e-003];

Figura 126 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 6.

Fórmula 7

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 7

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 118

Quantidade de avaliações da função objetivo: 1965

Método matemático utilizado: interior-point

Precisão = 3,237713e-002

Resíduo = 0,715316

Tempo = 19 segundos

Parâmetros calculados:

k = [2,5681e-001 6,1480e-002 1,3000e-010 1,7484e-001 3,4776e+002 ...
5,0000e+002 5,0000e+002 4,3457e+002 3,8607e+002 9,3890e+002 ...
1,0000e+000 9,7413e-002 1,5193e-003 9,4749e-003];

Figura 127 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 7.

Fórmula 8

Método:

Minimização com restrição (fmincon - interior-point)

Quantidade máxima de iterações: 10000

Quantidade máxima de avaliações da função objetivo: 100000

Precisão: 1,000000e-006

Cálculo dos resíduos: Fórmula 8

Variáveis de Estado Normalizadas

Resultados:

Quantidade de iterações: 87

Quantidade de avaliações da função objetivo: 1386

Método matemático utilizado: interior-point

Precisão = 2,458480e-008

Resíduo = 0,000017

Tempo = 7 segundos

Parâmetros calculados:

k = [1,1688e-001 1,0269e-002 4,4691e-003 5,3895e-001 4,7936e+002 ...
5,0000e+002 5,0000e+002 3,4575e+002 4,3103e+002 2,6940e+002 ...
1,0000e+000 1,0031e-001 4,5247e-008 8,2711e-003];

Figura 128 - Gráfico gerado pelo AjustaPar com os pontos dos dados experimentais (*) e com os dados resultantes da modelagem (linha), utilizando a Fórmula 8.

Tabela 25 – Resultados do estudo com cada fórmula do cálculo do resíduo da Tabela 1

Fórmula	Precisão	Tempo de processo	Valor do Resíduo	Nº de Iterações	μ_x (h ⁻¹)
1	$1,5 \cdot 10^{-4}$	2 s	9,18	6	$4,2 \cdot 10^{-2}$
2	$4,2 \cdot 10^{-7}$	4 s	3,48	43	$1,4 \cdot 10^{-1}$
3	$3,9 \cdot 10^{-5}$	364 s	125,63	6663	$4,3 \cdot 10^{-1}$
4	$2,2 \cdot 10^{-9}$	11 s	9,68	166	$5,9 \cdot 10^{-2}$
5	$7,4 \cdot 10^{-11}$	9 s	11,59	140	$5,1 \cdot 10^{-2}$
6	$7,4 \cdot 10^{-11}$	9 s	11,59	140	$5,1 \cdot 10^{-2}$
7	$3,2 \cdot 10^{-2}$	19 s	0,71	118	$2,5 \cdot 10^{-1}$
8	$2,4 \cdot 10^{-8}$	7 s	0,000017	87	$1,1 \cdot 10^{-1}$

APÊNDICE D – MANUAL DO SOFTWARE DE SIMULAÇÃO

1. Introdução

O *software* de Simulação foi desenvolvido pelo autor desta tese, em MATLAB versão 7.4 (R2007a), para ser utilizado como ferramenta para a visualização do modelo ajustado nesta tese aplicado a diversos ensaios. Sua versão mais recente utiliza os novos recursos do MATLAB 7.10.0 (R2010a). Para realizar a simulação, o usuário precisa apenas fornecer as condições iniciais do ensaio e, se desejar, os dados experimentais para comparação.

2. Instalação do software

Para utilizar o *software* de Simulação é necessário instalar a versão 7.13 da biblioteca MATLAB Compiler Runtime (MCR) de acesso gratuito (não é necessário ter uma cópia completa do MATLAB). Para isto basta executar o arquivo "MCRInstaller.exe" distribuído junto com o software e seguir as instruções apresentadas na tela.

3. Arquivos de dados dos ensaios

Para utilizar este *software* é necessário criar um arquivo Excel com as informações sobre o ensaio a ser simulado. Estas informações são:

- nome do ensaio;
- duração do ensaio (tempo inicial e tempo final);
- concentração inicial de X_v , GLC, GLN, LAC, NH_4 , ALA, GLU e GPV_e ;
- oxigênio dissolvido (%);
- meio de cultivo (TC100AG, SF900 II ou IPL-41 sup);
- construção (Ac ou Mt);

- dados experimentais (opcional).

Os dados iniciais do ensaio devem ser colocados na guia “Valores” da planilha Excel e são essenciais para o desenvolvimento da simulação (Figura 129).

	A	B	C	D	E	F	G	H
1	Ma-07							
2								
3	Tempo inicial (h)	22,742						
4	Tempo final (h)	350						
5	X inicial (cel/L)	821829638						
6	GLC inicial (mM)	53,189						
7	GLN inicial (mM)	26,75						
8	LAC inicial (mM)	2,5						
9	NH4 inicial (mM)	0,715						
10	ALA inicial (mM)	5,117						
11	GLU inicial (mM)	5,012						
12	GPVe inicial (ng/cel)	0,00012						
13	DO(%)	5						
14	Meio	1	Opções: 1 = TC100AG, 2 = SF900, 3 = IPL-41 sup					
15	Construção	1	Opções: 1 = Ac, 2 = Mt					

Figura 129 - Dados iniciais do ensaio.

Os dados experimentais são opcionais e podem ser colocados na guia “Dados” da planilha (Figura 130). Valores experimentais inexistentes devem ser indicados com o valor -1.

	A	B	C	D	E	F	G	H	I
1	tempo (h)	X (cel/L)	GLC (mM)	GLN (mM)	LAC (mM)	NH4 (mM)	ALA (mM)	GLU (mM)	GPVe (ng/cel)
2	0	651250000	54,30606316	30,991318	-1	0,387185843	3,4754784	6,0736832	0,000133632
3	21,33333333	822500000	53,20976676	-1	-1	1,212393264	-1	-1	0,000213628
4	28,33333333	946250000	52,7597829	-1	-1	0,02231564	-1	-1	0,000172015
5	45,75	1425000000	54,53622758	22,6841175	-1	1,898419766	7,1824736	5,9958944	0,000176345
6	52,33333333	1727500000	42,70320143	18,2103825	-1	1,898419766	7,3949056	5,3001168	0,00018501
7	69,33333333	2512500000	45,53912563	-1	-1	2,525359775	-1	-1	0,000152547
8	76,5	3450000000	45,22949914	-1	2,678994211	2,234652347	-1	-1	0,000139325
9	93,5	5565000000	44,64150698	-1	2,980571177	2,630435692	-1	-1	0,000109711
10	100,83333333	6660000000	43,59722909	-1	-1	2,630435692	-1	-1	0,000137332
11	118,33333333	10662500000	30,880611	10,3997385	3,25415229	2,630435692	17,9248304	5,1375536	0,000103868
12	126,08333333	11265625000	31,61467513	6,186098	3,84763198	2,853885528	12,0361472	3,558512	0,000094014
13	142	13343750000	26,4583034	-1	3,36119775	3,225149417	-1	-1	0,000050481
14	148,1666667	12928125000	27,78666155	-1	3,99656313	3,644711275	-1	-1	4,44895E-05
15	165	14656250000	21,61977973	-1	-1	3,359342388	-1	-1	4,07555E-05
16	173,9166667	15312500000	17,34396176	-1	-1	3,359342388	-1	-1	0,000050714
17	189,25	18250000000	13,80179006	4,3232405	5,338499564	2,972630841	32,5002784	4,2010096	0,000042946
18	197	19175000000	10,33376112	-1	-1	3,096316945	-1	-1	0,000040466
19	213,08333333	17690625000	5,062442512	2,7034895	6,10806129	3,225149417	24,7144752	3,0938512	0,00003402
20	221,1666667	19687500000	2,793030364	-1	6,108439612	2,525359775	-1	-1	0,000043596
21	237,33333333	21825000000	0,757374014	-1	-1	2,972630841	-1	-1	0,00002576
22	244,5	17859375000	0,757374014	-1	-1	3,644711275	-1	-1	0,000036399
23	261,75	16637500000	0,757374014	4,480996	6,400261267	8,440223924	37,0213312	4,168568	0,00003323
24	313,5	18525000000	0,757374014	-1	3,428376992	17,64991087	-1	-1	0,000040466
25	340	8831250000	0,757374014	-1	1,151660054	18,34869036	-1	-1	-1

Figura 130 - Dados experimentais do ensaio (opcional).

Se dados experimentais forem fornecidos o *software* mostrará um gráfico comparativo entre os valores simulados e os valores experimentais, caso contrário, exibirá apenas o gráfico dos valores simulados.

4. Execução do *software*

Ao dar um duplo clique sobre o arquivo executável “SimulaS2.exe”, o *software* de Simulação será acionado. A Janela Principal possui fundo preto, semelhante ao “Prompt de Comando” do Windows, e servirá como interface para o usuário do sistema.

Para iniciar a simulação, o usuário precisa escolher o arquivo Excel que possui as informações do ensaio a ser simulado (Figura 131).

Figura 131 - Arquivo Excel com as informações do ensaio.

Em seguida, o usuário deve indicar se a simulação deverá comparar os resultados calculados com valores experimentais (Figura 132).

Figura 132 - Comparação com dados experimentais.

Após a simulação ser executada, as informações (μ_x , K_p , resíduos, etc.) serão exibidas na Janela Principal do software (Figura 133).

Figura 133 - Resultados da simulação.

A seguir, o programa também exibe os gráficos com os dados experimentais (se os mesmos foram fornecidos) e os dados calculados através da simulação (Figura 134).

Após finalizar a simulação, a janela principal e a janelas dos gráficos ficarão disponíveis para consulta. Para encerrar o software, basta fechar a janela de gráficos e a janela principal será fechada automaticamente.

Figura 134 - Gráficos com pontos experimentais e calculados pelo simulador.

APÊNDICE E – LISTAGEM DO SOFTWARE TrataDados

1. Arquivos

Este software é composto pelos seguintes arquivos:

- Ajusta_Curva.m
- Conversao.m
- Dados.xls
- Estah_Vazio.m
- Graficos.m
- GraficosFinais.m
- GraficosFinaisY.m
- GravaDados.m
- GravaDadosY.m
- LeDados.m
- Painel.m
- PontosExp.m
- Remove_Pontos.m
- SalvaGraficos.m
- SalvaGraficosY.m
- TrataDados
- TrataDados.m
- VariaveisGlobais.m
- winontop.c
- winontop.m
- winontop.mexw32

A listagem de cada um desses arquivos é mostrada a seguir.

```

end
for i = 1:length(Xajustado)
    Yfinal(i+ppi-1) = Yajustado(i);
end
end
end

```

3. Conversao.m

```

% *****
% Módulo: Conversao.m
% Descrição: calcula as conversões (Y)
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Conversões Y
function Conversao
% Variáveis Globais
VariaveisGlobais
% Pega tamanho da tela
screensize = get(0,'ScreenSize');
% Define todas as janelas de figuras como "docked"
set(0,'DefaultFigureWindowStyle','docked')
% Define modo inicial e valores de retorno
modo = 1;
% Cria figura centralizada na tela com tamanho 800x600
hFigY = figure('WindowButtonDownFcn', @ButtonDownFcn, ...
    'name','Conversões','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Tag','Enabled',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixo (gráfico)
hEixo1 = axes('Parent',hFigY, ...
    'Position',[.08 .24 .80 .60],...
    'Color','none',... % Transparente (nenhuma cor de fundo)
    'Tag','Eixo1');

hold on;
% Cria controles
lblY = uicontrol(hFigY,'Style','text',...
    'String','Y=0',...
    'BackgroundColor','w', ...
    'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblY');
lblR = uicontrol(hFigY,'Style','text',...
    'String','R=0',...
    'BackgroundColor','w', ...
    'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblR');
lblEstado = uicontrol(hFigY,'Style','text',...
    'String','Selecione um ponto!',...
    'BackgroundColor','w', ...
    'ForegroundColor','r',...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblEstado');
btnOriginal = uicontrol(hFigY,'Callback', @btnOriginalClick, ...
    'Style','pushbutton',...
    'String','Usar Original',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnOriginal');
btnModificado = uicontrol(hFigY,'Callback', @btnModificadoClick, ...
    'Style','pushbutton',...
    'String','Usar Modificado',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnModificado');
btnRemover = uicontrol(hFigY,'Callback', @btnRemoverClick, ...
    'Style','pushbutton',...
    'String','Não Usar',...
    'FontSize', 8, 'FontName','Arial',...

```

```

end
for i = 1:length(Xajustado)
    Yfinal(i+ppi-1) = Yajustado(i);
end
end
end

```

3. Conversao.m

```

% *****
% Módulo: Conversao.m
% Descrição: calcula as conversões (Y)
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Conversões Y
function Conversao
% Variáveis Globais
VariaveisGlobais
% Pega tamanho da tela
screensize = get(0,'ScreenSize');
% Define todas as janelas de figuras como "docked"
set(0,'DefaultFigureWindowStyle','docked')
% Define modo inicial e valores de retorno
modo = 1;
% Cria figura centralizada na tela com tamanho 800x600
hFigY = figure('WindowButtonDownFcn', @ButtonDownFcn, ...
    'name','Conversões','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Tag','Enabled',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixo (gráfico)
hEixo1 = axes('Parent',hFigY, ...
    'Position',[.08 .24 .80 .60],...
    'Color','none',... % Transparente (nenhuma cor de fundo)
    'Tag','Eixo1');

hold on;
% Cria controles
lblY = uicontrol(hFigY,'Style','text',...
    'String','Y=0',...
    'BackgroundColor','w', ...
    'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblY');
lblR = uicontrol(hFigY,'Style','text',...
    'String','R=0',...
    'BackgroundColor','w', ...
    'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblR');
lblEstado = uicontrol(hFigY,'Style','text',...
    'String','Selecione um ponto!',...
    'BackgroundColor','w', ...
    'ForegroundColor','r',...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag','lblEstado');
btnOriginal = uicontrol(hFigY,'Callback', @btnOriginalClick, ...
    'Style','pushbutton',...
    'String','Usar Original',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnOriginal');
btnModificado = uicontrol(hFigY,'Callback', @btnModificadoClick, ...
    'Style','pushbutton',...
    'String','Usar Modificado',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnModificado');
btnRemover = uicontrol(hFigY,'Callback', @btnRemoverClick, ...
    'Style','pushbutton',...
    'String','Não Usar',...
    'FontSize', 8, 'FontName','Arial',...

```

```

        'Tag','btnRemover');
btnDesfazer = uicontrol(hFigY,'Callback', @btnDesfazerClick, ...
    'Style','pushbutton',...
    'String', 'Desfazer',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnDesfazer');
btnReiniciar = uicontrol(hFigY,'Callback', @btnReiniciarClick, ...
    'Style','pushbutton',...
    'String', 'Reiniciar',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnReiniciar');
btnEncerrar = uicontrol(hFigY,'Callback', @btnEncerrarClick,...
    'Style','pushbutton',...
    'String', 'Encerrar',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnEncerrar');
% Cria radiogroup para selecao de variável
radY = uibuttongroup('SelectionChangeFcn',@radYCallback,...
    'Units','points','pos', [ 0 0 1 1 ], ... % qualquer coordenada
    'BackgroundColor','w', 'BorderType','none', ...
    'Tag','radY');
ph = 10; % Posição horizontal
if (~Estah_Vazio(oGLC))
    radY1 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'Yx/GLC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY1');

    ph = ph + 80;
end
if (~Estah_Vazio(oGLN))
    radY2 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'Yx/GLN',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY2');

    ph = ph + 80;
end
if (~Estah_Vazio(oLAC))
    radY3 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'YLAC/x',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY3');

    ph = ph + 80;
end
if (~Estah_Vazio(oLAC) && ~Estah_Vazio(oGLC))
    radY4 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'YLAC/GLC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY4');

    ph = ph + 80;
end
if (~Estah_Vazio(oNH4))
    radY5 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'YNH4/x',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY5');

    ph = ph + 80;
end
if (~Estah_Vazio(oNH4) && ~Estah_Vazio(oGLN))
    radY6 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', 'String', 'YNH4/GLN',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY6');

    ph = ph + 80;
end
if (~Estah_Vazio(oProd) && ~Estah_Vazio(oGLC))
    radY7 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', ...
        'String', [ 'Y' NomeProduto '/GLC' ],...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY7');

    ph = ph + 80;
end
if (~Estah_Vazio(oALA) && ~Estah_Vazio(oGLC))
    radY8 = uicontrol('parent',radY, 'pos',[ph 10 100 13],...
        'Style','radiobutton', ...
        'String', [ 'YALA/GLC' ],...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radY8');

    ph = ph + 80;
end

```

```

elseif (op==6)
    % YNH4/GLN
    Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = 'NH4 (g/L)';
    origVX = oGLN; origVY = oNH4;
    modVX = copiados2(mGLN); modVY = copiados2(mNH4);
    ptUsado = uYNH4_GLN;
elseif (op==7)
    % YProd/GLC
    Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ NomeProduto ' (µg/L)' ];
    origVX = oGLC; origVY = oProd;
    modVX = copiados2(mGLC); modVY = copiados2(mProd);
    ptUsado = uYProd_GLC;
elseif (op==8)
    % YALA/GLC
    Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ 'ALA (g/L)' ];
    origVX = oGLC; origVY = oALA;
    modVX = copiados2(mGLC); modVY = copiados2(mALA);
    ptUsado = uYALA_GLC;
elseif (op==9)
    % YALA/GLN
    Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = [ 'ALA (g/L)' ];
    origVX = oGLN; origVY = oALA;
    modVX = copiados2(mGLN); modVY = copiados2(mALA);
    ptUsado = uYALA_GLN;
end
resp = 0;
R_resp = 0;
% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = origVX(1:ult_ponto);
y = origVY(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y); % só deixa pontos que existam nas duas variáveis
% Dados finais
xf = modVX(1:ult_ponto);
yf = modVY(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf); % só deixa pontos que existam nas duas variáveis
[ xf yf ] = AjustaModificados2(xf, yf); % troca -2 por realmin
% Inicialmente são usados os pontos originais
if (isempty(ptUsado) || length(ptUsado)~=length(x))
    ptUsado = zeros(length(x),1); % 0 = usar original, 1 = usar modificado, 2 = não usar
end
% Inicializa undo
pt = 1;
end

function AjusteFinal
VariaveisGlobais
if (op==1)
    uYx_GLC = ptUsado; Yx_GLC = resp; R_Yx_GLC = R_resp;
elseif (op==2)
    uYx_GLN = ptUsado; Yx_GLN = resp; R_Yx_GLN = R_resp;
elseif (op==3)
    uYLAC_x = ptUsado; YLAC_x = resp; R_YLAC_x = R_resp;
elseif (op==4)
    uYLAC_GLC = ptUsado; YLAC_GLC = resp; R_YLAC_GLC = R_resp;
elseif (op==5)
    uYNH4_x = ptUsado; YNH4_x = resp; R_YNH4_x = R_resp;
elseif (op==6)
    uYNH4_GLN = ptUsado; YNH4_GLN = resp; R_YNH4_GLN = R_resp;
elseif (op==7)
    uYProd_GLC = ptUsado; YProd_GLC = resp; R_YProd_GLC = R_resp;
elseif (op==8)
    uYALA_GLC = ptUsado; YALA_GLC = resp; R_YALA_GLC = R_resp;
elseif (op==9)
    uYALA_GLN = ptUsado; YALA_GLN = resp; R_YALA_GLN = R_resp;
end
end

function radYCallback(source,eventdata)
VariaveisGlobais
AjusteFinal;
if (get(source,'SelectedObject')==radY1)
    op = 1;
elseif (get(source,'SelectedObject')==radY2)
    op = 2;
elseif (get(source,'SelectedObject')==radY3)
    op = 3;
elseif (get(source,'SelectedObject')==radY4)
    op = 4;
elseif (get(source,'SelectedObject')==radY5)
    op = 5;
elseif (get(source,'SelectedObject')==radY6)
    op = 6;
elseif (get(source,'SelectedObject')==radY7)
    op = 7;
elseif (get(source,'SelectedObject')==radY8)
    op = 8;
elseif (get(source,'SelectedObject')==radY9)

```

```

elseif (op==6)
    % YNH4/GLN
    Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = 'NH4 (g/L)';
    origVX = oGLN; origVY = oNH4;
    modVX = copiados2(mGLN); modVY = copiados2(mNH4);
    ptUsado = uYNH4_GLN;
elseif (op==7)
    % YProd/GLC
    Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ NomeProduto ' (µg/L)' ];
    origVX = oGLC; origVY = oProd;
    modVX = copiados2(mGLC); modVY = copiados2(mProd);
    ptUsado = uYProd_GLC;
elseif (op==8)
    % YALA/GLC
    Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ 'ALA (g/L)' ];
    origVX = oGLC; origVY = oALA;
    modVX = copiados2(mGLC); modVY = copiados2(mALA);
    ptUsado = uYALA_GLC;
elseif (op==9)
    % YALA/GLN
    Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = [ 'ALA (g/L)' ];
    origVX = oGLN; origVY = oALA;
    modVX = copiados2(mGLN); modVY = copiados2(mALA);
    ptUsado = uYALA_GLN;
end
resp = 0;
R_resp = 0;
% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = origVX(1:ult_ponto);
y = origVY(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y); % só deixa pontos que existam nas duas variáveis
% Dados finais
xf = modVX(1:ult_ponto);
yf = modVY(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf); % só deixa pontos que existam nas duas variáveis
[ xf yf ] = AjustaModificados2(xf, yf); % troca -2 por realmin
% Inicialmente são usados os pontos originais
if (isempty(ptUsado) || length(ptUsado)~=length(x))
    ptUsado = zeros(length(x),1); % 0 = usar original, 1 = usar modificado, 2 = não usar
end
% Inicializa undo
pt = 1;
end

function AjusteFinal
VariaveisGlobais
if (op==1)
    uYx_GLC = ptUsado; Yx_GLC = resp; R_Yx_GLC = R_resp;
elseif (op==2)
    uYx_GLN = ptUsado; Yx_GLN = resp; R_Yx_GLN = R_resp;
elseif (op==3)
    uYLAC_x = ptUsado; YLAC_x = resp; R_YLAC_x = R_resp;
elseif (op==4)
    uYLAC_GLC = ptUsado; YLAC_GLC = resp; R_YLAC_GLC = R_resp;
elseif (op==5)
    uYNH4_x = ptUsado; YNH4_x = resp; R_YNH4_x = R_resp;
elseif (op==6)
    uYNH4_GLN = ptUsado; YNH4_GLN = resp; R_YNH4_GLN = R_resp;
elseif (op==7)
    uYProd_GLC = ptUsado; YProd_GLC = resp; R_YProd_GLC = R_resp;
elseif (op==8)
    uYALA_GLC = ptUsado; YALA_GLC = resp; R_YALA_GLC = R_resp;
elseif (op==9)
    uYALA_GLN = ptUsado; YALA_GLN = resp; R_YALA_GLN = R_resp;
end
end

function radYCallback(source,eventdata)
VariaveisGlobais
AjusteFinal;
if (get(source,'SelectedObject')==radY1)
    op = 1;
elseif (get(source,'SelectedObject')==radY2)
    op = 2;
elseif (get(source,'SelectedObject')==radY3)
    op = 3;
elseif (get(source,'SelectedObject')==radY4)
    op = 4;
elseif (get(source,'SelectedObject')==radY5)
    op = 5;
elseif (get(source,'SelectedObject')==radY6)
    op = 6;
elseif (get(source,'SelectedObject')==radY7)
    op = 7;
elseif (get(source,'SelectedObject')==radY8)
    op = 8;
elseif (get(source,'SelectedObject')==radY9)

```



```

        op = 9;
    end
    AjusteInicial;
    Desenha;
end

function btnOriginalClick(hObject,eventdata,handles)
% Variáveis Globais
VariaveisGlobais
% Verifica ponto mais próximo do click
pt = 1;
for i=2:length(x)
    if (abs(xc-x(i)) < abs(xc-x(pt)))
        pt = i;
    end
end
y_undo = ptUsado(pt);
ptUsado(pt) = 0;
% Desenha curvas
Desenha;
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Desfazer",...
    "Reiniciar" ou "Encerrar"!');
end

function btnModificadoClick(hObject,eventdata,handles)
% Variáveis Globais
VariaveisGlobais
% Verifica ponto mais próximo do click
pt = 1;
for i=2:length(x)
    if (abs(xc-x(i)) < abs(xc-x(pt)))
        pt = i;
    end
end
y_undo = ptUsado(pt);
if (xf(pt) == realmin || yf(pt) == realmin) % ponto foi excluído
    ptUsado(pt) = 2; % não usar
else
    ptUsado(pt) = 1; % usar modificado
end
% Desenha curvas
Desenha;
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Desfazer",...
    "Reiniciar" ou "Encerrar"!');
end

function btnRemoverClick(hObject,eventdata,handles)
% Variáveis Globais
VariaveisGlobais
% Verifica se ainda pode remover pontos (devem restar pelo menos 2)
cont = 0;
for i=1:length(ptUsado) %#ok<NODEF>
    if (ptUsado(i)~=2) % ponto i é usado
        cont = cont + 1;
    end
end
if (cont==2)
    screensize = get(0,'ScreenSize');
    hDialog = warndlg('Os dois últimos pontos não podem ser removidos!','Atenção');
    set(hDialog,'Position',[(screensize(3)-240)./2 (screensize(4)-80)./2 240 80]);
    return;
end
% Verifica ponto mais próximo do click
pt = 1;
for i=2:length(x)
    if (abs(xc-x(i)) < abs(xc-x(pt)))
        pt = i;
    end
end
% Elimina ponto
y_undo = ptUsado(pt);
ptUsado(pt)=2;
% Desenha curvas
Desenha;
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Desfazer",...
    "Reiniciar" ou "Encerrar"!');
end

function btnDesfazerClick(hObject,eventdata,handles)
% Variáveis Globais
VariaveisGlobais
% Desfazer
ptUsado(pt) = y_undo;
% Desenha curvas
Desenha;
set(findobj('Tag','lblEstado'),'String',...
    'Selecione um ponto ou clique em "Reiniciar" ou "Encerrar"!');
end

```

```

function btnReiniciarClick(hObject,eventdata,handles)
    VariaveisGlobais
    AjusteInicial;
    % Desenha curvas
    Desenha;
    set(findobj('Tag','lblEstado'),'String',...
        'Selecione um ponto ou clique em "Desfazer" ou "Encerrar"!');
end

function btnEncerrarClick(hObject,eventdata,handles)
    VariaveisGlobais
    % Ajusta modo
    modo = 2;
    % Ajuste Final
    AjusteFinal;
    % Fecha gráficos
    close(hFigY);
    % Sair da pausa
    uiresume;
end

function ButtonDownFcn(hObject, eventdata)
    % Variáveis Globais
    VariaveisGlobais
    % Verifica se edição foi encerrada
    if (modo==2)
        return;
    end
    % Pega posição do último click
    posClick = get(hEixo1,'CurrentPoint');
    xc = posClick(1,1); % x do click
    yc = posClick(1,2); % y do click
    % Não aceita clicks fora do Eixo
    LimiteX = get(hEixo1,'XLim');
    LimiteY = get(hEixo1,'YLim');
    if (xc<LimiteX(1) || yc<LimiteY(1) || xc>LimiteX(2) || yc>LimiteY(2))
        return;
    end
    set(findobj('Tag','lblEstado'),'String', 'Clique em "Usar Original",...
        "Usar Modificado" ou "Não Usar"!');
end

function Desenha
    % Variáveis Globais
    VariaveisGlobais
    % Dados para interpolação
    xinter = [];
    yinter = [];
    tam = length(x); % Quantidade de elementos do vetor x
    if tam > 0
        primeiro = min([x;xf]); % Menor valor de x
        ultimo = max([x;xf]); % Maior valor de x
        passo_interp = (ultimo-primeiro)/200; % Passo para interpolação ajusta_curva
        % Verifica pontos que serão utilizados
        cont = 1;
        for i=1:tam
            if (ptUsado(i) == 0) % usar ponto original
                xinter(cont) = x(i); %ok<AGROW>
                yinter(cont) = y(i); %ok<AGROW>
                cont = cont + 1;
            end
            if (ptUsado(i) == 1 && yf(i) ~= realmin) % usar ponto modificado não excluído
                xinter(cont) = xf(i); %ok<AGROW>
                yinter(cont) = yf(i); %ok<AGROW>
                cont = cont + 1;
            end
        end
    end
    % Verifica se existem pontos suficientes para ajustar reta (pelo menos dois)
    if length(xinter) < 2
        resp = 0;
        R_resp = 0;
        if (op == 1)
            set(lblY,'String',['Yx/GLC = ' num2str(resp,'%10.4e') ' (cel/g)']);
        elseif (op == 2)
            set(lblY,'String',['Yx/GLN = ' num2str(resp,'%10.4e') ' (cel/g)']);
        elseif (op == 3)
            set(lblY,'String',['YLAC/x = ' num2str(resp,'%10.4e') ' (g/cel)']);
        elseif (op == 4)
            set(lblY,'String',['YLAC/GLC = ' num2str(resp,'%10.4e') ' (g/g)']);
        elseif (op == 5)
            set(lblY,'String',['YNH4/x = ' num2str(resp,'%10.4e') ' (g/cel)']);
        elseif (op == 6)
            set(lblY,'String',['YNH4/GLN = ' num2str(resp,'%10.4e') ' (g/g)']);
        elseif (op == 7)
            set(lblY,'String',['Y NomeProduto '/GLC = ' num2str(resp,'%10.4e') ' (µg/g)']);
        elseif (op == 8)

```

```

        set(lblY,'String',['YALA/GLC ' num2str(resp,'%10.4e') ' (g/g)']);
    elseif (op == 9)
        set(lblY,'String',['YALA/GLN ' num2str(resp,'%10.4e') ' (g/g)']);
    end
    set(lblR,'String',['R = ' num2str(R_resp,4)]);
    cla(hEixol);
    % Desenha Rótulos
    xlabel(RotuloX,'Parent',hEixol);
    ylabel(RotuloY,'Parent',hEixol);
    % Exibe mensagem
    limiteX = get(hEixol,'Xlim');
    limiteY = get(hEixol,'Ylim');
    text('Position',[limiteX(1)+(limiteX(2)-limiteX(1))*0.2 (limiteY(1)+limiteY(2))/2],...
        'String','Pontos insuficientes para efetuar ajuste!','...
        'FontName','Arial',...
        'FontSize',14,'Color','r');
    return
end
% Faz Interpolação
xi = primeiro:passo_interp:ultimo;
warning off
p = polyfit(xinter,yinter,1); % polinômio de grau 1
warning on
yi = polyval(p,xi);
% Respostas
resp = abs(p(1));
R_resp = corrcoef(xinter,yinter);
R_resp = R_resp(1,2);
if (op == 1)
    resp = resp * 1000; % ajustar unidades L e mL
    set(lblY,'String',['Yx/GLC = ' num2str(resp,'%10.4e') ' (cel/g)']);
elseif (op == 2)
    resp = resp * 1000; % ajustar unidades L e mL
    set(lblY,'String',['Yx/GLN = ' num2str(resp,'%10.4e') ' (cel/g)']);
elseif (op == 3)
    resp = resp / 1000; % ajustar unidades L e mL
    set(lblY,'String',['YLAC/x = ' num2str(resp,'%10.4e') ' (g/cel)']);
elseif (op == 4)
    set(lblY,'String',['YLAC/GLC = ' num2str(resp,'%10.4e') ' (g/g)']);
elseif (op == 5)
    resp = resp / 1000; % ajustar unidades L e mL
    set(lblY,'String',['YNH4/x = ' num2str(resp,'%10.4e') ' (g/cel)']);
elseif (op == 6)
    set(lblY,'String',['YNH4/GLN = ' num2str(resp,'%10.4e') ' (g/g)']);
elseif (op == 7)
    set(lblY,'String',['Y' NomeProduto '/GLC ' num2str(resp,'%10.4e') ' (µg/g)']);
elseif (op == 8)
    set(lblY,'String',['YALA/GLC ' num2str(resp,'%10.4e') ' (g/g)']);
elseif (op == 9)
    set(lblY,'String',['YALA/GLN ' num2str(resp,'%10.4e') ' (g/g)']);
end
set(lblR,'String',['R = ' num2str(R_resp,4)]);
% Verifica pontos modificados que não foram removidos
cont = 1;
xmod = [];
ymod = [];
for i=1:length(yf)
    if (xf(i)~=realmin && yf(i)~=realmin) % ponto modificado i não foi removido
        xmod(cont) = xf(i); %ok<AGROW>
        ymod(cont) = yf(i); %ok<AGROW>
        cont = cont + 1;
    end
end
xmod = xmod';
ymod = ymod';
% Limpa eixos
cla(hEixol);
% Plota dados originais, modificados, usados e interpolados
if (cont==1)
    plot(hEixol,x,y,'or',xinter,yinter,'bs',xi,yi,'k-');
else
    plot(hEixol,x,y,'or',xmod,ymod,'sk',xinter,yinter,'g.',xi,yi,'k-');
end
% Desenha Rótulos e legenda
xlabel(RotuloX,'Parent',hEixol);
ylabel(RotuloY,'Parent',hEixol);
% Ajusta legendas
if (cont==1)
    legend(hEixol,'Original','Usado','Regressão','Location',Legenda);
else
    legend(hEixol,'Original','Modificado','Usado','Regressão','Location',Legenda);
end
% Ajusta limites do Eixol
xlim(hEixol, [ min([x;xmod]) (max([x;xmod])+(max([x;xmod])-min([x;xmod]))./3) ] );
% Copyright
limiteX = get(hEixol,'Xlim');
limiteY = get(hEixol,'Ylim');
text('Position',[limiteX(1)+(limiteX(2)-limiteX(1))*1.1 limiteY(1)],...

```

```
% *****
function r = Estah_Vazio(v)
r = 1; % Verdadeiro (está vazio)
for i = 1:length(v)
    if (v(i)~= -1)
        r = 0; % Falso (não está vazio)
    end
end
end
```

5. Graficos.m

```
% *****
% Módulo: Graficos.m
% Descrição: Permite ajuste de pontos nos gráficos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Desenha Gráfico Geral
function Graficos
% Variáveis Globais
VariaveisGlobais

% Cria timer para atualizar dados do painel
TimerAtualiza = timer('TimerFcn', @Atualiza);

% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigGG = figure('WindowButtonDownFcn', @hFigWindowButtonDownFcn, ...
    'name','Ajuste de pontos','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'ToolBar','figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos (gráfico)
hEixo1 = axes('Parent',hFigGG, ...
    'Position',[.07 .26 .84 .57],...
    'Color','none',... % Transparente (nenhuma cor de fundo)
    'Tag', 'Eixo1');

hold on;
hEixoXv = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
    'Visible','off',... % Invisível
    'Tag', 'EixoXv');

hold on;
if (~Estah_Vazio(oProd))
    hEixoProd = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoProd');

    hold on;
end
if (~Estah_Vazio(oProde))
    hEixoProde = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoProde');

    hold on;
end
if (~Estah_Vazio(oGLC))
    hEixoGLC = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLC');

    hold on;
end
if (~Estah_Vazio(oGLN))
    hEixoGLN = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLN');

    hold on;
end
if (~Estah_Vazio(oGLU))
    hEixoGLU = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLU');

    hold on;
end
if (~Estah_Vazio(oLAC))
    hEixoLAC = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
```

```
% *****
function r = Estah_Vazio(v)
r = 1; % Verdadeiro (está vazio)
for i = 1:length(v)
    if (v(i)~= -1)
        r = 0; % Falso (não está vazio)
    end
end
end
```

5. Graficos.m

```
% *****
% Módulo: Graficos.m
% Descrição: Permite ajuste de pontos nos gráficos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Desenha Gráfico Geral
function Graficos
% Variáveis Globais
VariaveisGlobais

% Cria timer para atualizar dados do painel
TimerAtualiza = timer('TimerFcn', @Atualiza);

% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigGG = figure('WindowButtonDownFcn', @hFigWindowButtonDownFcn, ...
    'name','Ajuste de pontos','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'ToolBar','figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos (gráfico)
hEixo1 = axes('Parent',hFigGG, ...
    'Position',[.07 .26 .84 .57],...
    'Color','none',... % Transparente (nenhuma cor de fundo)
    'Tag', 'Eixo1');

hold on;
hEixoXv = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
    'Visible','off',... % Invisível
    'Tag', 'EixoXv');

hold on;
if (~Estah_Vazio(oProd))
    hEixoProd = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoProd');

    hold on;
end
if (~Estah_Vazio(oProde))
    hEixoProde = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoProde');

    hold on;
end
if (~Estah_Vazio(oGLC))
    hEixoGLC = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLC');

    hold on;
end
if (~Estah_Vazio(oGLN))
    hEixoGLN = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLN');

    hold on;
end
if (~Estah_Vazio(oGLU))
    hEixoGLU = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',... % Invisível
        'Tag', 'EixoGLU');

    hold on;
end
if (~Estah_Vazio(oLAC))
    hEixoLAC = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
```

```

        'Visible','off',...           % Invisível
        'Tag', 'EixoLAC');

    hold on;
end
if (~Estah_Vazio(oNH4))
    hEixoNH4 = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',...           % Invisível
        'Tag', 'EixoNH4');

    hold on;
end
if (~Estah_Vazio(oALA))
    hEixoALA = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',...           % Invisível
        'Tag', 'EixoALA');

    hold on;
end
if (~Estah_Vazio(oViab))
    hEixoViab = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',...           % Invisível
        'YLim',[0 1],...             % Limites do Eixo Y
        'Tag', 'EixoViab');

    hold on;
end
if (~Estah_Vazio(oProd))
    hEixoPxt = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
        'Visible','off',...           % Invisível
        'Tag', 'EixoPxt');

    hold on;
end
hEixoMi = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
    'Visible','off',...           % Invisível
    'Tag', 'EixoMi');

hold on;
hEixoLnXv = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
    'Visible','off',...           % Invisível
    'Tag', 'EixoLnXv');

hold on;
hEixoVert = axes('Position',get(hEixo1,'Position'),... % Cria Eixo na mesma posição do Eixo 1
    'Visible','off',...           % Invisível
    'Tag', 'hEixoVert');

hold on;
% Cria controles
lblAlisa = uicontrol(hFigGG,'Style','text',...
    'String', 'Alisamento:',...
    'BackgroundColor','w', ...
    'FontSize', 8, 'FontName','Arial',...
    'FontWeight','bold', ...
    'Tag','lblAlisa');

lblEstado = uicontrol(hFigGG,'Style','text',...
    'String', 'Selecione um ponto!',...
    'BackgroundColor','w', ...
    'ForegroundColor','r',...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', ...
    'Tag', 'lblEstado');

btnInserir = uicontrol(hFigGG,'Callback', @btnInserirClick, ...
    'Style','pushbutton',...
    'String', 'Inserir',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnInserir');

btnRemover = uicontrol(hFigGG,'Callback', @btnRemoverClick, ...
    'Style','pushbutton',...
    'String', 'Remover',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnRemover');

btnDesfazer = uicontrol(hFigGG,'Callback', @btnDesfazerClick, ...
    'Style','pushbutton',...
    'String', 'Desfazer',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnDesfazer');

btnOriginal = uicontrol(hFigGG,'Callback', @btnOriginalClick,...
    'Style','pushbutton',...
    'String', 'Original',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnOriginal');

btnReiniciar = uicontrol(hFigGG,'Callback', @btnReiniciarClick, ...
    'Style','pushbutton',...
    'String', 'Reiniciar',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnReiniciar');

btnEncerrar = uicontrol(hFigGG,'Callback', @btnEncerrarClick,...
    'Style','pushbutton',...
    'String', 'Encerrar',...
    'FontSize', 8, 'FontName','Arial',...
    'Tag','btnEncerrar');

% Cria menu com opções de alisamento
mnuMetodo = uicontrol(hFigGG,'Callback', @mnuMetodoClick,...
    'Style','popupmenu',...

```

```

        'String', 'Smoothing Spline|Spline|Hermite',...
        'FontSize', 8, 'FontName','Arial',...
        'Tag','mnuMetodo');
% Cria radiogroup para selecao de variável
radGrupo = uibuttongroup('SelectionChangeFcn',@radGrupoCallback, ...
    'Units','points','pos',[ 0 0 1 1 ], ... % qualquer coordenada
    'BackgroundColor','w', 'BorderType','none', ...
    'Tag','radGrupo');
ph = 10; % Posição horizontal
radXv = uicontrol('parent',radGrupo, 'pos',[ph 20 50 13],...
    'Style','radiobutton', 'String', 'Xv',...
    'FontSize', 8, 'FontName','Arial',...
    'ForegroundColor','k', 'BackgroundColor','w',...
    'Tag','radXv');

ph = ph + 60;
if (~Estah_Vazio(oProd))
    radProd = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', NomeProduto,...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radProd');

    ph = ph + 60;
end
if (~Estah_Vazio(oProde))
    radProde = uicontrol('parent',radGrupo,'pos',[ph 20 60 13],...
        'Style','radiobutton', 'String', [ NomeProduto 'e' ],...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radProde');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLC))
    radGLC = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'GLC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radGLC');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLN))
    radGLN = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'GLN',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radGLN');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLU))
    radGLU = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'GLU',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radGLU');

    ph = ph + 60;
end
if (~Estah_Vazio(oLAC))
    radLAC = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'LAC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radLAC');

    ph = ph + 60;
end
if (~Estah_Vazio(oNH4))
    radNH4 = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'NH4',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radNH4');

    ph = ph + 60;
end
if (~Estah_Vazio(oALA))
    radALA = uicontrol('parent',radGrupo,'pos',[ph 20 50 13],...
        'Style','radiobutton', 'String', 'ALA',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','k', 'BackgroundColor','w',...
        'Tag','radALA');

    ph = ph + 60;
end
% Cria checkboxes para selecao de variável normalizada
ph = 10; % Posição horizontal
chkXv = uicontrol(hFigGG,'Callback', @chkXvClick, 'pos',[ph 3 50 13], ...
    'Style','checkbox', 'String', 'Xv',...
    'FontSize', 8, 'FontName','Arial',...
    'ForegroundColor','r', 'Tag','chkXv');

ph = ph + 60;
if (~Estah_Vazio(oProd))
    chkProd = uicontrol(hFigGG,'Callback', @chkProdClick, 'pos',[ph 3 50 13], ...

```

```

        'Style','checkbox', 'String', NomeProduto,...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.0 0.7 0.0 ], 'Tag','chkProd');

    ph = ph + 60;
end
if (~Estah_Vazio(oProde))
    chkProde = uicontrol(hFigGG,'Callback', @chkProdeClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', [ NomeProduto 'e' ],...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','b', 'Tag','chkProde');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLC))
    chkGLC = uicontrol(hFigGG,'Callback', @chkGLCClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'GLC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor','m', 'Tag','chkGLC');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLN))
    chkGLN = uicontrol(hFigGG,'Callback', @chkGLNClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'GLN',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.7 0.7 0.0 ], 'Tag','chkGLN');

    ph = ph + 60;
end
if (~Estah_Vazio(oGLU))
    chkGLU = uicontrol(hFigGG,'Callback', @chkGLUClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'GLU',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.0 0.5 0.5 ], 'Tag','chkGLU');

    ph = ph + 60;
end
if (~Estah_Vazio(oLAC))
    chkLAC = uicontrol(hFigGG,'Callback', @chkLACClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'LAC',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 1.0 0.5 0.0 ], 'Tag','chkLAC');

    ph = ph + 60;
end
if (~Estah_Vazio(oNH4))
    chkNH4 = uicontrol(hFigGG,'Callback', @chkNH4Click, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'NH4',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.5 0.5 0.5 ], 'Tag','chkNH4');

    ph = ph + 60;
end
if (~Estah_Vazio(oALA))
    chkALA = uicontrol(hFigGG,'Callback', @chkALAClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'ALA',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.5 0.3 0.1 ], 'Tag','chkALA');

    ph = ph + 60;
end
if (~Estah_Vazio(oViab))
    chkViab = uicontrol(hFigGG,'Callback', @chkViabClick, 'pos',[ph 3 50 13], ...
        'Style','checkbox', 'String', 'Viab',...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.5 0.0 0.5 ], 'Tag','chkViab');

    ph = ph + 60;
end
chkMi = uicontrol(hFigGG,'Callback', @chkMiClick, 'pos',[ph 3 50 13], ...
    'Style','checkbox', 'String', 'µx',...
    'FontSize', 8, 'FontName','Arial',...
    'ForegroundColor',[ 0.0 0.0 0.5 ], 'Tag','chkMi');

ph = ph + 60;
chkLnXv = uicontrol(hFigGG,'Callback', @chkLnXvClick, 'pos',[ph 3 50 13], ...
    'Style','checkbox', 'String', 'lnXv',...
    'FontSize', 8, 'FontName','Arial',...
    'ForegroundColor',[ 0.5 0.5 0.0 ], 'Tag','chkLnXv');

ph = ph + 60;
if (~Estah_Vazio(oProd))
    chkPXt = uicontrol(hFigGG,'Callback', @chkPXtClick, 'pos',[ph 3 90 13], ...
        'Style','checkbox', 'String', [ NomeProduto ' / Xt'],...
        'FontSize', 8, 'FontName','Arial',...
        'ForegroundColor',[ 0.0 0.5 0.0 ], 'Tag','chkPXt');

    ph = ph + 90;
end
% Define variável que está sendo manipulada
VarAtual = 'Xv';
% Atualiza valores do painel
Atualiza_LnXv;
% Faz ajuste inicial
AjusteInicial;
% Desenha curvas
Desenha;
% Mensagem de alerta
if (InicialExp==0 && FinalExp==0)

```



```

if (~Estah_Vazio(oProd))
    cla(hEixoPXt);
end
cla(hEixoMi);
legend(hEixoMi, 'off');
cla(hEixoLnXv);
legend(hEixoLnXv, 'off');
% Desmarca CheckBoxes
set(chkXv, 'Value', 0);
if (~Estah_Vazio(oProd))
    set(chkProd, 'Value', 0);
end
if (~Estah_Vazio(oProde))
    set(chkProde, 'Value', 0);
end
if (~Estah_Vazio(oGLC))
    set(chkGLC, 'Value', 0);
end
if (~Estah_Vazio(oGLN))
    set(chkGLN, 'Value', 0);
end
if (~Estah_Vazio(oGLU))
    set(chkGLU, 'Value', 0);
end
if (~Estah_Vazio(oLAC))
    set(chkLAC, 'Value', 0);
end
if (~Estah_Vazio(oNH4))
    set(chkNH4, 'Value', 0);
end
if (~Estah_Vazio(oALA))
    set(chkALA, 'Value', 0);
end
if (~Estah_Vazio(oViab))
    set(chkViab, 'Value', 0);
end
if (~Estah_Vazio(oProd))
    set(chkPXt, 'Value', 0);
end
set(chkMi, 'Value', 0);
set(chkLnXv, 'Value', 0);
% Faz ajuste inicial
AjusteInicial;
% Desenha curvas
Desenha;
end

function chkXvClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual, 'Xv')==1)
        set(hObject, 'Value', 0); % desmarca checkbox
        return;
    end
    if (get(hObject, 'Value') == get(hObject, 'Max'))
        % Desenha
        plot(hEixoXv, spt, normc(spXv), 'Color', 'r');
    else
        % Limpa
        cla(hEixoXv);
    end
end

function chkProdClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual, 'Prod')==1)
        set(hObject, 'Value', 0); % desmarca checkbox
        return;
    end
    if (get(hObject, 'Value') == get(hObject, 'Max'))
        % Desenha
        plot(hEixoProd, spt, normc(spProd), 'Color', [ 0.0 0.7 0.0 ]);
    else
        % Limpa
        cla(hEixoProd);
    end
end

function chkProdeClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual, 'Prode')==1)
        set(hObject, 'Value', 0); % desmarca checkbox
        return;
    end
    if (get(hObject, 'Value') == get(hObject, 'Max'))
        % Desenha
        plot(hEixoProde, spt, normc(spProde), 'Color', 'b');
    else
        % Limpa

```

```

if (~Estah_Vazio(oProd))
    cla(hEixoPXt);
end
cla(hEixoMi);
legend(hEixoMi,'off');
cla(hEixoLnXv);
legend(hEixoLnXv,'off');
% Desmarca CheckBoxes
set(chkXv,'Value',0);
if (~Estah_Vazio(oProd))
    set(chkProd,'Value',0);
end
if (~Estah_Vazio(oProde))
    set(chkProde,'Value',0);
end
if (~Estah_Vazio(oGLC))
    set(chkGLC,'Value',0);
end
if (~Estah_Vazio(oGLN))
    set(chkGLN,'Value',0);
end
if (~Estah_Vazio(oGLU))
    set(chkGLU,'Value',0);
end
if (~Estah_Vazio(oLAC))
    set(chkLAC,'Value',0);
end
if (~Estah_Vazio(oNH4))
    set(chkNH4,'Value',0);
end
if (~Estah_Vazio(oALA))
    set(chkALA,'Value',0);
end
if (~Estah_Vazio(oViab))
    set(chkViab,'Value',0);
end
if (~Estah_Vazio(oProd))
    set(chkPXt,'Value',0);
end
set(chkMi,'Value',0);
set(chkLnXv,'Value',0);
% Faz ajuste inicial
AjusteInicial;
% Desenha curvas
Desenha;
end

function chkXvClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'Xv')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoXv,spt,normc(spXv),'Color','r');
    else
        % Limpa
        cla(hEixoXv);
    end
end

function chkProdClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'Prod')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoProd,spt,normc(spProd),'Color',[ 0.0 0.7 0.0 ]);
    else
        % Limpa
        cla(hEixoProd);
    end
end

function chkProdeClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'Prode')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoProde,spt,normc(spProde),'Color','b');
    else
        % Limpa

```

```

        cla(hEixoProde);
    end
end

function chkGLCClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLC')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLC,spt,normc(spGLC),'m');
    else
        % Limpa
        cla(hEixoGLC);
    end
end

function chkGLNClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLN')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLN,spt,normc(spGLN),'Color', [ 0.7 0.7 0.0 ]);
    else
        % Limpa
        cla(hEixoGLN);
    end
end

function chkGLUClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLU')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLU,spt,normc(spGLU),'Color', [ 0.0 0.6 0.6 ]);
    else
        % Limpa
        cla(hEixoGLU);
    end
end

function chkLACClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'LAC')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoLAC,spt,normc(spLAC),'Color', [ 1.0 0.5 0.0 ]);
    else
        % Limpa
        cla(hEixoLAC);
    end
end

function chkNH4Click(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'NH4')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoNH4,spt,normc(spNH4),'Color', [ 0.5 0.5 0.5 ]);
    else
        % Limpa
        cla(hEixoNH4);
    end
end

function chkALAClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'ALA')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoALA,spt,normc(spALA),'Color', [ 0.5 0.3 0.1 ]);
    end
end

```

```

        cla(hEixoProde);
    end
end

function chkGLCClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLC')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLC,spt,normc(spGLC),'m');
    else
        % Limpa
        cla(hEixoGLC);
    end
end

function chkGLNClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLN')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLN,spt,normc(spGLN),'Color', [ 0.7 0.7 0.0 ]);
    else
        % Limpa
        cla(hEixoGLN);
    end
end

function chkGLUClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'GLU')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoGLU,spt,normc(spGLU),'Color', [ 0.0 0.6 0.6 ]);
    else
        % Limpa
        cla(hEixoGLU);
    end
end

function chkLACClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'LAC')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoLAC,spt,normc(spLAC),'Color', [ 1.0 0.5 0.0 ]);
    else
        % Limpa
        cla(hEixoLAC);
    end
end

function chkNH4Click(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'NH4')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoNH4,spt,normc(spNH4),'Color', [ 0.5 0.5 0.5 ]);
    else
        % Limpa
        cla(hEixoNH4);
    end
end

function chkALAClick(hObject, eventdata, handles)
    VariaveisGlobais
    if (strcmp(VarAtual,'ALA')==1)
        set(hObject,'Value',0); % desmarca checkbox
        return;
    end
    if (get(hObject,'Value') == get(hObject,'Max'))
        % Desenha
        plot(hEixoALA,spt,normc(spALA),'Color', [ 0.5 0.3 0.1 ]);
    end
end

```

```

end
if (strcmp(VarAtual,'Prode')==1)
    x = at;          y = aProde;          % Pontos originais + inseridos
    xins = instProde; yins = insProde;     % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mProde;         % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLC')==1)
    x = at;          y = aGLC;             % Pontos originais + inseridos
    xins = instGLC;  yins = insGLC;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLC;           % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLN')==1)
    x = at;          y = aGLN;             % Pontos originais + inseridos
    xins = instGLN;  yins = insGLN;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLN;           % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLU')==1)
    x = at;          y = aGLU;             % Pontos originais + inseridos
    xins = instGLU;  yins = insGLU;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLU;           % Inicializa Dados finais
end
if (strcmp(VarAtual,'LAC')==1)
    x = at;          y = aLAC;             % Pontos originais + inseridos
    xins = instLAC;  yins = insLAC;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mLAC;           % Inicializa Dados finais
end
if (strcmp(VarAtual,'NH4')==1)
    x = at;          y = aNH4;             % Pontos originais + inseridos
    xins = instNH4;  yins = insNH4;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mNH4;           % Inicializa Dados finais
end
if (strcmp(VarAtual,'ALA')==1)
    x = at;          y = aALA;             % Pontos originais + inseridos
    xins = instALA;  yins = insALA;         % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mALA;           % Inicializa Dados finais
end
% Inicializa undo
pt = 1;
y_undo = realmin; % não pode fazer undo inicialmente
end

function AjusteFinal
VariaveisGlobais
% Guarda pontos atuais, interpolados, modificados e inseridos
if (strcmp(VarAtual,'Xv')==1)
    at = x;          aXv = y;             % Pontos atuais (originais + inseridos)
    spt = xi;        spXv = yi;           % Pontos interpolados
    mt = xf;          mXv = yf;           % Pontos modificados
    instXv = xins;   insXv = yins;        % Pontos inseridos
end
if (strcmp(VarAtual,'Prod')==1)
    at = x;          aProd = y;           % Pontos atuais (originais + inseridos)
    spt = xi;        spProd = yi;         % Pontos interpolados
    mt = xf;          mProd = yf;         % Pontos modificados
    instProd = xins; insProd = yins;      % Pontos inseridos
end
if (strcmp(VarAtual,'Prode')==1)
    at = x;          aProde = y;          % Pontos atuais (originais + inseridos)
    spt = xi;        spProde = yi;        % Pontos interpolados
    mt = xf;          mProde = yf;        % Pontos modificados
    instProde = xins; insProde = yins;    % Pontos inseridos
end
if (strcmp(VarAtual,'GLC')==1)
    at = x;          aGLC = y;            % Pontos atuais (originais + inseridos)
    spt = xi;        spGLC = yi;          % Pontos interpolados
    mt = xf;          mGLC = yf;          % Pontos modificados
    instGLC = xins;  insGLC = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'GLN')==1)
    at = x;          aGLN = y;            % Pontos atuais (originais + inseridos)
    spt = xi;        spGLN = yi;          % Pontos interpolados
    mt = xf;          mGLN = yf;          % Pontos modificados
    instGLN = xins;  insGLN = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'GLU')==1)
    at = x;          aGLU = y;            % Pontos atuais (originais + inseridos)
    spt = xi;        spGLU = yi;          % Pontos interpolados
    mt = xf;          mGLU = yf;          % Pontos modificados
    instGLU = xins;  insGLU = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'LAC')==1)
    at = x;          aLAC = y;            % Pontos atuais (originais + inseridos)
    spt = xi;        spLAC = yi;          % Pontos interpolados
    mt = xf;          mLAC = yf;          % Pontos modificados
    instLAC = xins;  insLAC = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'NH4')==1)
    at = x;          aNH4 = y;            % Pontos atuais (originais + inseridos)

```

```

end
if (strcmp(VarAtual,'Prode')==1)
    x = at;          y = aProde;          % Pontos originais + inseridos
    xins = instProde; yins = insProde;    % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mProde;        % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLC')==1)
    x = at;          y = aGLC;            % Pontos originais + inseridos
    xins = instGLC;  yins = insGLC;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLC;          % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLN')==1)
    x = at;          y = aGLN;            % Pontos originais + inseridos
    xins = instGLN;  yins = insGLN;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLN;          % Inicializa Dados finais
end
if (strcmp(VarAtual,'GLU')==1)
    x = at;          y = aGLU;            % Pontos originais + inseridos
    xins = instGLU;  yins = insGLU;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mGLU;          % Inicializa Dados finais
end
if (strcmp(VarAtual,'LAC')==1)
    x = at;          y = aLAC;            % Pontos originais + inseridos
    xins = instLAC;  yins = insLAC;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mLAC;          % Inicializa Dados finais
end
if (strcmp(VarAtual,'NH4')==1)
    x = at;          y = aNH4;            % Pontos originais + inseridos
    xins = instNH4;  yins = insNH4;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mNH4;          % Inicializa Dados finais
end
if (strcmp(VarAtual,'ALA')==1)
    x = at;          y = aALA;            % Pontos originais + inseridos
    xins = instALA;  yins = insALA;        % Inicializa vetores de pontos inseridos
    xf = mt;          yf = mALA;          % Inicializa Dados finais
end
% Inicializa undo
pt = 1;
y_undo = realmin; % não pode fazer undo inicialmente
end

function AjusteFinal
VariaveisGlobais
% Guarda pontos atuais, interpolados, modificados e inseridos
if (strcmp(VarAtual,'Xv')==1)
    at = x;          aXv = y;            % Pontos atuais (originais + inseridos)
    spt = xi;         spXv = yi;          % Pontos interpolados
    mt = xf;          mXv = yf;          % Pontos modificados
    instXv = xins;    insXv = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'Prod')==1)
    at = x;          aProd = y;           % Pontos atuais (originais + inseridos)
    spt = xi;         spProd = yi;        % Pontos interpolados
    mt = xf;          mProd = yf;         % Pontos modificados
    instProd = xins;  insProd = yins;     % Pontos inseridos
end
if (strcmp(VarAtual,'Prode')==1)
    at = x;          aProde = y;          % Pontos atuais (originais + inseridos)
    spt = xi;         spProde = yi;       % Pontos interpolados
    mt = xf;          mProde = yf;        % Pontos modificados
    instProde = xins; insProde = yins;    % Pontos inseridos
end
if (strcmp(VarAtual,'GLC')==1)
    at = x;          aGLC = y;            % Pontos atuais (originais + inseridos)
    spt = xi;         spGLC = yi;         % Pontos interpolados
    mt = xf;          mGLC = yf;          % Pontos modificados
    instGLC = xins;  insGLC = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'GLN')==1)
    at = x;          aGLN = y;            % Pontos atuais (originais + inseridos)
    spt = xi;         spGLN = yi;         % Pontos interpolados
    mt = xf;          mGLN = yf;          % Pontos modificados
    instGLN = xins;  insGLN = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'GLU')==1)
    at = x;          aGLU = y;            % Pontos atuais (originais + inseridos)
    spt = xi;         spGLU = yi;         % Pontos interpolados
    mt = xf;          mGLU = yf;          % Pontos modificados
    instGLU = xins;  insGLU = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'LAC')==1)
    at = x;          aLAC = y;            % Pontos atuais (originais + inseridos)
    spt = xi;         spLAC = yi;         % Pontos interpolados
    mt = xf;          mLAC = yf;          % Pontos modificados
    instLAC = xins;  insLAC = yins;       % Pontos inseridos
end
if (strcmp(VarAtual,'NH4')==1)
    at = x;          aNH4 = y;            % Pontos atuais (originais + inseridos)

```

```

        spt = xi;          spNH4 = yi;          % Pontos interpolados
        mt = xf;          mNH4 = yf;          % Pontos modificados
        instNH4 = xins;    insNH4 = yins;      % Pontos inseridos
    end
    if (strcmp(VarAtual,'ALA')==1)
        at = x;          aALA = y;          % Pontos atuais (originais + inseridos)
        spt = xi;          spALA = yi;        % Pontos interpolados
        mt = xf;          mALA = yf;        % Pontos modificados
        instALA = xins;    insALA = yins;    % Pontos inseridos
    end
end

function mnuMetodoClick(hObject, eventdata, handles)
    VariaveisGlobais
    TipoAjuste = get(findobj('Tag','mnuMetodo'),'Value');
    if (TipoAjuste == 1)
        prompt = sprintf('Digite o fator de suavização (entre 0.0 e 1.0)\nPara utilizar o fator default digite -1');
        dlg_title = 'Fator de suavização';
        num_lines = 1;
        def = { num2str(Fator) };
        answer = inputdlg(prompt,dlg_title,num_lines,def);
        if (~isempty(answer))
            Fator = str2num(cell2mat(answer));
        end
    end
    Desenha;
end

function btnInserirClick(hObject,eventdata,handles)
    VariaveisGlobais
    if (modo == -1) %#ok<NODEF>
        % Define posição de inserção
        p_ins = 1;
        x = [ x; realmax ]; %#ok<NODEF>
        while (x(p_ins) < xc && p_ins <= length(x))
            p_ins = p_ins + 1;
        end
        % Desloca elementos da variável atual
        for i = length(x)-1 : -1 : p_ins
            x(i+1) = x(i);
            y(i+1) = y(i);    %#ok<AGROW>
            xf(i+1) = xf(i);  %#ok<AGROW>
            yf(i+1) = yf(i);  %#ok<AGROW>
        end
        % Desloca elementos de todas as variáveis
        for i = length(x)-1 : -1 : p_ins
            at(i+1) = at(i);    %#ok<AGROW>
            aXv(i+1) = aXv(i);  %#ok<AGROW>
            aGLC(i+1) = aGLC(i); %#ok<AGROW>
            aGLN(i+1) = aGLN(i); %#ok<AGROW>
            aGLU(i+1) = aGLU(i); %#ok<AGROW>
            aLAC(i+1) = aLAC(i); %#ok<AGROW>
            aNH4(i+1) = aNH4(i); %#ok<AGROW>
            aALA(i+1) = aALA(i); %#ok<AGROW>
            aProde(i+1) = aProde(i); %#ok<AGROW>
            aProd(i+1) = aProd(i); %#ok<AGROW>
            mt(i+1) = mt(i);    %#ok<AGROW>
            mXv(i+1) = mXv(i);  %#ok<AGROW>
            mGLC(i+1) = mGLC(i); %#ok<AGROW>
            mGLN(i+1) = mGLN(i); %#ok<AGROW>
            mGLU(i+1) = mGLU(i); %#ok<AGROW>
            mLAC(i+1) = mLAC(i); %#ok<AGROW>
            mNH4(i+1) = mNH4(i); %#ok<AGROW>
            mALA(i+1) = mALA(i); %#ok<AGROW>
            mProde(i+1) = mProde(i); %#ok<AGROW>
            mProd(i+1) = mProd(i); %#ok<AGROW>
        end
        % Insere ponto nas variável atual
        x(p_ins) = xc;
        y(p_ins) = yc;
        xf(p_ins) = xc;
        yf(p_ins) = yc;
        % Insere ponto em todas as variáveis
        at(p_ins) = xc;
        aXv(p_ins) = -1;
        aGLC(p_ins) = -1;
        aGLN(p_ins) = -1;
        aGLU(p_ins) = -1;
        aLAC(p_ins) = -1;
        aNH4(p_ins) = -1;
        aALA(p_ins) = -1;
        aProde(p_ins) = -1;
        aProd(p_ins) = -1;
        mt(p_ins) = xc;
        mXv(p_ins) = -1;
        mGLC(p_ins) = -1;
        mGLN(p_ins) = -1;
    end
end

```

```

mGLU(p_ins) = -1;
mLAC(p_ins) = -1;
mNH4(p_ins) = -1;
mALA(p_ins) = -1;
mProde(p_ins) = -1;
mProd(p_ins) = -1;
% Adiciona ao vetor de pontos inseridos da variável atual
xins(length(xins)+1) = xc; %#ok<NODEF>
yins(length(yins)+1) = yc; %#ok<NODEF>
% Dados para o undo
pt = p_ins;
y_undo = realmax;
% Ajusta modo
modo = 1;
set(findobj('Tag','lblEstado'),'String',...
    'Selecione um ponto ou clique em "Desfazer" ou "Reiniciar!');
% Desenha curvas
Desenha;
end
end

function btnRemoverClick(hObject,eventdata,handles)
VariaveisGlobais
if (modo == -1) %#ok<NODEF>
% Verifica se é um ponto que foi inserido
flag = false;
p_ins = 1;
while (p_ins <= length(xins) && flag == false) %#ok<NODEF>
    if (x(pt) == xins(p_ins)) %#ok<NODEF>
        flag = true;
        break;
    end
    p_ins = p_ins + 1;
end
if (flag == true) % Elimina ponto inserido
% Elimina da variável atual
x(pt) = [];
y(pt) = [];
xf(pt) = [];
yf(pt) = [];
% Elimina de todas as variáveis
at(pt) = [];
aXv(pt) = [];
aGLC(pt) = [];
aGLN(pt) = [];
aGLU(pt) = [];
aLAC(pt) = [];
aNH4(pt) = [];
aALA(pt) = [];
aProde(pt) = [];
aProd(pt) = [];
mt(pt) = [];
mXv(pt) = [];
mGLC(pt) = [];
mGLN(pt) = [];
mGLU(pt) = [];
mLAC(pt) = [];
mNH4(pt) = [];
mALA(pt) = [];
mProde(pt) = [];
mProd(pt) = [];
% Remove do vetor de pontos inseridos
xins(p_ins) = [];
yins(p_ins) = [];
% Ajusta dados do undo
y_undo = realmin; % não pode fazer undo
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Reiniciar!');
else % Elimina ponto original
y_undo = yf(pt); %#ok<NODEF>
yf(pt) = realmin;
set(findobj('Tag','lblEstado'),'String',...
    'Selecione um ponto ou clique em "Desfazer" ou "Reiniciar!');
end
modo = 1;
% Desenha curvas
Desenha;
end
end

function btnDesfazerClick(hObject,eventdata,handles)
VariaveisGlobais
if (modo == 1) %#ok<NODEF>
    if (y_undo == realmax) %#ok<NODEF> % eliminar ponto inserido
        % Elimina da variável atual
        x(pt) = [];
        y(pt) = [];
        xf(pt) = [];
        yf(pt) = [];
    end
end

```



```

% Elimina de todas as variáveis
at(pt) = [];
aXv(pt) = [];
aGLC(pt) = [];
aGLN(pt) = [];
aGLU(pt) = [];
aLAC(pt) = [];
aNH4(pt) = [];
aALA(pt) = [];
aProde(pt) = [];
aProd(pt) = [];
mt(pt) = [];
mXv(pt) = [];
mGLC(pt) = [];
mGLN(pt) = [];
mGLU(pt) = [];
mLAC(pt) = [];
mNH4(pt) = [];
mALA(pt) = [];
mProde(pt) = [];
mProd(pt) = [];
% Remove do vetor de pontos inseridos
xins(length(xins)) = []; %#ok<NODEF>
yins(length(yins)) = []; %#ok<NODEF>
modo = 1;
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Reiniciar!');
y_undo = realmin; % não pode fazer undo depois de outro undo
Desenha;
elseif (y_undo ~= realmin) % restaurar ponto movido ou removido
yf(pt) = y_undo;
modo = 1;
set(findobj('Tag','lblEstado'),'String', 'Selecione um ponto ou clique em "Reiniciar!');
y_undo = realmin; % não pode fazer undo depois de outro undo
Desenha;
end
end

end

function btnOriginalClick(hObject,eventdata,handles)
VariaveisGlobais
if (modo== -1) %#ok<NODEF>
y_undo = yf(pt); %#ok<NODEF>
yf(pt) = y(pt);
modo = 1;
set(findobj('Tag','lblEstado'),'String',...
'Selecione um ponto ou clique em "Desfazer" ou "Reiniciar!');
Desenha;
end
end

function btnReiniciarClick(hObject,eventdata,handles)
VariaveisGlobais
if (modo == 1) %#ok<NODEF>
% Inicialização Xv
if (strcmp(VarAtual,'Xv')==1)
aXv = copiados1(at,ot,oXv); % Pontos atuais
mXv = aXv; % Pontos modificados
instXv = []; insXv = []; % Pontos inseridos
if (get(chkMiCte,'Value')==1)
Recalcula_Xv;
end
[spt spXv] = Ajusta_Curva(ot,oXv,TipoAjuste); % Interpolação
end
% Inicialização Prod
if (strcmp(VarAtual,'Prod')==1)
aProd = copiados1(at,ot,oProd); % Pontos atuais
mProd = aProd; % Pontos modificados
instProd = []; insProd = []; % Pontos inseridos
[spt spProd] = Ajusta_Curva(ot,oProd,TipoAjuste); % Interpolação
end
% Inicialização Prode
if (strcmp(VarAtual,'Prode')==1)
aProde = copiados1(at,ot,oProde); % Pontos atuais
mProde = aProde; % Pontos modificados
instProde = []; insProde = []; % Pontos inseridos
[spt spProde] = Ajusta_Curva(ot,oProde,TipoAjuste); % Interpolação
end
% Inicialização GLC
if (strcmp(VarAtual,'GLC')==1)
aGLC = copiados1(at,ot,oGLC); % Pontos atuais
mGLC = aGLC; % Pontos modificados
instGLC = []; insGLC = []; % Pontos inseridos
[spt spGLC] = Ajusta_Curva(ot,oGLC,TipoAjuste); % Interpolação
end
% Inicialização GLN
if (strcmp(VarAtual,'GLN')==1)
aGLN = copiados1(at,ot,oGLN); % Pontos atuais

```

```

        mGLN = aGLN; % Pontos modificados
        instGLN = []; insGLN = []; % Pontos inseridos
        [spt spGLN] = Ajusta_Curva(ot,oGLN,TipoAjuste); % Interpolação
    end
    % Inicialização GLU
    if (strcmp(VarAtual,'GLU')==1)
        aGLU = copiados1(at,ot,oGLU); % Pontos atuais
        mGLU = aGLU; % Pontos modificados
        instGLU = []; insGLU = []; % Pontos inseridos
        [spt spGLU] = Ajusta_Curva(ot,oGLU,TipoAjuste); % Interpolação
    end
    % Inicialização LAC
    if (strcmp(VarAtual,'LAC')==1)
        aLAC = copiados1(at,ot,oLAC); % Pontos atuais
        mLAC = aLAC; % Pontos modificados
        instLAC = []; insLAC = []; % Pontos inseridos
        [spt spLAC] = Ajusta_Curva(ot,oLAC,TipoAjuste); % Interpolação
    end
    % Inicialização NH4
    if (strcmp(VarAtual,'NH4')==1)
        aNH4 = copiados1(at,ot,oNH4); % Pontos atuais
        mNH4 = aNH4; % Pontos modificados
        instNH4 = []; insNH4 = []; % Pontos inseridos
        [spt spNH4] = Ajusta_Curva(ot,oNH4,TipoAjuste); % Interpolação
    end
    % Inicialização ALA
    if (strcmp(VarAtual,'ALA')==1)
        aALA = copiados1(at,ot,oALA); % Pontos atuais
        mALA = aALA; % Pontos modificados
        instALA = []; insALA = []; % Pontos inseridos
        [spt spALA] = Ajusta_Curva(ot,oALA,TipoAjuste); % Interpolação
    end
    AjusteInicial;
    modo = 1;
    set(findobj('Tag','lblEstado'),'String','Selecione um ponto!');
    Desenha;
end
end

function btnEncerrarClick(hObject,eventdata,handles)
    VariaveisGlobais
    % Atualiza dados da variável atual
    AjusteFinal;
    % Se painel estiver aberto, fecha
    if ishandle(hFigPainel)
        close(hFigPainel)
    end
    % Fecha gráficos
    close(hFigGG)
    % Ajusta modo
    modo = 2;
    % sair da pausa
    uiresume;
    close all;
end

function hFigWindowButtonDownFcn(hObject, eventdata)
    % Variáveis Globais
    VariaveisGlobais
    % Pega posição do último click
    posClick = get(hEixo1,'CurrentPoint');
    xc = posClick(1,1); % x do click
    yc = posClick(1,2); % y do click
    % Não aceita clicks fora do Eixo
    LimiteX = get(hEixo1,'XLim');
    LimiteY = get(hEixo1,'YLim');
    if (xc<LimiteX(1) || yc<LimiteY(1) || xc>LimiteX(2) || yc>LimiteY(2))
        return;
    end
    % Troca modo
    modo = -modo; %#ok<NODEF>
    % Verifica modo (-1 = primeiro click, 1 = segundo click, 2 = edição encerrada)
    if (modo == -1)
        % Verifica ponto mais próximo do click
        pt = 1;
        for i=2:length(xf)
            if (abs(xc-xf(i)) < abs(xc-xf(pt)))
                pt = i;
            end
        end
        set(findobj('Tag','lblEstado'),'String',...
            'Selecione a nova posição ou clique em "Inserir", "Remover" ou "Original"!');
    end
    if (modo == 1)
        % Altera ponto
        y_undo = yf(pt); %#ok<NODEF>
        yf(pt) = yc;
        set(findobj('Tag','lblEstado'),'String',...

```

```

        'Selecione um ponto ou clique em "Desfazer" ou "Reiniciar!');
    % Desenha curvas
    Desenha;
end
end

function Desenha
% Variáveis Globais
VariaveisGlobais
% Verifica pontos que foram modificados
cont_mod = 1;
xmod = []; % criar xmod e ymod, mesmo que nenhum ponto tenha sido
ymod = []; % modificado, para evitar problema na plotagem
for i=1:length(x)
    if (y(i)~=yf(i) && yf(i)~=realmin) % ponto i foi modificado
        xmod(cont_mod) = xf(i); %ok<AGROW>
        ymod(cont_mod) = yf(i); %ok<AGROW>
        cont_mod = cont_mod + 1;
    end
end
% Verifica pontos que foram removidos
cont_rem = 1;
cont_inter = 1;
xrem=[];
yrem=[];
xinter = [];
yinter = [];
for i=1:length(x)
    if (yf(i)==realmin) % ponto i foi removido
        xrem(cont_rem) = x(i); %ok<AGROW>
        yrem(cont_rem) = y(i); %ok<AGROW>
        cont_rem = cont_rem + 1;
    else
        xinter(cont_inter) = xf(i); %ok<AGROW>
        yinter(cont_inter) = yf(i); %ok<AGROW>
        cont_inter = cont_inter + 1;
    end
end
% Limpa Eixol
cla(hEixol);
% Faz interpolação da variável atual
[ xi yi ] = Ajusta_Curva(xinter,yinter,TipoAjuste);
% Faz interpolação de Xv usando CubicSpline
% Estes valores serão usados apenas na função Atualiza_Mi
% para que a curva de mi fique melhor
if (strcmp(VarAtual,'Xv')==1)
    [ spxi spyi ] = Ajusta_Curva(xinter,yinter,2);
end
% Remove pontos inexistentes (-1)
[ xxi yyi ] = Remove_Pontos(xi,yi);
[ xx yy ] = Remove_Pontos(x,y);
% Plota dados originais, interpolados, modificados, inseridos e removidos
plot(hEixol,xx,yy,'or');
plot(hEixol,xxi,yyi,'k-');
plot(hEixol,xmod,ymod,'sk');
plot(hEixol,xins,yins,'g*');
plot(hEixol,xrem,yrem,'xb');
% Desenha Rótulos e legenda
xlabel('Tempo (dias)', 'Parent', hEixol);
if (strcmp(VarAtual,'Xv')==1)
    ylabel('Células Viáveis (cel/mL)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'Prod')==1)
    ylabel( [ NomeProduto ' (µg/L)' ], 'Parent', hEixol);
end
if (strcmp(VarAtual,'Prode')==1)
    ylabel( [ NomeProduto 'e (ng/1e7 cel)' ], 'Parent', hEixol);
end
if (strcmp(VarAtual,'GLC')==1)
    ylabel('GLC (g/L)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'GLN')==1)
    ylabel('GLN (g/L)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'GLU')==1)
    ylabel('GLU (g/L)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'LAC')==1)
    ylabel('LAC (g/L)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'NH4')==1)
    ylabel('NH4 (g/L)', 'Parent', hEixol);
end
if (strcmp(VarAtual,'ALA')==1)
    ylabel('ALA (g/L)', 'Parent', hEixol);
end
% Ajusta legendas
cont_ins = length(xins)+1;

```

```

set(fig,'Units','pixels');
figpos = get(fig,'Position');
% Dimensões da área útil
Largura = figpos(3);
Altura = figpos(4);
% Reposiciona lblEstado
u = findobj('Tag','lblEstado');
upos = [0, 45, Largura, 12];
set(u,'Position',upos);
% Reposiciona lblAlisa
u = findobj('Tag','lblAlisa');
upos = [10, Altura-30, 90, 20];
set(u,'Position',upos);
% Reposiciona mnuMetodo
u = findobj('Tag','mnuMetodo');
upos = [100, Altura-15, 120, 10];
set(u,'Position',upos);
% Reposiciona btnInserir
u = findobj('Tag','btnInserir');
upos = [Largura-540, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnRemover
u = findobj('Tag','btnRemover');
upos = [Largura-450, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnDesfazer
u = findobj('Tag','btnDesfazer');
upos = [Largura-360, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnOriginal
u = findobj('Tag','btnOriginal');
upos = [Largura-270, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnReiniciar
u = findobj('Tag','btnReiniciar');
upos = [Largura-180, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnEncerrar
u = findobj('Tag','btnEncerrar');
upos = [Largura-90, Altura-25, 80, 20];
set(u,'Position',upos);
% Finalização
set(fig,'Units',old_units);
end

% Cria vetor com dados originais da variável
% Insere -1 nas posições onde não existe tempo original correspondente
function [ aVar ] = copiados1(tVar, origX, origY)
aVar = zeros(length(tVar),1);
for i=1:length(tVar)
    p = find(origX==tVar(i)); % Verifica se tempo existe no original
    if (isempty(p))
        aVar(i) = -1;
    else
        aVar(i) = origY(p);
    end
end
end
end

```

6. GraficosFinais.m

```

% *****
% Módulo: GraficosFinais.m
% Descrição: Desenha gráficos após ajuste de pontos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Desenha Gráficos Finais
function GraficosFinais
% Variáveis Globais
VariaveisGlobais
% Exibe Gráficos
GraficoLnXv;
GraficoMi;
GraficoViab;
if (~Estah_Vazio(oProd))

```

```

set(fig,'Units','pixels');
figpos = get(fig,'Position');
% Dimensões da área útil
Largura = figpos(3);
Altura = figpos(4);
% Reposiciona lblEstado
u = findobj('Tag','lblEstado');
upos = [0, 45, Largura, 12];
set(u,'Position',upos);
% Reposiciona lblAlisa
u = findobj('Tag','lblAlisa');
upos = [10, Altura-30, 90, 20];
set(u,'Position',upos);
% Reposiciona mnuMetodo
u = findobj('Tag','mnuMetodo');
upos = [100, Altura-15, 120, 10];
set(u,'Position',upos);
% Reposiciona btnInserir
u = findobj('Tag','btnInserir');
upos = [Largura-540, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnRemover
u = findobj('Tag','btnRemover');
upos = [Largura-450, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnDesfazer
u = findobj('Tag','btnDesfazer');
upos = [Largura-360, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnOriginal
u = findobj('Tag','btnOriginal');
upos = [Largura-270, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnReiniciar
u = findobj('Tag','btnReiniciar');
upos = [Largura-180, Altura-25, 80, 20];
set(u,'Position',upos);
% Reposiciona btnEncerrar
u = findobj('Tag','btnEncerrar');
upos = [Largura-90, Altura-25, 80, 20];
set(u,'Position',upos);
% Finalização
set(fig,'Units',old_units);
end

% Cria vetor com dados originais da variável
% Insere -1 nas posições onde não existe tempo original correspondente
function [ aVar ] = copiados1(tVar, origX, origY)
aVar = zeros(length(tVar),1);
for i=1:length(tVar)
    p = find(origX==tVar(i)); % Verifica se tempo existe no original
    if (isempty(p))
        aVar(i) = -1;
    else
        aVar(i) = origY(p);
    end
end
end
end

```

6. GraficosFinais.m

```

% *****
% Módulo: GraficosFinais.m
% Descrição: Desenha gráficos após ajuste de pontos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Desenha Gráficos Finais
function GraficosFinais
% Variáveis Globais
VariaveisGlobais
% Exibe Gráficos
GraficoLnXv;
GraficoMi;
GraficoViab;
if (~Estah_Vazio(oProd))

```

```

    GraficoProd1;
end
if (~(Estah_Vazio(oProd)&&Estah_Vazio(oProde)))
    GraficoProd2;
end
if (~Estah_Vazio(oGLC))
    GraficoGLC;
end
if (~Estah_Vazio(oGLN))
    GraficoGLN;
end
if (~Estah_Vazio(oGLU))
    GraficoGLU;
end
if (~Estah_Vazio(oLAC))
    GraficoLAC;
end
if (~Estah_Vazio(oNH4))
    GraficoNH4;
end
if (~Estah_Vazio(oALA))
    GraficoALA;
end
end
end

function GraficoLnXv
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigLnXvF = figure('name', 'Tratamento de  $\mu$ ', 'NumberTitle','off',...
    'Toolbar', 'figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoLnXvF = axes('Parent',hFigLnXvF, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none'); % Transparente (nenhuma cor de fundo)
xlim(hEixoLnXvF, [ 0 (max(ot)+(max(x)-min(ot))./2.5) ] );
hold on;
% Recalcula prim_ponto ult_ponto
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
% Ajusta reta da fase exponencial
lnX = log(oXv); % ATENÇÃO: usar sempre pontos originais
% Regressão
ti = ot(prim_ponto:ult_ponto); % ATENÇÃO: usar sempre pontos originais
lnXi = lnX(prim_ponto:ult_ponto);
p = polyfit(ti,lnXi,1); % Regressão Linear (polinômio de grau 1)
% Calcula e Exibe coeficientes
umax = p(1);
lnXcalc = p(2);
R_umax = corrccoef(ti,lnXi);
R_umax = R_umax(1,2);
% Apaga eixo
cla(hEixoLnXvF)
% Plota LnXv
plot(hEixoLnXvF,ot,log(oXv),'ro');
% Desenha reta prolongada com valor calculado de lnX
tam = length(ot); % Quantidade de elemento do vetor ot
primeiro = ot(1); % Primeiro elemento do vetor ot
ultimo = ot(tam); % Ultimo elemento do vetor ot
ti = primeiro:0.1:ultimo;
lnXi = polyval(p,ti);
plot(hEixoLnXvF,ti,lnXi,'b');
% Legenda, título e rótulos
legend(hEixoLnXvF,'Experimental','Regressão','Location','SouthEast');
Tit = ['Tratamento de  $\mu$  (Ensaio: NomeEnsaio)'];
title(hEixoLnXvF,Tit, 'FontSize', 12);
xlabel(hEixoLnXvF,'Tempo (dias)');
ylabel(hEixoLnXvF,'ln(X)');
% Copyright
limiteX = get(hEixoLnXvF,'Xlim');
limiteY = get(hEixoLnXvF,'Ylim');
text('Position',[limiteX(2)*1.10 limiteY(1)],...
    'String','© 2009-2011 - Marilena Martins Pamboukian',...
    'FontName','Arial',...
    'FontSize',8,'Color','r','Rotation',90);
end

function GraficoMi
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigMi = figure('name','Curva  $\mu x$ ', 'NumberTitle','off',...
    'Toolbar', 'figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos
hEixoMil = axes('Parent',hFigMi, ...

```

```

        'Position',[.07 .10 .80 .80],...
        'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
hEixoMi2 = axes('Parent',hFigMi, ...
    'Position',get(hEixoMi1,'Position'),... % Cria Eixo 2 na mesma posição do Eixo 1
    'XAxisLocation','bottom',... % Eixo X embaixo
    'YAxisLocation','right',... % Eixo Y à direita
    'YColor','b',... % Eixo Y azul
    'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
hEixoMi3 = axes('Parent',hFigMi, ...
    'Position',get(hEixoMi1,'Position'),... % Cria Eixo 2 na mesma posição do Eixo 1
    'XAxisLocation','bottom',... % Eixo X embaixo
    'YAxisLocation','right',... % Eixo Y à direita
    'YColor','b',... % Eixo Y azul
    'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
% Desenha gráfico de Xv
% Copia valores
x = at; y = aXv; % Pontos originais + inseridos
xins = instXv; yins = insXv; % Iniciliza vetores de pontos inseridos
xf = mt; yf = mXv; % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoMi1;
hEixo2 = hEixoMi2;
Desenha2;
% Ajusta título e rótulos
Tit = [ 'Curva ux (Ensaio:' NomeEnsaio ') ' ];
title(hEixoMi1,Tit, 'FontSize', 12);
xlabel(hEixoMi1,'Tempo (dias)');
ylabel(hEixoMi1,'Células Viáveis (cel/mL)');
% Desenha gráfico de mi
% Recalcula prim_ponto ult_ponto
[prim_ponto ult_ponto] = PontosExp(mt,mXv);
% Calcula ux usando os pontos de Xv interpolados
xmi = zeros(length(spxi)-1,1);
ymi = zeros(length(spxi)-1,1);
xmig = [];
ymig = [];
for i=1:length(spt)-1
    xmi(i) = (spxi(i+1)+spxi(i))./2; % ponto médio
    ymi(i) = (log(spyi(i+1))-log(spyi(i)))./(spxi(i+1)-spxi(i)); % derivada
    if xmi(i) <= mt(ult_ponto+1) % grafico até 1 ponto após o exponencial
        xmig(i) = xmi(i);
        ymig(i) = ymi(i);
    end
end
% Plota u
plot(hEixoMi2,xmig,ymig,'b');
% Plota umax
plot(hEixoMi2,[mt(prim_ponto) mt(ult_ponto)],[umax umax],'m');
ylim(hEixoMi2,[ 0 max(ymi) ] );
legend(hEixoMi2,'ux','umax','Location','SouthEast');
ylabel(hEixoMi2,'Velocidade Específica (1/d)');
% Desenha linhas verticais
plot(hEixoMi3,[InicialExp InicialExp],[0 1],'Color','b','LineStyle',':');
plot(hEixoMi3,[FinalExp FinalExp],[0 1],'Color','r','LineStyle',':');
plot(hEixoMi3,[FinalCresc FinalCresc],[0 1],'Color',[ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoMi3,[InicialMorte InicialMorte],[0 1],'Color','m','LineStyle',':');
plot(hEixoMi3,[Prodmax Prodmax],[0 1],'Color',[ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoMi3,[FinalProde FinalProde],[0 1],'Color',[ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoMi3,'Inicio Exp','Final Exp','Final Cresc','Morte',...
    [ NomeProduto 'max'], [NomeProduto 'e,max'],'Location','East');
xlim(hEixoMi3,[ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoMi3,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoMi1);
end

function GraficoViab
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigViab = figure('name','Células Viáveis','NumberTitle','off',...
    'Toolbar','figure', ...
    'Color','w',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixos
hEixoViab1 = axes('Parent',hFigViab, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
hEixoViab2 = axes('Parent',hFigViab, ...
    'Position',get(hEixoViab1,'Position'),... % Cria Eixo2 na mesma posição do Eixo1
    'XAxisLocation','bottom',... % Eixo X embaixo
    'YAxisLocation','right',... % Eixo Y à direita
    'YColor','b',... % Eixo Y azul
    'Color','none', ... % Transparente (nenhuma cor de fundo)

```

```

        'YLim',[0 100],...                % Limites do Eixo Y
        'YTick',0:10:100);                % Divisões do Eixo Y

hold on;
hEixoViab3 = axes('Parent',hFigViab, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none');                    % Transparente (nenhuma cor de fundo)

hold on;
% Desenha gráfico de Xv
% Copia valores
x = at;          y = aXv;                % Pontos originais + inseridos
xins = instXv;   yins = insXv;           % Iniciliza vetores de pontos inseridos
xf = mt;         yf = mXv;               % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoViab1;
hEixo2 = hEixoViab2;
Desenha2;
% Ajusta título e rótulos
Tit = [ 'Células Viáveis (Ensaio:' NomeEnsaio ') ' ];
title(hEixoViab1,Tit, 'FontSize', 12);
xlabel(hEixoViab1,'Tempo (dias)');
ylabel(hEixoViab1,'Células Viáveis (cel/mL)');
% Desenha gráfico de viabilidade
plot(hEixoViab2,spt,spViab,'b');
ylabel(hEixoViab2,'Viabilidade (%)');
% Desenha linhas verticais
plot(hEixoViab3,[InicialExp InicialExp],[0 1],'Color','b','LineStyle',':');
plot(hEixoViab3,[FinalExp FinalExp],[0 1],'Color','r','LineStyle',':');
plot(hEixoViab3,[FinalCresc FinalCresc],[0 1],'Color',[ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoViab3,[InicialMorte InicialMorte],[0 1],'Color','m','LineStyle',':');
plot(hEixoViab3,[Prodmax Prodmax],[0 1],'Color',[ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoViab3,[FinalProde FinalProde],[0 1],'Color',[ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoViab3,'Início Exp','Final Exp','Final Cresc','Morte',...
    [ NomeProduto 'e,max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoViab3, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoViab3,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoViab1);
end

function GraficoProd1
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigProd = figure('name',NomeProduto,'NumberTitle','off',...
    'ToolBar','figure', ...
    'Color','w',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos
hEixoProd1 = axes('Parent',hFigProd, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none');                    % Transparente (nenhuma cor de fundo)

hold on;
hEixoProd2 = axes('Parent',hFigProd, ...
    'Position',[.07 .10 .80 .80],...
    'XAxisLocation','bottom',...        % Eixo X embaixo
    'YAxisLocation','right',...         % Eixo Y à direita
    'YColor','b',...                    % Eixo Y azul
    'Color','none');                    % Transparente (nenhuma cor de fundo)

% Desenha gráfico de Prod
% Copia valores
x = at;          y = aProd;                % Pontos originais + inseridos
xins = instProd; yins = insProd;           % Iniciliza vetores de pontos inseridos
xf = mt;         yf = mProd;               % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoProd1;
hEixo2 = hEixoProd2;
Desenha2;
% Ajusta título e rótulos
Tit = [ NomeProduto ' (Ensaio:' NomeEnsaio ') ' ];
title(hEixoProd1,Tit, 'FontSize', 12);
xlabel(hEixoProd1,'Tempo (dias)');
ylabel(hEixoProd1,[NomeProduto ' (µg/L)']);
% Desenha linhas verticais
plot(hEixoProd2,[InicialExp InicialExp],[0 1],'Color','b','LineStyle',':');
plot(hEixoProd2,[FinalExp FinalExp],[0 1],'Color','r','LineStyle',':');
plot(hEixoProd2,[FinalCresc FinalCresc],[0 1],'Color',[ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoProd2,[InicialMorte InicialMorte],[0 1],'Color','m','LineStyle',':');
plot(hEixoProd2,[Prodmax Prodmax],[0 1],'Color',[ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoProd2,[FinalProde FinalProde],[0 1],'Color',[ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoProd2,'Início Exp','Final Exp','Final Cresc','Morte',...
    [ NomeProduto 'e,max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoProd2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoProd2,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoProd1);
end

```



```

function GraficoProd2
    VariaveisGlobais
    % Cria figura centralizada na tela com tamanho 800x600
    screensize = get(0,'ScreenSize'); % Tamanho da tela
    hFigProde = figure('name', [ NomeProduto 'e - ' NomeProduto '/Xt' ],...
        'NumberTitle','off',...
        'Toolbar', 'figure', ...
        'Color','w',...
        'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

    % Cria eixos
    hEixoProde1 = axes('Parent',hFigProde, ...
        'Position',[.07 .10 .80 .80],...
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    hEixoProde2 = axes('Parent',hFigProde, ...
        'Position',get(hEixoProde1,'Position'),...% Cria Eixo2 na mesma posição do Eixo1
        'XAxisLocation','bottom',... % Eixo X embaixo
        'YAxisLocation','right',... % Eixo Y à direita
        'YColor','b',... % Eixo Y azul
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    hEixoProde3 = axes('Parent',hFigProde, ...
        'Position',[.07 .10 .80 .80],...
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    % Desenha gráfico de Prode
    if (~Estah_Vazio(oProde))
        % Copia valores
        x = at; y = aProde; % Pontos originais + inseridos
        xins = instProde; yins = insProde; % Inicializa vetores de pontos inseridos
        xf = mt; yf = mProde; % Inicializa Dados finais
        % Desenha gráfico
        hEixo1 = hEixoProde1;
        hEixo2 = hEixoProde2;
        Desenha2;
        Tit = [ NomeProduto 'e (Ensaio:' NomeEnsaio ')' ];
        title(hEixoProde1,Tit, 'FontSize', 12);
        ylabel(hEixoProde1,[ NomeProduto 'e (ng/1E7 cel)' ]);
        ylim(hEixoProde1,[ 0 (max(y)+(max(y)-min(y))./20) ] );
        ylim(hEixoProde2,[ 0 (max(y)+(max(y)-min(y))./20) ] );
    end
    % Ajusta título e rótulos
    xlabel(hEixoProde1,'Tempo (dias)');
    % Desenha gráfico de Prod/Xt
    if (~Estah_Vazio(oProd))
        % Calcula Prod/Xt usando os pontos de Prod e de Xt interpolados
        xm = spt;
        ym = spProd./spXt*1E7;
        % Plota Prod/Xt
        plot(hEixoProde2,xm,ym,'b-');
        Tit = [ NomeProduto '/Xt (Ensaio:' NomeEnsaio ')' ];
        title(hEixoProde1,Tit, 'FontSize', 12);
        ylabel(hEixoProde2,[ NomeProduto '/Xt (ng/1E7 cel)']);
        ylim(hEixoProde1,[ 0 (max(ym)+(max(ym)-min(ym))./20) ] );
        ylim(hEixoProde2,[ 0 (max(ym)+(max(ym)-min(ym))./20) ] );
        xlim(hEixoProde1,[ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
        xlim(hEixoProde2,[ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
    end
    if (~Estah_Vazio(oProd) && ~Estah_Vazio(oProde))
        Tit = [ NomeProduto 'e - ' NomeProduto '/Xt (Ensaio:' NomeEnsaio ')' ];
        title(hEixoProde1,Tit, 'FontSize', 12);
        yaxx = [y;ym]; % Junta y da Prode e de Prod/Xt para descobrir qual é o maior
        ylim(hEixoProde1,[ 0 (max(yaux)+(max(yaux)-min(yaux))./20) ] );
        ylim(hEixoProde2,[ 0 (max(yaux)+(max(yaux)-min(yaux))./20) ] );
    end
    % Desenha linhas verticais
    plot(hEixoProde3,[InicialExp InicialExp],[0 1],'Color','b','LineStyle',':');
    plot(hEixoProde3,[FinalExp FinalExp],[0 1],'Color','r','LineStyle',':');
    plot(hEixoProde3,[FinalCresc FinalCresc],[0 1],'Color',[ 0.0 0.7 0.0 ],'LineStyle',':');
    plot(hEixoProde3,[InicialMorte InicialMorte],[0 1],'Color','m','LineStyle',':');
    plot(hEixoProde3,[Prodmax Prodmax],[0 1],'Color',[ 1.0 0.5 0.0 ],'LineStyle',':');
    plot(hEixoProde3,[FinalProde FinalProde],[0 1],'Color',[ 0.5 0.0 0.5 ],'LineStyle',':');
    legend(hEixoProde3,'Inicio Exp','Final Exp','Final Cresc','Morte',...
        [ NomeProduto ',max'], [NomeProduto 'e,max'],'Location','SouthEast');
    xlim(hEixoProde3,[ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
    set(hEixoProde3,'Visible','off');
    % Retorna foco para Eixo1
    axes(hEixoProde1);
end

function GraficoGLC
    VariaveisGlobais
    % Cria figura centralizada na tela com tamanho 800x600
    screensize = get(0,'ScreenSize'); % Tamanho da tela
    hFigGLC = figure('name','GLC','NumberTitle','off',...
        'Toolbar', 'figure', ...
        'Color','w',...
        'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

```

```

% Cria eixos
hEixoGLC1 = axes('Parent',hFigGLC, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
hEixoGLC2 = axes('Parent',hFigGLC, ...
    'Position',[.07 .10 .80 .80],...
    'XAxisLocation','bottom',... % Eixo X embaixo
    'YAxisLocation','right',... % Eixo Y à direita
    'YColor','b',... % Eixo Y azul
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Desenha gráfico de GLC
% Copia valores
x = at; y = aGLC; % Pontos originais + inseridos
xins = instGLC; yins = insGLC; % Iniciliza vetores de pontos inseridos
xf = mt; yf = mGLC; % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoGLC1;
hEixo2 = hEixoGLC2;
Desenha2;
% Ajusta título e rótulos
Tit = [ 'GLC (Ensaio:' NomeEnsaio ') ' ];
title(hEixoGLC1,Tit, 'FontSize', 12);
xlabel(hEixoGLC1,'Tempo (dias)');
ylabel(hEixoGLC1,'GLC (g/L)');
% Desenha linhas verticais
plot(hEixoGLC2,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
plot(hEixoGLC2,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');
plot(hEixoGLC2,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoGLC2,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
plot(hEixoGLC2,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoGLC2,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoGLC2,'Inicio Exp','Final Exp','Final Cresc','Morte',...
    [ NomeProduto 'max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoGLC2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoGLC2,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoGLC1);
end

function GraficoGLN
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigGLN = figure('name','GLN','NumberTitle','off',...
    'ToolBar','figure', ...
    'Color','w',...
    'Position',[ (screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixos
hEixoGLN1 = axes('Parent',hFigGLN, ...
    'Position',[.07 .10 .80 .80],...
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
hEixoGLN2 = axes('Parent',hFigGLN, ...
    'Position',[.07 .10 .80 .80],...
    'XAxisLocation','bottom',... % Eixo X embaixo
    'YAxisLocation','right',... % Eixo Y à direita
    'YColor','b',... % Eixo Y azul
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Desenha gráfico de GLN
% Copia valores
x = at; y = aGLN; % Pontos originais + inseridos
xins = instGLN; yins = insGLN; % Iniciliza vetores de pontos inseridos
xf = mt; yf = mGLN; % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoGLN1;
hEixo2 = hEixoGLN2;
Desenha2;
% Ajusta título e rótulos
Tit = [ 'GLN (Ensaio:' NomeEnsaio ') ' ];
title(hEixoGLN1,Tit, 'FontSize', 12);
xlabel(hEixoGLN1,'Tempo (dias)');
ylabel(hEixoGLN1,'GLN (g/L)');
% Desenha linhas verticais
plot(hEixoGLN2,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
plot(hEixoGLN2,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');
plot(hEixoGLN2,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoGLN2,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
plot(hEixoGLN2,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoGLN2,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoGLN2,'Inicio Exp','Final Exp','Final Cresc','Morte',...
    [ NomeProduto 'max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoGLN2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoGLN2,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoGLN1);

```

```

end

function GraficoGLU
    VariaveisGlobais
    % Cria figura centralizada na tela com tamanho 800x600
    screensize = get(0,'ScreenSize'); % Tamanho da tela
    hFigGLU = figure('name','GLU','NumberTitle','off',...
        'Toolbar','figure', ...
        'Color','w',...
        'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

    % Cria eixos
    hEixoGLU1 = axes('Parent',hFigGLU, ...
        'Position',[.07 .10 .80 .80],...
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    hEixoGLU2 = axes('Parent',hFigGLU, ...
        'Position',[.07 .10 .80 .80],...
        'XAxisLocation','bottom',... % Eixo X embaixo
        'YAxisLocation','right',... % Eixo Y à direita
        'YColor','b',... % Eixo Y azul
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    % Desenha gráfico de GLU
    % Copia valores
    x = at; y = aGLU; % Pontos originais + inseridos
    xins = instGLU; yins = insGLU; % Inicializa vetores de pontos inseridos
    xf = mt; yf = mGLU; % Inicializa Dados finais
    % Desenha gráfico
    hEixo1 = hEixoGLU1;
    hEixo2 = hEixoGLU2;
    Desenha2;
    % Ajusta título e rótulos
    Tit = [ 'GLU (Ensaio:' NomeEnsaio ') ' ];
    title(hEixoGLU1,Tit, 'FontSize', 12);
    xlabel(hEixoGLU1,'Tempo (dias)');
    ylabel(hEixoGLU1,'GLU (g/L)');
    % Desenha linhas verticais
    plot(hEixoGLU2,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
    plot(hEixoGLU2,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');
    plot(hEixoGLU2,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
    plot(hEixoGLU2,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
    plot(hEixoGLU2,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
    plot(hEixoGLU2,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
    legend(hEixoGLU2,'Inicio Exp','Final Exp','Final Cresc','Morte',...
        [ NomeProduto 'e,max'], [NomeProduto 'e,max'],'Location','SouthEast');
    xlim(hEixoGLU2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
    set(hEixoGLU2,'Visible','off');
    % Retorna foco para Eixo1
    axes(hEixoGLU1);
end

function GraficoLAC
    VariaveisGlobais
    % Cria figura centralizada na tela com tamanho 800x600
    screensize = get(0,'ScreenSize'); % Tamanho da tela
    hFigLAC = figure('name','LAC','NumberTitle','off',...
        'Toolbar','figure', ...
        'Color','w',...
        'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

    % Cria eixos
    hEixoLAC1 = axes('Parent',hFigLAC, ...
        'Position',[.07 .10 .80 .80],...
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    hEixoLAC2 = axes('Parent',hFigLAC, ...
        'Position',[.07 .10 .80 .80],...
        'XAxisLocation','bottom',... % Eixo X embaixo
        'YAxisLocation','right',... % Eixo Y à direita
        'YColor','b',... % Eixo Y azul
        'Color','none'); % Transparente (nenhuma cor de fundo)

    hold on;
    % Desenha gráfico de LAC
    % Copia valores
    x = at; y = aLAC; % Pontos originais + inseridos
    xins = instLAC; yins = insLAC; % Inicializa vetores de pontos inseridos
    xf = mt; yf = mLAC; % Inicializa Dados finais
    % Desenha gráfico
    hEixo1 = hEixoLAC1;
    hEixo2 = hEixoLAC2;
    Desenha2;
    % Ajusta título e rótulos
    Tit = [ 'LAC (Ensaio:' NomeEnsaio ') ' ];
    title(hEixoLAC1,Tit, 'FontSize', 12);
    xlabel(hEixoLAC1,'Tempo (dias)');
    ylabel(hEixoLAC1,'LAC (g/L)');
    % Desenha linhas verticais
    plot(hEixoLAC2,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
    plot(hEixoLAC2,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');

```

```

plot(hEixoLAC2,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoLAC2,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
plot(hEixoLAC2,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoLAC2,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoLAC2,'Inicio Exp','Final Exp','Final Cresc','Morte',...
[ NomeProduto 'max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoLAC2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoLAC2,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoLAC1);
end

function GraficoNH4
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigNH4 = figure('name','NH4','NumberTitle','off',...
'Toolbar','figure',...
'Color','w',...
'Position',[ (screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos
hEixoNH41 = axes('Parent',hFigNH4,...
'Position',[.07 .10 .80 .80],...
'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
hEixoNH42 = axes('Parent',hFigNH4,...
'Position',[.07 .10 .80 .80],...
'XAxisLocation','bottom',... % Eixo X embaixo
'YAxisLocation','right',... % Eixo Y à direita
'YColor','b',... % Eixo Y azul
'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Desenha gráfico de NH4
% Copia valores
x = at; y = aNH4; % Pontos originais + inseridos
xins = instNH4; yins = insNH4; % Iniciliza vetores de pontos inseridos
xf = mt; yf = mNH4; % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoNH41;
hEixo2 = hEixoNH42;
Desenha2;
% Ajusta título e rótulos
Tit = [ 'NH4 (Ensaio: NomeEnsaio )' ];
title(hEixoNH41,Tit, 'FontSize', 12);
xlabel(hEixoNH41,'Tempo (dias)');
ylabel(hEixoNH41,'NH4 (g/L)');
% Desenha linhas verticais
plot(hEixoNH42,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
plot(hEixoNH42,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');
plot(hEixoNH42,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoNH42,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
plot(hEixoNH42,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoNH42,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoNH42,'Inicio Exp','Final Exp','Final Cresc','Morte',...
[ NomeProduto 'max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoNH42, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoNH42,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoNH41);
end

function GraficoALA
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFigALA = figure('name','ALA','NumberTitle','off',...
'Toolbar','figure',...
'Color','w',...
'Position',[ (screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixos
hEixoALA1 = axes('Parent',hFigALA,...
'Position',[.07 .10 .80 .80],...
'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
hEixoALA2 = axes('Parent',hFigALA,...
'Position',[.07 .10 .80 .80],...
'XAxisLocation','bottom',... % Eixo X embaixo
'YAxisLocation','right',... % Eixo Y à direita
'YColor','b',... % Eixo Y azul
'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Desenha gráfico de ALA
% Copia valores
x = at; y = aALA; % Pontos originais + inseridos
xins = instALA; yins = insALA; % Iniciliza vetores de pontos inseridos
xf = mt; yf = mALA; % Inicializa Dados finais
% Desenha gráfico
hEixo1 = hEixoALA1;

```

```

hEixo2 = hEixoALA2;
Desenha2;
% Ajusta titulo e rótulos
Tit = [ 'ALA (Ensaio:' NomeEnsaio ' )' ];
title(hEixoALA1,Tit, 'FontSize', 12);
xlabel(hEixoALA1,'Tempo (dias)');
ylabel(hEixoALA1,'ALA (g/L)');
% Desenha linhas verticais
plot(hEixoALA2,[InicialExp InicialExp],[0 1],'Color', 'b','LineStyle',':');
plot(hEixoALA2,[FinalExp FinalExp],[0 1],'Color', 'r','LineStyle',':');
plot(hEixoALA2,[FinalCresc FinalCresc],[0 1],'Color', [ 0.0 0.7 0.0 ],'LineStyle',':');
plot(hEixoALA2,[InicialMorte InicialMorte],[0 1],'Color', 'm','LineStyle',':');
plot(hEixoALA2,[Prodmax Prodmax],[0 1],'Color', [ 1.0 0.5 0.0 ],'LineStyle',':');
plot(hEixoALA2,[FinalProde FinalProde],[0 1],'Color', [ 0.5 0.0 0.5 ],'LineStyle',':');
legend(hEixoALA2,'Inicio Exp','Final Exp','Final Cresc','Morte',...
        [ NomeProduto ' ,max'], [NomeProduto 'e,max'],'Location','SouthEast');
xlim(hEixoALA2, [ 0 (max(ot)+(max(ot)-min(ot))./2.5) ] );
set(hEixoALA2,'Visible','off');
% Retorna foco para Eixo1
axes(hEixoALA1);
end

function Desenha2
VariaveisGlobais
% Verifica pontos que foram modificados
cont_mod = 1;
xmod = []; % criar xmod e ymod, mesmo que nenhum ponto tenha sido
ymod = []; % modificado, para evitar problema na plotagem
for i=1:length(x)
    if (y(i)~=yf(i) && yf(i)~=realmin) % ponto i foi modificado
        xmod(cont_mod) = xf(i); %ok<AGROW>
        ymod(cont_mod) = yf(i); %ok<AGROW>
        cont_mod = cont_mod + 1;
    end
end
% Verifica pontos que foram removidos
cont_rem = 1;
cont_inter = 1;
xrem = [];
yrem = [];
xinter = [];
yinter = [];
for i=1:length(x)
    if (yf(i)==realmin) % ponto i foi removido
        xrem(cont_rem) = x(i); %ok<AGROW>
        yrem(cont_rem) = y(i); %ok<AGROW>
        cont_rem = cont_rem + 1;
    else
        xinter(cont_inter) = xf(i); %ok<AGROW>
        yinter(cont_inter) = yf(i); %ok<AGROW>
        cont_inter = cont_inter + 1;
    end
end
% Faz interpolação da variável atual
[ xi yi ] = Ajusta_Curva(xinter,yinter,TipoAjuste);
% Remove pontos inexistentes (-1)
[ xxi yyi ] = Remove_Pontos(xi,yi);
[ xx yy ] = Remove_Pontos(x,y);
% Plota dados originais, interpolados, modificados, removidos e inseridos
plot(hEixo1,xxi,yyi,'k-');
plot(hEixo1,xx,yy,'or');
plot(hEixo1,xmod,ymod,'sk');
plot(hEixo1,xins,yins,'g*');
plot(hEixo1,xrem,yrem,'xb');
% Ajusta legendas
cont_ins = length(xins)+1;
if (cont_mod==1 && cont_rem==1 && cont_ins==1)
    legend(hEixo1,'Alisado','Original','Location','NorthEast');
elseif (cont_mod~=1 && cont_rem==1 && cont_ins==1)
    legend(hEixo1,'Alisado','Original','Modificado','Location','NorthEast');
elseif (cont_mod==1 && cont_rem~=1 && cont_ins==1)
    legend(hEixo1,'Alisado','Original','Removido','Location','NorthEast');
elseif (cont_mod==1 && cont_rem==1 && cont_ins~=1)
    legend(hEixo1,'Alisado','Original','Inserido','Location','NorthEast');
elseif (cont_mod~=1 && cont_rem~=1 && cont_ins==1)
    legend(hEixo1,'Alisado','Original','Modificado','Removido','Location','NorthEast');
elseif (cont_mod~=1 && cont_rem==1 && cont_ins~=1)
    legend(hEixo1,'Alisado','Original','Modificado','Inserido','Location','NorthEast');
elseif (cont_mod==1 && cont_rem==1 && cont_ins==1)
    legend(hEixo1,'Alisado','Original','Inserido','Removido','Location','NorthEast');
else
    legend(hEixo1,'Alisado','Original','Modificado','Inserido','Removido','Location','NorthEast');
end
% Ajusta limites do Eixo1
xlim(hEixo1, [ 0 (max(x)+(max(x)-min(x))./2.5) ] );
ylim(hEixo1, [ 0 (max(y)+(max(y)-min(y))./20) ] );
% Ajusta limites do Eixo 2
xlim(hEixo2, [ 0 (max(x)+(max(x)-min(x))./2.5) ] );

```

```
% Copyright
limiteX = get(hEixo1,'Xlim');
limiteY = get(hEixo1,'Ylim');
text('Position',[limiteX(2)*1.10 limiteY(1)],...
     'String','© 2009-2011 - Marilena Martins Pamboukian',...
     'FontName','Arial',...
     'FontSize',8,'Color','r','Rotation',90);
end
```

7. GraficosFinaisY.m

```
% *****
% Módulo: GraficosFinaisY.m
% Descrição: Desenha gráficos após conversões
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Desenha Gráficos Finais de Conversão
function GraficosFinaisY
% Variáveis Globais
VariaveisGlobais
% Exibe Gráficos
if (~Estah_Vazio(oGLC))
    Grafico1;
end
if (~Estah_Vazio(oGLN))
    Grafico2;
end
if (~Estah_Vazio(oLAC))
    Grafico3;
end
if (~Estah_Vazio(oLAC) && ~Estah_Vazio(oGLC))
    Grafico4;
end
if (~Estah_Vazio(oNH4))
    Grafico5;
end
if (~Estah_Vazio(oNH4) && ~Estah_Vazio(oGLN))
    Grafico6;
end
if (~Estah_Vazio(oProd) && ~Estah_Vazio(oGLC))
    Grafico7;
end
if (~Estah_Vazio(oALA) && ~Estah_Vazio(oGLC))
    Grafico8;
end
if (~Estah_Vazio(oALA) && ~Estah_Vazio(oGLN))
    Grafico9;
end
end

function Grafico1
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig1 = figure('name','Yx/GLC','NumberTitle','off',...
               'ResizeFcn',@hFigResize, ...
               'ToolBar', 'figure', ...
               'Color','w',...
               'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoY1 = axes('Parent',hFig1, ...
               'Position',[.07 .15 .80 .75],...
               'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY1 = uicontrol(hFig1,'Style','text','String','Y=0',...
                  'BackgroundColor','w','ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold','Tag','lblY1');
lblR1 = uicontrol(hFig1,'Style','text','String','R=0',...
                  'BackgroundColor','w','ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold','Tag','lblR1');
```

```

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLC(1:ult_ponto);
y = oXv(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLC); xf = xf(1:ult_ponto);
yf = copiados2(mXv); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYx_GLC;
% Desenha gráfico
hEixo1 = hEixoY1;
Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = 'Células Viáveis (cel/mL)';
Desenha2;
% Ajusta título
title(hEixoY1,['Yx/GLC (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY1,'String',['Yx/GLC = ' num2str(resp*1000,'%10.4e') ' (cel/g)']);
set(lblR1,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico2
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig2 = figure('name','Yx/GLN','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixo
hEixoY2 = axes('Parent',hFig2, ...
    'Position',[.07 .15 .80 .75],...
    'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
% Cria rótulos para Y e R
lblY2 = uicontrol(hFig2,'Style','text','String','Y=0',...
    'BackgroundColor','w','ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold','Tag','lblY');
lblR2 = uicontrol(hFig2,'Style','text','String','R=0',...
    'BackgroundColor','w','ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold','Tag','lblR');
% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLN(1:ult_ponto);
y = oXv(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLN); xf = xf(1:ult_ponto);
yf = copiados2(mXv); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYx_GLN;
% Desenha gráfico
hEixo1 = hEixoY2;
Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = 'Células Viáveis (cel/mL)';
Desenha2;
% Ajusta título
title(hEixoY2,['Yx/GLN (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY2,'String',['Yx/GLN = ' num2str(resp*1000,'%10.4e') ' (cel/g)']);
set(lblR2,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico3
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig3 = figure('name','YLAC/x','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Position',[(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);
% Cria eixo
hEixoY3 = axes('Parent',hFig3, ...
    'Position',[.07 .15 .80 .75],...
    'Color','none'); % Transparente (nenhuma cor de fundo)
hold on;
% Cria rótulos para Y e R
lblY3 = uicontrol(hFig3,'Style','text','String','Y=0',...
    'BackgroundColor','w','ForegroundColor','k',...

```

```

        'HorizontalAlignment','left', ...
        'FontSize', 8, 'FontName','Arial', ...
        'FontWeight','bold', 'Tag', 'lblY');
lblR3 = uicontrol(hFig3,'Style','text', 'String', 'R=0',...
        'BackgroundColor','w', 'ForegroundColor','k',...
        'HorizontalAlignment','left', ...
        'FontSize', 8, 'FontName','Arial', ...
        'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oXv(1:ult_ponto);
y = oLAC(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mXv);    xf = xf(1:ult_ponto);
yf = copiados2(mLAC);   yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYLAC_x;
% Desenha gráfico
hEixo1 = hEixoY3;
Legenda = 'NorthEast'; RotuloX = 'Células Viáveis (cel/mL)'; RotuloY = 'LAC (g/L)';
Desenha2;
% Ajusta título
title(hEixoY3,['YLAC/x (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY3,'String',['YLAC/x = ' num2str(resp/1000,'%10.4e') ' (g/cel)']);
set(lblR3,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico4
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig4 = figure('name','YLAC/GLC','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoY4 = axes('Parent',hFig4, ...
    'Position',[.07 .15 .80 .75],...
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY4 = uicontrol(hFig4,'Style','text', 'String', 'Y=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblY');
lblR4 = uicontrol(hFig4,'Style','text', 'String', 'R=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLC(1:ult_ponto);
y = oLAC(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLC);   xf = xf(1:ult_ponto);
yf = copiados2(mLAC);   yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYLAC_GLC;
% Desenha gráfico
hEixo1 = hEixoY4;
Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = 'LAC (g/L)';
Desenha2;
% Ajusta título
title(hEixoY4,['YLAC/GLC (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY4,'String',['YLAC/GLC = ' num2str(resp,'%10.4e') ' (g/g)']);
set(lblR4,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico5
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig5 = figure('name','YNH4/x','NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar','figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

```



```

% Cria eixo
hEixoY5 = axes('Parent',hFig5, ...
               'Position',[.07 .15 .80 .75],...
               'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY5 = uicontrol(hFig5,'Style','text', 'String', 'Y=0',...
                  'BackgroundColor','w', 'ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold', 'Tag', 'lblY');
lblR5 = uicontrol(hFig5,'Style','text', 'String', 'R=0',...
                  'BackgroundColor','w', 'ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oXv(1:ult_ponto);
y = oNH4(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mXv); xf = xf(1:ult_ponto);
yf = copiados2(mNH4); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYNH4_x;
% Desenha gráfico
hEixo1 = hEixoY5;
Legenda = 'NorthEast'; RotuloX = 'Células Viáveis (cel/mL)'; RotuloY = 'NH4 (g/L)';
Desenha2;
% Ajusta título
title(hEixoY5,['YNH4/x (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY5,'String',['YNH4/x = ' num2str(resp/1000,'%10.4e') ' (g/cel)']);
set(lblR5,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico6
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig6 = figure('name','YNH4/GLN','NumberTitle','off',...
               'ResizeFcn',@hFigResize, ...
               'ToolBar','figure', ...
               'Color','w',...
               'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoY6 = axes('Parent',hFig6, ...
               'Position',[.07 .15 .80 .75],...
               'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY6 = uicontrol(hFig6,'Style','text', 'String', 'Y=0',...
                  'BackgroundColor','w', 'ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold', 'Tag', 'lblY');
lblR6 = uicontrol(hFig6,'Style','text', 'String', 'R=0',...
                  'BackgroundColor','w', 'ForegroundColor','k',...
                  'HorizontalAlignment','left', ...
                  'FontSize', 8, 'FontName','Arial', ...
                  'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLN(1:ult_ponto);
y = oNH4(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLN); xf = xf(1:ult_ponto);
yf = copiados2(mNH4); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYNH4_GLN;
% Desenha gráfico
hEixo1 = hEixoY6;
Legenda = 'NorthEast'; RotuloX = 'GLN (g/L)'; RotuloY = 'NH4 (g/L)';
Desenha2;
% Ajusta título
title(hEixoY6,['YNH4/GLN (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY6,'String',['YNH4/GLN = ' num2str(resp,'%10.4e') ' (g/g)']);
set(lblR6,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico7

```

```

VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig7 = figure('name',['Y' NomeProduto '/GLC'],'NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar', 'figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoY7 = axes('Parent',hFig7, ...
    'Position',[.07 .15 .80 .75],...
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY7 = uicontrol(hFig7,'Style','text', 'String', 'Y=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblY');
lblR7 = uicontrol(hFig7,'Style','text', 'String', 'R=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLC(1:ult_ponto);
y = oProd(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLC); xf = xf(1:ult_ponto);
yf = copiados2(mProd); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYProd_GLC;
% Desenha gráfico
hEixo1 = hEixoY7;
Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ NomeProduto ' (µg/L)' ];
Desenha2;
% Ajusta titulo
title(hEixoY7,['Y' NomeProduto '/GLC (Ensaio:' NomeEnsaio ')'], 'FontSize', 12);
% Ajusta rótulos
set(lblY7,'String',['Y' NomeProduto '/GLC ' num2str(resp,'%10.4e') ' (µg/g)']);
set(lblR7,'String',['R = ' num2str(R_resp,4)]);
end

function Grafico8
VariaveisGlobais
% Cria figura centralizada na tela com tamanho 800x600
screensize = get(0,'ScreenSize'); % Tamanho da tela
hFig8 = figure('name',['YALA/GLC'],'NumberTitle','off',...
    'ResizeFcn',@hFigResize, ...
    'Toolbar', 'figure', ...
    'Color','w',...
    'Position', [(screensize(3)-800)./2 (screensize(4)-600)./2 800 600]);

% Cria eixo
hEixoY8 = axes('Parent',hFig8, ...
    'Position',[.07 .15 .80 .75],...
    'Color','none'); % Transparente (nenhuma cor de fundo)

hold on;
% Cria rótulos para Y e R
lblY8 = uicontrol(hFig8,'Style','text', 'String', 'Y=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblY');
lblR8 = uicontrol(hFig8,'Style','text', 'String', 'R=0',...
    'BackgroundColor','w', 'ForegroundColor','k',...
    'HorizontalAlignment','left', ...
    'FontSize', 8, 'FontName','Arial', ...
    'FontWeight','bold', 'Tag', 'lblR');

% Dados originais
[prim_ponto ult_ponto] = PontosExp(ot,oXv);
x = oGLC(1:ult_ponto);
y = oALA(1:ult_ponto);
[ x y ] = Remove_Pontos(x,y);
% Dados finais
xf = copiados2(mGLC); xf = xf(1:ult_ponto);
yf = copiados2(mALA); yf = yf(1:ult_ponto);
[ xf yf ] = Remove_Pontos(xf,yf);
[ xf yf ] = AjustaModificados2(xf, yf);
% Pontos usados
ptUsado = uYALA_GLC;
% Desenha gráfico
hEixo1 = hEixoY8;
Legenda = 'NorthEast'; RotuloX = 'GLC (g/L)'; RotuloY = [ 'ALA (g/L)' ];
Desenha2;

```

```

        cont = cont + 1;
    end
end
end
% Verifica se existem pontos suficientes para ajustar reta (pelo menos dois)
if length(xinter) < 2
    resp = 0;
    R_resp = 0;
    cla(hEixo1);
    % Desenha Rótulos
    xlabel(RotuloX, 'Parent', hEixo1);
    ylabel(RotuloY, 'Parent', hEixo1);
    % Exibe mensagem
    limiteX = get(hEixo1, 'Xlim');
    limiteY = get(hEixo1, 'Ylim');
    text('Position', [limiteX(1)+(limiteX(2)-limiteX(1))*0.2 (limiteY(1)+limiteY(2))/2], ...
        'String', 'Pontos insuficientes para efetuar ajuste!', ...
        'FontName', 'Arial', ...
        'FontSize', 14, 'Color', 'r');
    return
end
% Faz Interpolação
xi = primeiro:passo_interp:ultimo;
warning off
p = polyfit(xinter, yinter, 1); % polinômio de grau 1
warning on
yi = polyval(p, xi);
% Respostas
resp = abs(p(1));
R_resp = corrcoef(xinter, yinter);
R_resp = R_resp(1, 2);
% Verifica pontos modificados que não foram removidos
cont = 1;
xmod = [];
ymod = [];
for i=1:length(yf)
    if (xf(i)~=realmin && yf(i)~=realmin) % ponto modificado i não foi removido
        xmod(cont) = xf(i); %ok<AGROW>
        ymod(cont) = yf(i); %ok<AGROW>
        cont = cont + 1;
    end
end
xmod = xmod';
ymod = ymod';
% Limpa eixos
cla(hEixo1);
% Plota dados originais, modificados, usados e interpolados
if (cont==1)
    plot(hEixo1, x, y, 'or', xinter, yinter, 'bs', xi, yi, 'k-');
else
    plot(hEixo1, x, y, 'or', xmod, ymod, 'sk', xinter, yinter, 'g.', xi, yi, 'k-');
end
% Desenha Rótulos e legenda
xlabel(RotuloX, 'Parent', hEixo1);
ylabel(RotuloY, 'Parent', hEixo1);
% Ajusta legendas
if (cont==1)
    legend(hEixo1, 'Original', 'Usado', 'Regressão', 'Location', Legenda);
else
    legend(hEixo1, 'Original', 'Modificado', 'Usado', 'Regressão', 'Location', Legenda);
end
% Ajusta limites do Eixo1
xlim(hEixo1, [ min([x;xmod]) (max([x;xmod])+(max([x;xmod])-min([x;xmod]))./3) ] );
% Copyright
limiteX = get(hEixo1, 'Xlim');
limiteY = get(hEixo1, 'Ylim');
text('Position', [limiteX(1)+(limiteX(2)-limiteX(1))*1.10 limiteY(1)], ...
    'String', '© 2009-2011 - Marilena Martins Pamboukian', ...
    'FontName', 'Arial', ...
    'FontSize', 8, 'Color', 'r', 'Rotation', 90);
end

function hFigResize(source, event)
    VariaveisGlobais
    % Inicialização (copiada do doc)
    fig = gcb0;
    old_units = get(fig, 'Units');
    set(fig, 'Units', 'pixels');
    figpos = get(fig, 'Position');
    % Dimensões da área útil
    Largura = figpos(3);
    Altura = figpos(4);
    % Reposiciona lblY
    u = findobj('Tag', 'lblY');
    upos = [Largura*0.68, Altura/2, 200, 15];
    set(u, 'Position', upos);
    % Reposiciona lblR
    u = findobj('Tag', 'lblR');

```

```

        cont = cont + 1;
    end
end
end
% Verifica se existem pontos suficientes para ajustar reta (pelo menos dois)
if length(xinter) < 2
    resp = 0;
    R_resp = 0;
    cla(hEixo1);
    % Desenha Rótulos
    xlabel(RotuloX, 'Parent', hEixo1);
    ylabel(RotuloY, 'Parent', hEixo1);
    % Exibe mensagem
    limiteX = get(hEixo1, 'Xlim');
    limiteY = get(hEixo1, 'Ylim');
    text('Position', [limiteX(1)+(limiteX(2)-limiteX(1))*0.2 (limiteY(1)+limiteY(2))/2], ...
        'String', 'Pontos insuficientes para efetuar ajuste!', ...
        'FontName', 'Arial', ...
        'FontSize', 14, 'Color', 'r');
    return
end
% Faz Interpolação
xi = primeiro:passo_interp:ultimo;
warning off
p = polyfit(xinter, yinter, 1);          % polinômio de grau 1
warning on
yi = polyval(p, xi);
% Respostas
resp = abs(p(1));
R_resp = corrcoef(xinter, yinter);
R_resp = R_resp(1, 2);
% Verifica pontos modificados que não foram removidos
cont = 1;
xmod = [];
ymod = [];
for i=1:length(yf)
    if (xf(i)~=realmin && yf(i)~=realmin)    % ponto modificado i não foi removido
        xmod(cont) = xf(i);    %#ok<AGROW>
        ymod(cont) = yf(i);    %#ok<AGROW>
        cont = cont + 1;
    end
end
xmod = xmod';
ymod = ymod';
% Limpa eixos
cla(hEixo1);
% Plota dados originais, modificados, usados e interpolados
if (cont==1)
    plot(hEixo1, x, y, 'or', xinter, yinter, 'bs', xi, yi, 'k-');
else
    plot(hEixo1, x, y, 'or', xmod, ymod, 'sk', xinter, yinter, 'g.', xi, yi, 'k-');
end
% Desenha Rótulos e legenda
xlabel(RotuloX, 'Parent', hEixo1);
ylabel(RotuloY, 'Parent', hEixo1);
% Ajusta legendas
if (cont==1)
    legend(hEixo1, 'Original', 'Usado', 'Regressão', 'Location', Legenda);
else
    legend(hEixo1, 'Original', 'Modificado', 'Usado', 'Regressão', 'Location', Legenda);
end
% Ajusta limites do Eixo1
xlim(hEixo1, [ min([x;xmod]) (max([x;xmod])+(max([x;xmod])-min([x;xmod]))./3) ] );
% Copyright
limiteX = get(hEixo1, 'Xlim');
limiteY = get(hEixo1, 'Ylim');
text('Position', [limiteX(1)+(limiteX(2)-limiteX(1))*1.10 limiteY(1)], ...
    'String', '© 2009-2011 - Marilena Martins Pamboukian', ...
    'FontName', 'Arial', ...
    'FontSize', 8, 'Color', 'r', 'Rotation', 90);
end

function hFigResize(source, event)
    VariaveisGlobais
    % Inicialização (copiada do doc)
    fig = gcb0;
    old_units = get(fig, 'Units');
    set(fig, 'Units', 'pixels');
    figpos = get(fig, 'Position');
    % Dimensões da área útil
    Largura = figpos(3);
    Altura = figpos(4);
    % Reposiciona lblY
    u = findobj('Tag', 'lblY');
    upos = [Largura*0.68, Altura/2, 200, 15];
    set(u, 'Position', upos);
    % Reposiciona lblR
    u = findobj('Tag', 'lblR');

```

```

    upos = [Largura*0.68, Altura/2-20, 110, 15];
    set(u,'Position',upos);
    % Finalização (copiada do doc)
    set(fig,'Units',old_units);
end

% Cria vetor com dados modificados (sem os inseridos)
% Não copia valores onde não existe tempo original correspondente
function [ mVar ] = copiados2(origY)
    VariaveisGlobais
    mVar = zeros(length(ot),1);
    c = 1;
    for i=1:length(mt)
        p = find(ot==mt(i)); % Verifica se tempo existe no original
        if (~isempty(p))
            mVar(c) = origY(i);
            c = c + 1;
        end
    end
end
end

```

8. GravaDados.m

```

% *****
% Módulo: GravaDados.m
% Descrição: Grava dados em arquivo xls
% Autor: Marilena Martins Pamboukian
% Versão: 2.7
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

%Grava dados em arquivo xls
function GravaDados(arquivo)

% Variáveis Globais
VariaveisGlobais
% Prepara Dados Alisados
disp('Aguarde! Gravando Dados Alisados...');
Tit1 = [ NomeProduto 'e (ng/1E7 cel)' ];
Tit2 = [ NomeProduto ' (µg/L)' ];
titulos = {'Tempo (d)', 'Xv (cel/mL)', 'Xd (cel/L)', 'Xt (cel/L)', 'Viab (%)', 'GLC (g/L)', ...
            'GLN (g/L)', 'GLU (g/L)', 'LAC (g/L)', 'NH4 (g/L)', 'ALA (g/L)', Tit1, Tit2 };
dados = [ spt spXv spXd spXt spViab spGLC spGLN spGLU spLAC spNH4 spALA spProde spProd ];
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados Alisados no arquivo
warning off;
xlswrite(arquivo,d,'Dados Alisados','A1')
pause( 0.1 );
warning on;

% Prepara Dados Ajuste
disp('Aguarde! Gravando Dados Ajuste...');
% Calcula dados ajuste
i=1;
p=1;
while (i<length(spt))
    spt1(p) = spt(i);
    spXv1(p) = spXv(i);
    spXd1(p) = spXd(i);
    spXt1(p) = spXt(i);
    spViab1(p) = spViab(i);
    spGLC1(p) = spGLC(i);
    spGLN1(p) = spGLN(i);
    spGLU1(p) = spGLU(i);
    spLAC1(p) = spLAC(i);
    spNH41(p) = spNH4(i);
    spALA1(p) = spALA(i);
    spProde1(p) = spProde(i);
    spProd1(p) = spProd(i);
    p = p + 1;
    i = i + 20; % Pega 1 ponto a cada 20 pontos
end
spt1(p) = spt(length(spt));
spXv1(p) = spXv(length(spXv));
spXd1(p) = spXd(length(spXd));
spXt1(p) = spXt(length(spXt));

```

```

spViab1(p) = spViab(length(spViab));
spGLC1(p) = spGLC(length(spGLC));
spGLN1(p) = spGLN(length(spGLN));
spGLU1(p) = spGLU(length(spGLU));
spLAC1(p) = spLAC(length(spLAC));
spNH41(p) = spNH4(length(spNH4));
spALA1(p) = spALA(length(spALA));
spProde1(p) = spProde(length(spProde));
spProd1(p) = spProd(length(spProd));
spt1 = spt1';
spXv1 = spXv1';
spXd1 = spXd1';
spXt1 = spXt1';
spViab1 = spViab1';
spGLC1 = spGLC1';
spGLN1 = spGLN1';
spGLU1 = spGLU1';
spLAC1 = spLAC1';
spNH41 = spNH41';
spALA1 = spALA1';
spProde1 = spProde1';
spProd1 = spProd1';
Tit1 = [ NomeProduto 'e (ng/1E7 cel)' ];
Tit2 = [ NomeProduto ' (µg/mL)' ];
titulos = {'Tempo (h)', 'Xv (1E7cel/mL)', 'Xd (1E7cel/mL)', 'Xt (1E7cel/mL)', 'Viab (%)', 'GLC (g/mL)', ...
           'GLN (g/mL)', 'GLU (g/mL)', 'LAC (g/mL)', 'NH4 (mg/mL)', 'ALA (g/mL)', Tit1, Tit2 };

spt1 = spt1 * 24;
for ii = 1:length(spt1)
    if (spXv1(ii) ~= -1)
        spXv1(ii) = spXv1(ii) / 1E7;
    end
    if (spXd1(ii) ~= -1)
        spXd1(ii) = spXd1(ii) / 1E7;
    end
    if (spXt1(ii) ~= -1)
        spXt1(ii) = spXt1(ii) / 1E7;
    end
    if (spGLC1(ii) ~= -1)
        spGLC1(ii) = spGLC1(ii) / 1000;
    end
    if (spGLN1(ii) ~= -1)
        spGLN1(ii) = spGLN1(ii) / 1000;
    end
    if (spGLU1(ii) ~= -1)
        spGLU1(ii) = spGLU1(ii) / 1000;
    end
    if (spLAC1(ii) ~= -1)
        spLAC1(ii) = spLAC1(ii) / 1000;
    end
    if (spALA1(ii) ~= -1)
        spALA1(ii) = spALA1(ii) / 1000;
    end
    if (spProd1(ii) ~= -1)
        spProd1(ii) = spProd1(ii) / 1000;
    end
end
dados = [ spt1 spXv1 spXd1 spXt1 spViab1 spGLC1 spGLN1 ...
          spGLU1 spLAC1 spNH41 spALA1 spProde1 spProd1 ];
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados Ajuste no arquivo
warning off;
xlswrite(arquivo,d,'Dados Ajuste','A1')
pause( 0.1 );
warning on;

% Calculando e gravando µmax
disp('Aguarde! Gravando µmax...');

% Calcula µmax
mi = umax;
R_mi = R_umax;
% Prepara valor de µmax
titulos = {'µmax (1/dia)', 'µmax (1/h)', 'R = ' };
dados = [ mi mi/24 R_mi ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de µmax no arquivo
warning off;
xlswrite(arquivo,d,'µ Máximo','A1')
pause( 0.1 );
warning on;

% Calcula µGLC
[mi R_mi] = CalculaMi(spGLC);
% Prepara valor de µ
titulos = {'µGLC (g/cel*dia)', 'µGLC (g/cel*h)', 'R = ' };

```

```

dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A5')
pause( 0.1 );
warning on;

% Calcula  $\mu_{GLN}$ 
[mi R_mi] = CalculaMi(spGLN);
% Prepara valor de  $\mu$ 
titulos = {' $\mu_{GLN}$  (g/cel*dia)'; ' $\mu_{GLN}$  (g/cel*h)'; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A9')
pause( 0.1 );
warning on;

% Calcula  $\mu_{GLU}$ 
[mi R_mi] = CalculaMi(spGLU);
% Prepara valor de  $\mu$ 
titulos = {' $\mu_{GLU}$  (g/cel*dia)'; ' $\mu_{GLU}$  (g/cel*h)'; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A13')
pause( 0.1 );
warning on;

% Calcula  $\mu_{LAC}$ 
[mi R_mi] = CalculaMi(spLAC);
% Prepara valor de  $\mu$ 
titulos = {' $\mu_{LAC}$  (g/cel*dia)'; ' $\mu_{LAC}$  (g/cel*h)'; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A17')
pause( 0.1 );
warning on;

% Calcula  $\mu_{NH4}$ 
[mi R_mi] = CalculaMi(spNH4);
% Prepara valor de  $\mu$ 
titulos = {' $\mu_{NH4}$  (g/cel*dia)'; ' $\mu_{NH4}$  (g/cel*h)'; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A21')
pause( 0.1 );
warning on;

% Calcula  $\mu_{ALA}$ 
[mi R_mi] = CalculaMi(spALA);
% Prepara valor de  $\mu$ 
titulos = {' $\mu_{ALA}$  (g/cel*dia)'; ' $\mu_{ALA}$  (g/cel*h)'; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A25')
pause( 0.1 );
warning on;

% Calcula  $\mu_{Prode}$ 
[mi R_mi] = CalculaMi(spProde);
% Prepara valor de  $\mu$ 
Tit1 = [ ' $\mu$  NomeProduto 'e (g/cel*dia)' ];
Tit2 = [ ' $\mu$  NomeProduto 'e (g/cel*h)' ];
titulos = {Tit1; Tit2; 'R = ' };
dados = [ mi mi/24 R_mi ]';
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,' $\mu$  Máximo','A29')
pause( 0.1 );
warning on;

```

```

% Calcula  $\mu$ Prod
[mi R_mi] = CalculaMi(spProd);
% Prepara valor de  $\mu$ 
Tit1 = [ 'μ' NomeProduto ' (g/cel*dia)' ];
Tit2 = [ 'μ' NomeProduto ' (g/cel*h)' ];
titulos = {Tit1; Tit2; 'R =' };
dados = [ mi mi/24 R_mi ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valor de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,'μ Máximo','A33')
pause( 0.1 );
warning on;

% Prepara os valores da curva de  $\mu$ 
titulos = { 't(d)', 'μ(1/d)' };
dados = [ xmi, ymi ];
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava os valores da curva de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,'μ Máximo','D1')
pause( 0.1 );
warning on;

% Prepara os valores da curva de  $\mu$  para ajuste
i=1;
p=1;
while (i<length(xmi))
    xmi1(p) = xmi(i);
    ymi1(p) = ymi(i);
    p = p + 1;
    i = i + 20; % Pega 1 ponto a cada 20 pontos
end
xmi1(p) = xmi(length(xmi));
ymi1(p) = ymi(length(ymi));
xmi1 = xmi1';
ymi1 = ymi1';
titulos = { 't(d)', 'μ(1/d)' };
dados = [ xmi1, ymi1 ];
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava os valores da curva de  $\mu$  no arquivo
warning off;
xlswrite(arquivo,d,'μ Máximo','G1')
pause( 0.1 );
warning on;

% Prepara Tempos
disp('Aguarde! Gravando Tempos...');
Tit1 = [ 'Tempo final da ' NomeProduto 'e,max (d)' ];
Tit2 = [ 'Tempo para ' NomeProduto ',max (d)' ];
titulos = {'Tempo inicial da fase exponencial (d)'; ...
           'Tempo final da fase exponencial (d)'; ...
           'Tempo final do crescimento (d)'; ...
           'Tempo de inicio da morte (d)'; ...
           Tit1; Tit2 };
dados = [ InicialExp FinalExp FinalCresc InicialMorte FinalProde Prodmax ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava Tempos
warning off;
xlswrite(arquivo,d,'Tempos','A1')
pause( 0.1 );
warning on;

% Prepara informações de ajuste de curva
disp('Aguarde! Gravando informações do ajuste de curva...');
titulos = {'Método de ajuste'; ...
           'Fator de suavização' };
dados = [ TipoAjuste Fator ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava informações de ajuste de curva
warning off;
xlswrite(arquivo,d,'Ajuste de curva','A1')
pause( 0.1 );
warning on;

% Grava dados modificados
disp('Aguarde! Gravando Dados Modificados...');
% Prepara dados
titulos = {'Tempo (d)','Células (cel/mL)','GLC (g/L)','GLN (g/L)','GLU (g/L)', ...
           'LAC (g/L)','NH4 (g/L)', 'ALA (g/L)', [ NomeProduto 'e (ng/1E7 cel)' ],...
           [ NomeProduto ' (μg/L)' ]};
dados = [ mt AjustaRemovidos1(mXv) AjustaRemovidos1(mGLC) AjustaRemovidos1(mGLN) ...

```



```

        AjustaRemovidos1(mGLU) AjustaRemovidos1(mLAC) AjustaRemovidos1(mNH4) ...
        AjustaRemovidos1(mALA) AjustaRemovidos1(mProde) AjustaRemovidos1(mProd)];
% Elimina linhas onde todos os elementos são -1
[dl dc] = size(dados);
i=1;
while (i<=dl)
    flag = true;
    for j=2:dc
        if (dados(i,j)~-1)
            flag = false;
        end
    end
    if (flag==true)
        dados(i,:) = [];
        dl = dl - 1;
    else
        i = i + 1;
    end
end
% Junta titulos e dados
c = num2cell(dados);
d = cat(1,titulos,c);
% Limpa planilha (100x50 células)
Linha = ' ';
Matriz = [];
for i=1:100
    Matriz = [ Matriz; Linha ];
end
warning off;
xlswrite(arquivo,Matriz,'Dados Modificados','A1')
pause( 0.1 );
% Grava arquivo xls
warning off;
xlswrite(arquivo,d,'Dados Modificados','A1')
pause( 0.1 );
warning on;
% Finaliza
disp('Dados gravados com sucesso!');
disp(' ');
end

% Ajusta dados removidos (troca realmin por -2)
function [ aj ] = AjustaRemovidos1(orig)
    aj = orig;
    for i=1:length(aj)
        if (aj(i)==realmin)
            aj(i) = -2;
        end
    end
end

function [mi R_mi] = CalculaMi(spy)
    global spt spXv umax

    % Elimina pontos inexistentes (spy=-1)
    x = []; y = []; t = [];
    cont = 1;
    for i=1:length(spy)
        if (spy(i)~-1)
            t(cont) = spt(i);
            x(cont) = spXv(i);
            y(cont) = spy(i);
            cont = cont + 1;
        end
    end
    % verifica se existem valores diferentes de -1
    if (~isempty(t))
        % Obtém início e fim da fase exponencial
        [p1 p2] = PontosExp(t,y);
        % Regressão
        xi = x(p1:p2);
        yi = y(p1:p2);
        p = polyfit(xi,yi,1); % Regressão Linear (polinômio de grau 1)
        % Calcula coeficientes
        mi = p(1);
        mi = mi * umax / 1000; % divide por 1000 para passar de mL para L
        R_mi = corrcoef(xi,yi);
        R_mi = R_mi(1,2);
    else
        mi = 0;
        R_mi = 0;
    end
end

function result=iif(cond, t, f)
    if cond
        result = t;
    else

```

```

    result = f;
end
end

```

9. GravaDadosY.m

```

% *****
% Módulo: GravaDadosY.m
% Descrição: Grava dados de conversão em arquivo xls
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

%Grava dados em arquivo xls
function GravaDadosY(arquivo)

% Variáveis Globais
VariaveisGlobais

% Prepara valores de Conversões (Y) encontrados
disp('Aguarde! Gravando Conversões (Y)...');
Tit = [ 'Y' NomeProduto '/GLC (µg/g)' ];
titulos = {'Yx/GLC (cel/g)'; 'Yx/GLN (cel/g)'; 'YLAC/x (g/cel)'; ...
           'YLAC/GLC (g/g)'; 'YNH4/x (g/cel)'; 'YNH4/GLN (g/g)'; Tit; 'YALA/GLC (g/g)';...
           'YALA/GLN (g/g)' };
dados = [ Yx_GLC Yx_GLN YLAC_x YLAC_GLC YNH4_x YNH4_GLN YProd_GLC YALA_GLC YALA_GLN ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava valores de Conversões (Y) no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','A1')
pause( 0.1 );
warning on;

% Prepara Coeficientes de Correlação
titulos = { 'R =' ; 'R =' ; 'R =' ; 'R =' ; 'R =' ; 'R =' ; 'R =' ; 'R =' ; 'R =' };
dados = [ R_Yx_GLC R_Yx_GLN R_YLAC_x R_YLAC_GLC R_YNH4_x R_YNH4_GLN R_YProd_GLC ...
          R_YALA_GLC R_YALA_GLN ];
c = num2cell(dados);
d = cat(2,titulos,c);
% Grava Coeficientes de Correlação
warning off;
xlswrite(arquivo,d,'Conversões (Y)','C1')
pause( 0.1 );
warning on;

% Grava pontos usados nas conversões
% Prepara Dados de Yx_GLC
titulos = {'Yx/GLC'};
dados = uYx_GLC;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','E1')
pause( 0.1 );
warning on;
% Prepara Dados de Yx_GLN
titulos = {'Yx/GLN'};
dados = uYx_GLN;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','F1')
pause( 0.1 );
warning on;
% Prepara Dados de YLAC_x
titulos = {'YLAC/x'};
dados = uYLAC_x;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','G1')
pause( 0.1 );
warning on;

```

```

% Prepara Dados de YLAC_GLC
titulos = {'YLAC/GLC'};
dados = uYLAC_GLC;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','H1')
pause( 0.1 );
warning on;
% Prepara Dados de YNH4_x
titulos = {'YNH4/x'};
dados = uYNH4_x;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','I1')
pause( 0.1 );
warning on;
% Prepara Dados de YNH4_GLN
titulos = {'YNH4/GLN'};
dados = uYNH4_GLN;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','J1')
pause( 0.1 );
warning on;
% Prepara Dados de YProd_GLC
titulos = {[ 'Y' NomeProduto '/GLC' ]};
dados = uYProd_GLC;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','K1')
pause( 0.1 );
warning on;
% Prepara Dados de YALA_GLC
titulos = {'YALA/GLC'};
dados = uYALA_GLC;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','L1')
pause( 0.1 );
warning on;
% Prepara Dados de YALA_GLN
titulos = {'YALA/GLN'};
dados = uYALA_GLN;
c = num2cell(dados);
d = cat(1,titulos,c);
% Grava Dados de Y no arquivo
warning off;
xlswrite(arquivo,d,'Conversões (Y)','M1')
pause( 0.1 );
warning on;
% Finaliza
disp('Dados de Conversão gravados com sucesso!');
disp(' ');
end

```

10. LeDados.m

```

% *****
% Módulo: LeDados.m
% Descrição: Lê dados de arquivo xls
% Autor: Marilena Martins Pamboukian
% Versão: 2.3
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Le dados de arquivo xls
function [LeituraCorreta] = LeDados(arquivo)

```

```

    spXd = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oXt))
    [spt spXt] = Ajusta_Curva(ot,oXt,TipoAjuste); % Interpolação
else
    spXt = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oViab))
    [spt spViab] = Ajusta_Curva(ot,oViab,TipoAjuste); % Interpolação
else
    spViab = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLC))
    [spt spGLC] = Ajusta_Curva(ot,oGLC,TipoAjuste); % Interpolação
else
    spGLC = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLN))
    [spt spGLN] = Ajusta_Curva(ot,oGLN,TipoAjuste); % Interpolação
else
    spGLN = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLU))
    [spt spGLU] = Ajusta_Curva(ot,oGLU,TipoAjuste); % Interpolação
else
    spGLU = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oLAC))
    [spt spLAC] = Ajusta_Curva(ot,oLAC,TipoAjuste); % Interpolação
else
    spLAC = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oNH4))
    [spt spNH4] = Ajusta_Curva(ot,oNH4,TipoAjuste); % Interpolação
else
    spNH4 = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oALA))
    [spt spALA] = Ajusta_Curva(ot,oALA,TipoAjuste); % Interpolação
else
    spALA = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oProde))
    [spt spProde] = Ajusta_Curva(ot,oProde,TipoAjuste); % Interpolação
else
    spProde = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oProd))
    [spt spProd] = Ajusta_Curva(ot,oProd,TipoAjuste); % Interpolação
else
    spProd = -1.*ones(300,1); % 300 pontos na interpolação
end
disp('Dados Alisados não encontrados!');
end

% Lê umax do arquivo xls
if (cell2mat(strfind(descr,'μ Máximo'))==1)
    disp('Aguarde! Lendo umax...');
    matriz = xlsread(arquivo,'μ Máximo','B1:B2');
    umax = matriz(1,1);
    R_umax = matriz(2,1);
    disp('    umax lido!');
else
    % Inicializa dados para garantir que existam
    umax = 0;
    R_umax = 0;
    disp('umax não encontrado!');
end

% Lê Produções (Y) do arquivo xls
if (cell2mat(strfind(descr,'Conversões (Y)'))==1)
    disp('Aguarde! Lendo Conversões (Y)...');
    matriz = xlsread(arquivo,'Conversões (Y)','B1:B9');
    Yx_GLC = matriz(1,1);
    Yx_GLN = matriz(2,1);
    YLAC_x = matriz(3,1);
    YLAC_GLC = matriz(4,1);
    YNH4_x = matriz(5,1);
    YNH4_GLN = matriz(6,1);
    YProd_GLC = matriz(7,1);
    YALA_GLC = matriz(8,1);
    YALA_GLN = matriz(9,1);
    matriz = xlsread(arquivo,'Conversões (Y)','D1:D9');
    R_Yx_GLC = matriz(1,1);
    R_Yx_GLN = matriz(2,1);
    R_YLAC_x = matriz(3,1);
    R_YLAC_GLC = matriz(4,1);
    R_YNH4_x = matriz(5,1);

```

```

    spXd = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oXt))
    [spt spXt] = Ajusta_Curva(ot,oXt,TipoAjuste); % Interpolação
else
    spXt = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oViab))
    [spt spViab] = Ajusta_Curva(ot,oViab,TipoAjuste); % Interpolação
else
    spViab = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLC))
    [spt spGLC] = Ajusta_Curva(ot,oGLC,TipoAjuste); % Interpolação
else
    spGLC = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLN))
    [spt spGLN] = Ajusta_Curva(ot,oGLN,TipoAjuste); % Interpolação
else
    spGLN = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oGLU))
    [spt spGLU] = Ajusta_Curva(ot,oGLU,TipoAjuste); % Interpolação
else
    spGLU = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oLAC))
    [spt spLAC] = Ajusta_Curva(ot,oLAC,TipoAjuste); % Interpolação
else
    spLAC = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oNH4))
    [spt spNH4] = Ajusta_Curva(ot,oNH4,TipoAjuste); % Interpolação
else
    spNH4 = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oALA))
    [spt spALA] = Ajusta_Curva(ot,oALA,TipoAjuste); % Interpolação
else
    spALA = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oProde))
    [spt spProde] = Ajusta_Curva(ot,oProde,TipoAjuste); % Interpolação
else
    spProde = -1.*ones(300,1); % 300 pontos na interpolação
end
if (~Estah_Vazio(oProd))
    [spt spProd] = Ajusta_Curva(ot,oProd,TipoAjuste); % Interpolação
else
    spProd = -1.*ones(300,1); % 300 pontos na interpolação
end
disp('Dados Alisados não encontrados!');
end

% Lê umax do arquivo xls
if (cell2mat(strfind(descr,'μ Máximo'))==1)
    disp('Aguarde! Lendo umax...');
    matriz = xlsread(arquivo,'μ Máximo','B1:B2');
    umax = matriz(1,1);
    R_umax = matriz(2,1);
    disp('    umax lido!');
else
    % Inicializa dados para garantir que existam
    umax = 0;
    R_umax = 0;
    disp('umax não encontrado!');
end

% Lê Produções (Y) do arquivo xls
if (cell2mat(strfind(descr,'Conversões (Y)'))==1)
    disp('Aguarde! Lendo Conversões (Y)...');
    matriz = xlsread(arquivo,'Conversões (Y)','B1:B9');
    Yx_GLC = matriz(1,1);
    Yx_GLN = matriz(2,1);
    YLAC_x = matriz(3,1);
    YLAC_GLC = matriz(4,1);
    YNH4_x = matriz(5,1);
    YNH4_GLN = matriz(6,1);
    YProd_GLC = matriz(7,1);
    YALA_GLC = matriz(8,1);
    YALA_GLN = matriz(9,1);
    matriz = xlsread(arquivo,'Conversões (Y)','D1:D9');
    R_Yx_GLC = matriz(1,1);
    R_Yx_GLN = matriz(2,1);
    R_YLAC_x = matriz(3,1);
    R_YLAC_GLC = matriz(4,1);
    R_YNH4_x = matriz(5,1);

```

```

R_YNH4_GLN = matriz(6,1);
R_YProd_GLC = matriz(7,1);
R_YALA_GLC = matriz(8,1);
R_YALA_GLN = matriz(9,1);
disp('    Conversões (Y) lidas!');
else
    % Inicializa dados para garantir que existam
    Yx_GLC = 0;
    Yx_GLN = 0;
    YLAC_x = 0;
    YLAC_GLC = 0;
    YNH4_x = 0;
    YNH4_GLN = 0;
    YProd_GLC = 0;
    YALA_GLC = 0;
    YALA_GLN = 0;
    R_Yx_GLC = 0;
    R_Yx_GLN = 0;
    R_YLAC_x = 0;
    R_YLAC_GLC = 0;
    R_YNH4_x = 0;
    R_YNH4_GLN = 0;
    R_YProd_GLC = 0;
    R_YALA_GLC = 0;
    R_YALA_GLN = 0;
    disp('Conversões (Y) não encontrados!');
end

% Lê pontos usados nas conversões
if (cell2mat(strfind(descr,'Conversões (Y)'))==1)
    disp('Aguarde! Lendo pontos usados nas conversões...');
    % Lê pontos utilizados na conversão
    uYx_GLC = xlsread(arquivo,'Conversões (Y)','E1:E1000');
    uYx_GLN = xlsread(arquivo,'Conversões (Y)','F1:F1000');
    uYLAC_x = xlsread(arquivo,'Conversões (Y)','G1:G1000');
    uYLAC_GLC = xlsread(arquivo,'Conversões (Y)','H1:H1000');
    uYNH4_x = xlsread(arquivo,'Conversões (Y)','I1:I1000');
    uYNH4_GLN = xlsread(arquivo,'Conversões (Y)','J1:J1000');
    uYProd_GLC = xlsread(arquivo,'Conversões (Y)','K1:K1000');
    uYALA_GLC = xlsread(arquivo,'Conversões (Y)','L1:L1000');
    uYALA_GLN = xlsread(arquivo,'Conversões (Y)','M1:M1000');
    disp('    Pontos usados nas conversões lidos!');
else
    % Inicializa dados modificados para garantir que existam
    uYx_GLC = [];
    uYx_GLN = [];
    uYLAC_x = [];
    uYLAC_GLC = [];
    uYNH4_x = [];
    uYNH4_GLN = [];
    uYProd_GLC = [];
    uYALA_GLC = [];
    uYALA_GLN = [];
    disp('Valores Modificados nas conversões não encontrados!');
end

% Lê tempos do arquivo xls
if (cell2mat(strfind(descr,'Tempos'))==1)
    disp('Aguarde! Lendo Tempos...');
    matriz = xlsread(arquivo,'Tempos','B1:B6');
    InicialExp = matriz(1,1);
    FinalExp = matriz(2,1);
    FinalCresc = matriz(3,1);
    InicialMorte = matriz(4,1);
    Prodmax = matriz(5,1);
    FinalProde = matriz(6,1);
    disp('    Tempos lidos!');
else
    % Inicializa dados para garantir que existam
    InicialExp = 0;
    FinalExp = 0;
    FinalCresc = 0;
    InicialMorte = 0;
    FinalProde = 0;
    Prodmax = 0;
    disp('Tempos não encontrados!');
end

% Lê Dados Modificados do arquivo xls
if (cell2mat(strfind(descr,'Dados Modificados'))==1)
    disp('Aguarde! Lendo Dados Modificados...');
    % Ajusta dados modificados
    matriz = xlsread(arquivo,'Dados Modificados');
    mt = matriz(:,1);
    mXv = AjustaRemovidos2(matriz(:,2));
    mGLC = AjustaRemovidos2(matriz(:,3));
    mGLN = AjustaRemovidos2(matriz(:,4));
    mGLU = AjustaRemovidos2(matriz(:,5));

```

```

mLAC = AjustaRemovidos2(matriz(:,6));
mNH4 = AjustaRemovidos2(matriz(:,7));
mALA = AjustaRemovidos2(matriz(:,8));
mProde = AjustaRemovidos2(matriz(:,9));
mProd = AjustaRemovidos2(matriz(:,10));
% Ajusta dados inseridos
[ instXv insXv ] = AjustaInseridos(ot,oXv,mt,mXv); % Pontos inseridos
[ instGLC insGLC ] = AjustaInseridos(ot,oGLC,mt,mGLC); % Pontos inseridos
[ instGLN insGLN ] = AjustaInseridos(ot,oGLN,mt,mGLN); % Pontos inseridos
[ instGLU insGLU ] = AjustaInseridos(ot,oGLU,mt,mGLU); % Pontos inseridos
[ instLAC insLAC ] = AjustaInseridos(ot,oLAC,mt,mLAC); % Pontos inseridos
[ instNH4 insNH4 ] = AjustaInseridos(ot,oNH4,mt,mNH4); % Pontos inseridos
[ instALA insALA ] = AjustaInseridos(ot,oALA,mt,mALA); % Pontos inseridos
[ instProde insProde ] = AjustaInseridos(ot,oProde,mt,mProde); % Pontos inseridos
[ instProd insProd ] = AjustaInseridos(ot,oProd,mt,mProd); % Pontos inseridos
% Ajusta dados atuais
[ at aXv ] = AjustaAtuais(ot,oXv,mt,mXv); % Pontos atuais
[ at aGLC ] = AjustaAtuais(ot,oGLC,mt,mGLC); % Pontos atuais
[ at aGLN ] = AjustaAtuais(ot,oGLN,mt,mGLN); % Pontos atuais
[ at aGLU ] = AjustaAtuais(ot,oGLU,mt,mGLU); % Pontos atuais
[ at aLAC ] = AjustaAtuais(ot,oLAC,mt,mLAC); % Pontos atuais
[ at aNH4 ] = AjustaAtuais(ot,oNH4,mt,mNH4); % Pontos atuais
[ at aALA ] = AjustaAtuais(ot,oALA,mt,mALA); % Pontos atuais
[ at aProde ] = AjustaAtuais(ot,oProde,mt,mProde); % Pontos atuais
[ at aProd ] = AjustaAtuais(ot,oProd,mt,mProd); % Pontos atuais
disp(' Dados Modificados lidos!');
else
% Inicializa dados modificados para garantir que existam
mt = ot;
mXv = oXv;
mGLC = oGLC;
mGLN = oGLN;
mGLU = oGLU;
mLAC = oLAC;
mNH4 = oNH4;
mALA = oALA;
mProde = oProde;
mProd = oProd;
% Inicializa dados inseridos para garantir que existam
instXv = []; insXv = []; % Pontos inseridos
instGLC = []; insGLC = []; % Pontos inseridos
instGLN = []; insGLN = []; % Pontos inseridos
instGLU = []; insGLU = []; % Pontos inseridos
instLAC = []; insLAC = []; % Pontos inseridos
instNH4 = []; insNH4 = []; % Pontos inseridos
instALA = []; insALA = []; % Pontos inseridos
instProde = []; insProde = []; % Pontos inseridos
instProd = []; insProd = []; % Pontos inseridos
% Inicializa dados atuais para garantir que existam
at = ot;
aXv = oXv; % Pontos originais
aGLC = oGLC; % Pontos originais
aGLN = oGLN; % Pontos originais
aGLU = oGLU; % Pontos originais
aLAC = oLAC; % Pontos originais
aNH4 = oNH4; % Pontos originais
aALA = oALA; % Pontos originais
aProde = oProde; % Pontos originais
aProd = oProd; % Pontos originais
disp('Dados Modificados não encontrados!');
end
% Finaliza
LeituraCorreta = true;
disp('Dados lidos com sucesso!');
disp(' ');
end

% Ajusta dados removidos (troca -2 por realmin)
function [ aj ] = AjustaRemovidos2(mod)
aj = mod;
for i=1:length(aj)
if (aj(i)==-2)
aj(i) = realmin;
end
end
end

% Cria vetor de inseridos com dados modificados que não existiam nos originais
function [ ajX ajY ] = AjustaInseridos(origX, origY, modX, modY) %#ok<INUSL>
ajX = [];
ajY = [];
cont = 1;
for i=1:length(modX)
p = find(origX==modX(i)); % Verifica se ponto modificado existe no original
if (isempty(p))
if (modY(i)~= -1) % Verifica se não foi ponto adicionado no tratamento
ajX(cont) = modX(i); %#ok<AGROW>
ajY(cont) = modY(i); %#ok<AGROW>

```

```

        cont = cont + 1;
    end
end
end
end

% Cria vetor de dados atuais, a partir dos dados modificados,
% com valores originais + inseridos
function [ ajX ajY ] = AjustaAtuais(origX, origY, modX, modY)
    ajX = modX;
    ajY = modY;
    for i=1:length(modX)
        p = find(origX==modX(i)); % Verifica posição do ponto modificado no original
        if (~isempty(p))
            ajY(i) = origY(p);      % Copia valor original
        end
    end
end
end
end

```

11. Painei.m

```

% *****
% Módulo: Painei.m
% Descrição: Painei de Controle para ajuste de tempos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Painei
function Painei
    % Variáveis globais
    VariaveisGlobais
    % Cria figura centralizada na tela
    screensize = get(0,'ScreenSize'); % Tamanho da tela
    FigAltura = 470;
    hFigPainei = figure('name','Tempos','NumberTitle','off',...
        'WindowStyle','normal',...
        'Position',[ 10 (screensize(4)-FigAltura)./2 200 FigAltura ], ...
        'Color',[ 0.8 0.8 0.8 ], ...
        'MenuBar','none');

    pause( 0.1 );
    winontop('Tempos',1);
    % Cria controles
    chkInicial = uicontrol(hFigPainei,...
        'Style','checkbox',...
        'String','Tempo inicial da fase exponencial',...
        'FontSize', 8, 'FontName','Arial',...
        'HorizontalAlignment','left', ...
        'Position',[ 10 FigAltura-30 200 20 ], ...
        'BackgroundColor',[ 0.8 0.8 0.8 ], ...
        'ForegroundColor','b', ...
        'Tag','chkInicial');

    edtInicial = uicontrol(hFigPainei,...
        'Style','edit',...
        'String', num2str(InicialExp),...
        'FontSize', 8, 'FontName','Arial',...
        'HorizontalAlignment','left', ...
        'Position',[ 30 FigAltura-50 100 20 ], ...
        'BackgroundColor',[ 1 1 1 ], ...
        'Tag','edtInicial');

    lblError1 = uicontrol(hFigPainei,...
        'Style','text',...
        'String','Inválido!',...
        'FontSize', 8, 'FontName','Arial',...
        'HorizontalAlignment','left', ...
        'Position',[ 140 FigAltura-55 100 20 ], ...
        'BackgroundColor',[ 0.8 0.8 0.8 ], ...
        'ForegroundColor','r', ...
        'Visible','off',...
        'Tag','lblError1');

    chkFinal = uicontrol(hFigPainei,...
        'Style','checkbox',...
        'String','Tempo final da fase exponencial',...
        'FontSize', 8, 'FontName','Arial',...
        'HorizontalAlignment','left', ...
        'Position',[ 10 FigAltura-75 200 20 ], ...
        'BackgroundColor',[ 0.8 0.8 0.8 ], ...

```



```

        'ForegroundColor', 'r', ...
        'Tag','chkFinal');
edtFinal = uicontrol(hFigPainel,...
    'Style','edit',...
    'String', num2str(FinalExp), ...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 30 FigAltura-95 100 20 ], ...
    'BackgroundColor', [ 1 1 1 ], ...
    'Tag','edtFinal');
lblErro2 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Inválido!',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 140 FigAltura-100 100 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Visible','off',...
    'Tag','lblErro2');
chkFinalCresc = uicontrol(hFigPainel,...
    'Style','checkbox',...
    'String', 'Tempo final do crescimento',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 10 FigAltura-120 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', [ 0.0 0.7 0.0 ], ...
    'Tag','chkFinalCresc');
edtFinalCresc = uicontrol(hFigPainel,...
    'Style','edit',...
    'String', num2str(FinalCresc), ...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 30 FigAltura-140 100 20 ], ...
    'BackgroundColor', [ 1 1 1 ], ...
    'Tag','edtFinalCresc');
lblErro3 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Inválido!',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 140 FigAltura-145 100 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Visible','off',...
    'Tag','lblErro3');
chkInicialMorte = uicontrol(hFigPainel,...
    'Style','checkbox',...
    'String', 'Tempo de inicio da morte',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 10 FigAltura-165 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'm', ...
    'Tag','chkInicialMorte');
edtInicialMorte = uicontrol(hFigPainel,...
    'Style','edit',...
    'String', num2str(InicialMorte), ...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 30 FigAltura-185 100 20 ], ...
    'BackgroundColor', [ 1 1 1 ], ...
    'Tag','edtInicialMorte');
lblErro4 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Inválido!',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 140 FigAltura-190 100 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Visible','off',...
    'Tag','lblErro4');
Tit = [ 'Tempo para ' NomeProduto ' ,max' ];
chkProdmax = uicontrol(hFigPainel,...
    'Style','checkbox',...
    'String', Tit,...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 10 FigAltura-210 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', [ 1.0 0.5 0.0 ], ...
    'Tag','chkProdmax');
edtProdmax = uicontrol(hFigPainel,...
    'Style','edit',...
    'String', num2str(Prodmax), ...
    'FontSize', 8, 'FontName','Arial',...

```

```

        'HorizontalAlignment', 'left', ...
        'Position', [ 30 FigAltura-230 100 20 ], ...
        'BackgroundColor', [ 1 1 1 ], ...
        'Tag','edtProdmax');
lblErro5 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Inválido!',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 140 FigAltura-235 100 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Visible','off',...
    'Tag','lblErro5');
Tit = [ 'Tempo final da ' NomeProduto 'e,max' ];
chkFinalProde = uicontrol(hFigPainel,...
    'Style','checkbox',...
    'String', Tit,...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 10 FigAltura-255 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', [ 0.5 0.0 0.5 ], ...
    'Tag','chkFinalProde');
edtFinalProde = uicontrol(hFigPainel,...
    'Style','edit',...
    'String', num2str(FinalProde), ...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 30 FigAltura-275 100 20 ], ...
    'BackgroundColor', [ 1 1 1 ], ...
    'Tag','edtFinalProde');
lblErro6 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Inválido!',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 140 FigAltura-280 100 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Visible','off',...
    'Tag','lblErro6');
chkMiCte = uicontrol(hFigPainel,...
    'Style','checkbox','Callback', @chkMiCteClick,...
    'String', 'Admitir  $\mu$  constante',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 10 FigAltura-305 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'k', ...
    'Tag','chkMiCte');
btnAtualizar = uicontrol(hFigPainel,'Callback', @btnAtualizarClick,...
    'Style','pushbutton',...
    'String', 'Atualizar',...
    'FontSize', 8, 'FontName','Arial',...
    'Position', [ 10 FigAltura-345 170 30 ], ...
    'Tag','btnAtualizar');
lblResp1 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', ' $\mu_{max} = 0.000000$  (1/dia)',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 20 FigAltura-380 180 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'k', ...
    'Tag','lblResp1');
lblResp2 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', ' $\mu_{max} = 0.000000$  (1/h)',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 20 FigAltura-400 180 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'k', ...
    'Tag','lblResp2');
lblResp3 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'lnX calculado = 0.000000',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'left', ...
    'Position', [ 20 FigAltura-420 180 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'k', ...
    'Tag','lblResp3');
lblResp4 = uicontrol(hFigPainel,...
    'Style','text',...
    'String', 'Coef. Correlação = 0.000',...
    'FontSize', 8, 'FontName','Arial',...

```

```

        'HorizontalAlignment', 'left', ...
        'Position', [ 20 FigAltura-440 180 20 ], ...
        'BackgroundColor', [ 0.8 0.8 0.8 ], ...
        'ForegroundColor', 'k', ...
        'Tag','lblResp4');
lblMari = uicontrol(hFigPainel,...
    'Style','text',...
    'String', '© 2010 - Marilena M. Pamboukian',...
    'FontSize', 8, 'FontName','Arial',...
    'HorizontalAlignment', 'center', ...
    'Position', [ 0 FigAltura-470 200 20 ], ...
    'BackgroundColor', [ 0.8 0.8 0.8 ], ...
    'ForegroundColor', 'r', ...
    'Tag','lblMari');
end

function btnAtualizarClick(hObject,eventdata,handles)
% Variáveis globais
VariaveisGlobais
% Mensagens de Erro ficam invisíveis
set(lblErro1,'Visible','off');
set(lblErro2,'Visible','off');
set(lblErro3,'Visible','off');
set(lblErro4,'Visible','off');
set(lblErro5,'Visible','off');
set(lblErro6,'Visible','off');
% Tempo inicial da fase exponencial
InicialExp = str2num(get(edtInicial,'String'));
if length(InicialExp)~=1
    set(lblErro1,'Visible','on');
    InicialExp = 0;
    return
end
if isnan(InicialExp)
    set(lblErro1,'Visible','on');
    InicialExp = 0;
    return
end
% Tempo final da fase exponencial
FinalExp = str2num(get(edtFinal,'String'));
if length(FinalExp)~=1
    set(lblErro2,'Visible','on');
    FinalExp = 0;
    return
end
if isnan(FinalExp)
    set(lblErro2,'Visible','on');
    FinalExp = 0;
    return
end
% Tempo final do crescimento
FinalCresc = str2num(get(edtFinalCresc,'String'));
if length(FinalCresc)~=1
    set(lblErro3,'Visible','on');
    FinalCresc = 0;
    return
end
if isnan(FinalCresc)
    set(lblErro3,'Visible','on');
    FinalCresc = 0;
    return
end
% Tempo de inicio da morte
InicialMorte = str2num(get(edtInicialMorte,'String'));
if length(InicialMorte)~=1
    set(lblErro4,'Visible','on');
    InicialMorte = 0;
    return
end
if isnan(InicialMorte)
    set(lblErro4,'Visible','on');
    InicialMorte = 0;
    return
end
% Tempo para Prodmax
Prodmax = str2num(get(edtProdmax,'String'));
if length(Prodmax)~=1
    set(lblErro5,'Visible','on');
    Prodmax = 0;
    return
end
if isnan(Prodmax)
    set(lblErro5,'Visible','on');
    Prodmax = 0;
    return
end
% Tempo final da Prode,max
FinalProde = str2num(get(edtFinalProde,'String'));

```

13. Remove_Pontos.m

```
% *****
% Módulo: Remove_Pontos.m
% Descrição: Remove do vetor pontos que possuem valor -1
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Remove pontos que possuem valor -1
function [x1 y1] = Remove_Pontos(x, y)
n = length(x);
x1 = [];
y1 = [];
cont = 1;
for i = 1:n
    if (x(i)~= -1 && y(i)~= -1)
        x1(cont) = x(i); %#ok<AGROW>
        y1(cont) = y(i); %#ok<AGROW>
        cont=cont+1;
    end
end
x1 = x1';
y1 = y1';
end
```

14. SalvaGraficos.m

```
% *****
% Módulo: SalvaGraficos.m
% Descrição: Salva os graficos no formato BMP
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

function SalvaGraficos(local)
    VariaveisGlobais

    figure(hFigLnXvF);
    pause( 0.1 );
    I = getframe(hFigLnXvF);
    imwrite(I.cdata,[local 'Fig01_Trata_mi.bmp'])
    figure(hFigMi);
    pause( 0.1 );
    I = getframe(hFigMi);
    imwrite(I.cdata,[local 'Fig02_Curva_ux.bmp'])
    if (~Estah_Vazio(oViab))
        figure(hFigViab);
        pause( 0.1 );
        I = getframe(hFigViab);
        imwrite(I.cdata,[local 'Fig03_Viab.bmp'])
    end
    if (~Estah_Vazio(oProd))
        figure(hFigProd);
        pause( 0.1 );
        I = getframe(hFigProd);
        imwrite(I.cdata,[local 'Fig04_Prod.bmp'])
    end
    if (~(Estah_Vazio(oProd) && Estah_Vazio(oProde)))
        figure(hFigProde);
        pause( 0.1 );
        I = getframe(hFigProde);
        imwrite(I.cdata,[local 'Fig05_Prode_PXt.bmp'])
    end
    if (~Estah_Vazio(oGLC))
        figure(hFigGLC);
        pause( 0.1 );
        I = getframe(hFigGLC);
        imwrite(I.cdata,[local 'Fig06_GLC.bmp'])
    end
end
```

13. Remove_Pontos.m

```
% *****
% Módulo: Remove_Pontos.m
% Descrição: Remove do vetor pontos que possuem valor -1
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% Remove pontos que possuem valor -1
function [x1 y1] = Remove_Pontos(x, y)
n = length(x);
x1 = [];
y1 = [];
cont = 1;
for i = 1:n
    if (x(i)~= -1 && y(i)~= -1)
        x1(cont) = x(i); %#ok<AGROW>
        y1(cont) = y(i); %#ok<AGROW>
        cont=cont+1;
    end
end
x1 = x1';
y1 = y1';
end
```

14. SalvaGraficos.m

```
% *****
% Módulo: SalvaGraficos.m
% Descrição: Salva os graficos no formato BMP
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

function SalvaGraficos(local)
    VariaveisGlobais

    figure(hFigLnXvF);
    pause( 0.1 );
    I = getframe(hFigLnXvF);
    imwrite(I.cdata,[local 'Fig01_Trata_mi.bmp'])
    figure(hFigMi);
    pause( 0.1 );
    I = getframe(hFigMi);
    imwrite(I.cdata,[local 'Fig02_Curva_ux.bmp'])
    if (~Estah_Vazio(oViab))
        figure(hFigViab);
        pause( 0.1 );
        I = getframe(hFigViab);
        imwrite(I.cdata,[local 'Fig03_Viab.bmp'])
    end
    if (~Estah_Vazio(oProd))
        figure(hFigProd);
        pause( 0.1 );
        I = getframe(hFigProd);
        imwrite(I.cdata,[local 'Fig04_Prod.bmp'])
    end
    if (~(Estah_Vazio(oProd) && Estah_Vazio(oProde)))
        figure(hFigProde);
        pause( 0.1 );
        I = getframe(hFigProde);
        imwrite(I.cdata,[local 'Fig05_Prode_PXt.bmp'])
    end
    if (~Estah_Vazio(oGLC))
        figure(hFigGLC);
        pause( 0.1 );
        I = getframe(hFigGLC);
        imwrite(I.cdata,[local 'Fig06_GLC.bmp'])
    end
end
```

```

if (~Estah_Vazio(oGLN))
    figure(hFigGLN);
    pause( 0.1 );
    I = getframe(hFigGLN);
    imwrite(I.cdata,[local 'Fig07_GLN.bmp'])
end
if (~Estah_Vazio(oGLU))
    figure(hFigGLU);
    pause( 0.1 );
    I = getframe(hFigGLU);
    imwrite(I.cdata,[local 'Fig08_GLU.bmp'])
end
if (~Estah_Vazio(oLAC))
    figure(hFigLAC);
    pause( 0.1 );
    I = getframe(hFigLAC);
    imwrite(I.cdata,[local 'Fig09_LAC.bmp'])
end
if (~Estah_Vazio(oNH4))
    figure(hFigNH4);
    pause( 0.1 );
    I = getframe(hFigNH4);
    imwrite(I.cdata,[local 'Fig10_NH4.bmp'])
end
if (~Estah_Vazio(oALA))
    figure(hFigALA);
    pause( 0.1 );
    I = getframe(hFigALA);
    imwrite(I.cdata,[local 'Fig11_ALA.bmp'])
end
end
end

```

15. SalvaGraficosY.m

```

% *****
% Módulo: SalvaGraficosY.m
% Descrição: Salva os graficos das conversões no formato BMP
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

function SalvaGraficosY(local)
    VariaveisGlobais

    if (~Estah_Vazio(oGLC))
        figure(hFig1);
        pause( 0.1 );
        I = getframe(hFig1);
        imwrite(I.cdata,[local 'Fig12_Yx_GLC.bmp'])
    end
    if (~Estah_Vazio(oGLN))
        figure(hFig2);
        pause( 0.1 );
        I = getframe(hFig2);
        imwrite(I.cdata,[local 'Fig13_Yx_GLN.bmp'])
    end
    if (~Estah_Vazio(oLAC))
        figure(hFig3);
        pause( 0.1 );
        I = getframe(hFig3);
        imwrite(I.cdata,[local 'Fig14_YLAC_x.bmp'])
    end
    if (~Estah_Vazio(oLAC) && ~Estah_Vazio(oGLC))
        figure(hFig4);
        pause( 0.1 );
        I = getframe(hFig4);
        imwrite(I.cdata,[local 'Fig15_YLAC_GLC.bmp'])
    end
    if (~Estah_Vazio(oNH4))
        figure(hFig5);
        pause( 0.1 );
        I = getframe(hFig5);
        imwrite(I.cdata,[local 'Fig16_YNH4_x.bmp'])
    end
    if (~Estah_Vazio(oNH4) && ~Estah_Vazio(oGLN))
        figure(hFig6);
        pause( 0.1 );
    end

```

```

        I = getframe(hFig6);
        imwrite(I.cdata,[local 'Fig17_YNH4_GLN.bmp'])
    end
    if (~Estah_Vazio(oProd) && ~Estah_Vazio(oGLC))
        figure(hFig7);
        pause( 0.1 );
        I = getframe(hFig7);
        imwrite(I.cdata,[local 'Fig18_YProd_GLC.bmp'])
    end
    if (~Estah_Vazio(oALA) && ~Estah_Vazio(oGLC))
        figure(hFig8);
        pause( 0.1 );
        I = getframe(hFig8);
        imwrite(I.cdata,[local 'Fig19_YALA_GLC.bmp'])
    end
    if (~Estah_Vazio(oALA) && ~Estah_Vazio(oGLN))
        figure(hFig9);
        pause( 0.1 );
        I = getframe(hFig9);
        imwrite(I.cdata,[local 'Fig20_YALA_GLN.bmp'])
    end
end
end

```

16. TrataDados.m

```

% *****
% Módulo: TrataDados.m
% Descrição: Faz tratamento de dados experimentais
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% Restrição: necessária instalação do Java (Swing)
% *****

% Faz tratamento de dados
function TrataDados
% Limpa o Workspace (não afeta dados lidos)
clear
% Limpa Command Window
clc
% Se precisar declarar Variáveis Globais não pode ser antes do clear
VariaveisGlobais
% Ajusta formato de exibição dos valores
format short e
% Define todas as janelas de figuras como "docked"
set(0,'DefaultFigureWindowStyle','docked')
% Leitura de dados
resp = questdlg('Efetuar leitura de dados ?','Atenção','Sim','Não','Sim');
if (strcmp(resp,'Sim')==1)
    [filename, caminho] = uigetfile('*.xls', 'Selecione o arquivo de dados','Dados.xls');
    if isequal(filename,0) || isequal(caminho,0)
        return
    else
        arq = fullfile(caminho, filename);
        end
        if (LeDados(arq)==false)
            return;
        end
        Dados_Lidos = 'sim';
    end
    if (strcmp(Dados_Lidos,'sim')==0)
        return;
    end
end
% Ajuste de curvas
resp = questdlg('Efetuar ajuste de curvas ?','Atenção','Sim','Não','Sim');
if (strcmp(resp,'Sim')==1)
    % Exibe Painel
    Painel;
    % Faz ajuste de pontos
    Graficos;
    % Gera gráficos finais
    GraficosFinais;
    % Gravação de dados
    resp = questdlg('Efetuar gravação de dados ajustados ?','Atenção','Sim','Não','Sim');
    if (strcmp(resp,'Sim')==1)
        [filename, caminho] = uiputfile('*.xls', 'Selecione o arquivo de dados',arq);
        if ~(isequal(filename,0) || isequal(caminho,0))

```

```

        arq = fullfile(caminho, filename);
        GravaDados(arq);
    end
end
% Salva Gráficos
resp = questdlg('Salvar os gráficos ?', 'Atenção', 'Sim', 'Não', 'Sim');
if (strcmp(resp, 'Sim')==1)
    SalvaGráficos(caminho);
end
end
% Conversões
resp = questdlg('Efetuar conversões (Y) ?', 'Atenção', 'Sim', 'Não', 'Sim');
if (strcmp(resp, 'Sim')==1)
    % Faz conversões
    Conversao;
    % Remove 'Figure 1' que aparece quando não é feito o ajuste de curvas
    if (findobj('Type', 'figure')==1)
        close all;
    end
    % Gera gráficos finais
    GraficosFinaisY;
    % Gravação de dados
    resp = questdlg('Efetuar gravação de dados de Conversão ?', 'Atenção', 'Sim', 'Não', 'Sim');
    if (strcmp(resp, 'Sim')==1)
        [filename, caminho] = uinputfile('*.xls', 'Selecione o arquivo de dados', arq);
        if ~ (isequal(filename, 0) || isequal(caminho, 0))
            arq = fullfile(caminho, filename);
            GravaDadosY(arq);
        end
    end
    % Salva Gráficos
    resp = questdlg('Salvar os gráficos ?', 'Atenção', 'Sim', 'Não', 'Sim');
    if (strcmp(resp, 'Sim')==1)
        SalvaGráficosY(caminho);
    end
end
end
end

```

17. VariaveisGlobais.m

```

% *****
% Módulo: VariáveisGlobais.m
% Descrição: variáveis globais incluídas nos demais arquivos
% Autor: Marilena Martins Pamboukian
% Versão: 2.5
% Criação: 21/04/2009
% Última modificação: 04/07/2012
% *****

% *****
%
% DADOS DO ENSAIO
%
% *****

% -----
% Nome do ensaio que irá aparecer nos gráficos
global NomeEnsaio %#ok
% -----
% Nome do produto que irá aparecer nos gráficos
global NomeProduto %#ok
% -----

% *****
%
% OBJETOS EXISTENTES NO GRÁFICO DE AJUSTE DE PONTOS
%
% *****

% -----
% Figura onde serão exibidos todos os gráficos para o ajuste de pontos
global hFigGG %#ok
% -----
% Eixo principal onde será plotado o gráfico que está sendo editado
global hEixo1 %#ok
% -----
% Eixos onde serão plotados os gráficos que aparecem no fundo
% Todos os gráficos são normalizados, exceto o de Viabilidade que é dividido por 100
global hEixoXv hEixoProd hEixoProde hEixoGLC hEixoGLN hEixoGLU %#ok

```



```

global hEixoLAC hEixoNH4 hEixoALA hEixoViab hEixoPXt hEixoMi hEixoLnXv hEixoVert %#ok
%
% Botões de rádio (radiobuttons) para selecionar qual gráfico está sendo editado
global radXv radProd radProde radGLC radGLN radGLU radLAC radNH4 radALA %#ok
%
% Caixas de verificação (checkboxes) para determinar quais gráficos aparecem no fundo
global chkXv chkProd chkProde chkGLC chkGLN chkGLU chkLAC chkNH4 chkALA chkViab chkPXt chkMi chkLnXv %#ok
%
% Caixas de verificação (checkboxes) para determinar quais linhas verticais aparecem no fundo
global chkInicial chkFinal chkFinalCresc chkInicialMorte chkProdmax chkFinalProde %#ok
%
% Timer que faz a atualização do gráfico de acordo com as opções do painel de controle
global TimerAtualiza %#ok
%
% *****
%
% OBJETOS EXISTENTES NO PAINEL DE CONTROLE
%
% *****
%
% Figura que representa o painel de controle
global hFigPainel %#ok
%
% Caixas de texto presentes no painel de controle
global edtInicial edtFinal edtFinalCresc edtInicialMorte edtProdmax edtFinalProde %#ok
%
% Rótulos (labels) que exibem mensagem quando caixas de texto possuem conteúdo inválido
global lblErro1 lblErro2 lblErro3 lblErro4 lblErro5 lblErro6 %#ok
%
% Rótulos (labels) que exibem umax, lnX calculado e Coeficiente de Correlação
global lblResp1 lblResp2 lblResp3 lblResp4 %#ok
%
% Caixa de verificação (checkbox) para determinar se admite mi constante
global chkMiCte %#ok
%
% *****
%
% VARIÁVEIS QUE ARMAZENAM DADOS EXPERIMENTAIS
%
% *****
%
% Variáveis que armazenam os pontos originais (o=original)
% Todos os vetores possuem o mesmo tamanho
% Valores inexistentes são representados por -1
global ot oXv oXd oXt oProd oProde oGLC oGLN oGLU oLAC oNH4 oALA oViab %#ok
%
% Variáveis que armazenam os pontos interpolados (sp=spline)
% Todos os vetores possuem o mesmo tamanho (300)
% No gráfico em edição são representados pelas variáveis xi e yi
global spt spXv spXd spXt spProd spProde spGLC spGLN spGLU splAC spNH4 spALA spViab %#ok
%
% Variáveis que armazenam a cruva de mi interpolada
% Vetores possui tamanho 300
global xmi ymi %#ok
%
% Variáveis que armazenam os pontos interpolados (i=interpolados) durante a edição
% Vetores possui tamanho 300
global xi yi %#ok
%
% Variáveis que armazenam os pontos modificados (m=modificados)
% Armazenam pontos originais, pontos inseridos e pontos removidos
% Os pontos removidos possuem coordenada y = realmin
% No gráfico em edição são representados pelas variáveis xf e yf (quadrados)
% Os vetores NÃO possuem o mesmo tamanho, por isso é necessário armazenar o tempo
global mt mXv mProd mProde mGLC mGLN mGLU mLAC mNH4 mALA %#ok
%
% Variáveis que armazenam os pontos modificados durante a edição (f=final)
% Armazenam pontos originais, pontos inseridos e pontos removidos
% Os pontos removidos possuem coordenada y = realmin
% Os vetores NÃO possuem o mesmo tamanho, por isso é necessário armazenar o tempo
global xf yf %#ok
%
% Variáveis que armazenam os pontos inseridos (ins=inseridos)
% Inicialmente são vetores vazios
% No gráfico em edição são representados pelas variáveis xins e yins (asterisco)
% Os vetores NÃO possuem o mesmo tamanho, por isso é necessário armazenar o tempo
global instXv instXv instProd instProd instProde instProde instGLC instGLC %#ok
global instGLN instGLN instGLU instGLU instLAC instLAC instNH4 instNH4 instALA instALA %#ok
%
% Variáveis que armazenam os pontos inseridos (ins=inseridos) durante a edição
% Inicialmente são vetores vazios
% Os vetores NÃO possuem o mesmo tamanho, por isso é necessário armazenar o tempo
global xins yins %#ok
%

```

```

% Variáveis que armazenam os pontos atuais (a=atual)
% Armazenam pontos originais, pontos inseridos e pontos removidos
% Os pontos removidos NÃO possuem coordenada y = realmin
% No gráfico em edição são representados pelas variáveis x e y (bolinhas)
% Os vetores NÃO possuem o mesmo tamanho, por isso é necessário armazenar o tempo
global at aXv aProd aProde aGLC aGLN aGLU aLAC aNH4 aALA %ok
%-----
% Variáveis que armazenam os pontos atuais durante a edição
% Armazenam pontos originais, pontos inseridos e pontos removidos
% Os pontos removidos NÃO possuem coordenada y = realmin
global x y %ok
%-----
% Coordenadas do último click do mouse (c=click)
global xc yc %ok
%-----
%
% *****
%
% OUTRAS VARIÁVEIS
%
% *****
%-----
% Indica a variável (gráfico) que está em edição
global VarAtual %ok
%-----
% Indica o modo de trabalho
% modo = -1 indica primeiro click do mouse
% modo = 1 indica segundo click do mouse
% modo = 2 indica edição encerrada
global modo %ok
%-----
% Informações para o undo
% pt indica o número do último ponto alterado
% y_undo indica o antigo valor (y) do último ponto alterado
% Se y_undo = realmin não é possível fazer undo
% Se y_undo = realmax então um novo ponto foi inserido
global pt y_undo %ok
%-----
% Variáveis que armazenam os valores digitados nas caixas de texto do painel de controle
global InicialExp FinalExp FinalCresc InicialMorte Prodmax FinalProde %ok
%-----
% Mi máximo e o coeficiente de correlação do lnX
global umax R_umax %ok
%-----
% Valores interpolados usando Cubic Spline que serão utilizados
% na função Atualiza Mi e GradicoMi
global spxi spyi %ok
%-----
% Fator de suavização (smoothing)
% -1 indica que será utilizado o valor default
global Fator %ok
%-----
% Tipo de ajuste de curva
% 1 = Smoothing spline, 2 = spline, 3 = Hermite
global TipoAjuste %ok
%-----
% Indica se os dados já foram lidos
global Dados_Lidos %ok
%-----
% Nome do arquivo xls e seu caminho
global arq %ok
global caminho %ok
%-----
%
% *****
%
% GRÁFICOS FINAIS
%
% *****
%-----
% Figuras correspondente a cada gráfico
global hFigLnXvF hFigMi hFigViab hFigProd hFigProde %ok
global hFigGLC hFigGLN hFigGLU hFigLAC hFigNH4 hFigALA %ok
%-----
% Eixo secundário utilizado nos gráfico final de mi
global hEixo2 %ok
%-----
%
% *****
%
% DADOS DE CONVERSÃO
%
% *****
%-----

```

19. winontop.m

```
% *****
% Módulo: winontop.m
% Descrição: Permite janela sobrepor as demais
% Fonte: código retirado da Internet e adaptado
% Copyright: by John Anderson
%           04 Oct 2005 (Updated 14 Dec 2005)
%           No BSD License
% Download em 19/07/2010 feito de
% http://www.mathworks.com/matlabcentral/fileexchange/8642
% *****

%WINONTOP
% WINONTOP(FT) set "topmost" property of the figure specified by
% title FT to state "on".
%
% WINONTOP(FT,1) same as WINONTOP(FT)
%
% WINONTOP(FT,0) set "topmost" property of the figure specified by
% title FT to state "off".

% Modificações:
% A função original só funciona para figuras com nome padrão
% como 'Figure 1', 'Figure 2', etc.
% A função (winontop.c) foi alterada para aceitar qualquer nome.
% A figura deve ser criada com os seguintes parâmetros:
%     h = figure('name','Nome da Figura','NumberTitle','off');
% e a chamada da função deve ser feita assim:
%     winontop('Nome da Figura',1)
% Para recompilar o arquivo winontop.mex32 basta digitar 'mex winontop.c'
```

20. winontop.mexw32

Este arquivo é uma DLL (*Dynamic Linked Library*) que foi criada para armazenar a função winontop criada em Linguagem C.

Para gerar o arquivo winontop.mex32 basta digitar 'mex winontop.c' na área de comando do MATLAB.

A extensão "mexw32" é específica do MATLAB.

21. Dados.xls

Este é o arquivo Excel que contém as condições do experimento e os dados experimentais (Figuras 135 e 136). Maiores detalhes podem ser vistos no Manual do TrataDados.

19. winontop.m

```
% *****
% Módulo: winontop.m
% Descrição: Permite janela sobrepor as demais
% Fonte: código retirado da Internet e adaptado
% Copyright: by John Anderson
%           04 Oct 2005 (Updated 14 Dec 2005)
%           No BSD License
% Download em 19/07/2010 feito de
% http://www.mathworks.com/matlabcentral/fileexchange/8642
% *****

%WINONTOP
% WINONTOP(FT) set "topmost" property of the figure specified by
% title FT to state "on".
%
% WINONTOP(FT,1) same as WINONTOP(FT)
%
% WINONTOP(FT,0) set "topmost" property of the figure specified by
% title FT to state "off".

% Modificações:
% A função original só funciona para figuras com nome padrão
% como 'Figure 1', 'Figure 2', etc.
% A função (winontop.c) foi alterada para aceitar qualquer nome.
% A figura deve ser criada com os seguintes parâmetros:
%     h = figure('name','Nome da Figura','NumberTitle','off');
% e a chamada da função deve ser feita assim:
%     winontop('Nome da Figura',1)
% Para recompilar o arquivo winontop.mex32 basta digitar 'mex winontop.c'
```

20. winontop.mexw32

Este arquivo é uma DLL (*Dynamic Linked Library*) que foi criada para armazenar a função winontop criada em Linguagem C.

Para gerar o arquivo winontop.mex32 basta digitar 'mex winontop.c' na área de comando do MATLAB.

A extensão "mexw32" é específica do MATLAB.

21. Dados.xls

Este é o arquivo Excel que contém as condições do experimento e os dados experimentais (Figuras 135 e 136). Maiores detalhes podem ser vistos no Manual do TrataDados.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																			
2																			
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29																			
30																			
31																			
32																			
33																			
34																			
35																			

Figura 135 - Planilha excel com as condições de ensaio.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	INSTRUÇÕES													
2	1) Inserir comentários apenas nas linhas de 1 a 10. Não inserir células com valores numéricos.													
3	2) Não alterar a ordem das colunas													
4	3) Indicar dados inexistentes com o valor -1 (pontos serão ignorados no cálculo)													
5	4) Não alterar células que possuem fórmulas													
6	5) Colocar os dados apenas nesta planilha													
7	6) Não trocar o nome das planilhas (guias)													
8	7) Não alterar as planilhas geradas automaticamente pelo MATLAB													
9	8) O NOME DO ENSAIO deve estar obrigatoriamente na célula C12													
10	9) O NOME DO PRODUTO deve estar obrigatoriamente na célula G12													
11														
12	NOME DO ENSAIO		XPTO	NOME DO PRODUTO			GPV							
13														
14			Concentração celular (cel/mL)											
15	Amostra	Tempo	X (cel/mL)			Viab	GLC	GLN	GLU	LAC	NH ₄	Elisa (GPV)	GPV _E	GPV
16		(d)	X _{VIÁVEL}	X _{MORTA}	X _{TOTAL}	(%)	(g/L)	(mg/L)	(mg/L)	(g/L)	(mg/L)	(ng/10 ⁶ cel)	(ng/10 ⁷ cel)	(ng/mL)
17	0	0,00	7,93E+05	6,25E+03	7,99E+05	99,2	9,81	3012,00	-1,00	0,01943	12,50	48,34	483,40	38,31
18	1	1,06	1,36E+06	3,25E+04	1,39E+06	97,7	9,86	2744,00	-1,00	0,01943	23,90	69,33	693,30	93,94
19	2	1,84	2,44E+06	6,00E+04	2,50E+06	97,6	-1,00	-1,00	-1,00	-1,00000	42,14	-1,00	-1,00	-1,00
20	3	2,13	3,19E+06	1,20E+05	3,31E+06	96,4	8,94	2140,00	-1,00	0,01943	40,47	117,34	1173,40	374,31
21	4	2,72	4,91E+06	1,65E+05	5,08E+06	96,8	-1,00	-1,00	-1,00	-1,00000	60,69	-1,00	-1,00	-1,00
22	5	3,10	7,17E+06	2,60E+05	7,43E+06	96,5	7,41	986,00	-1,00	0,08650	53,74	118,03	1180,30	846,28
23	6	3,88	1,09E+07	2,06E+05	1,11E+07	98,1	7,17	690,00	-1,00	0,14870	77,40	91,66	916,60	1000,70
24	7	4,09	1,12E+07	3,19E+05	1,15E+07	97,2	6,77	634,00	-1,00	0,15716	71,37	112,17	1121,70	1251,40

	O	P	Q	R	S	T	U	V	W	X	Y	Z
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14			Concentração celular (cel/mL)									
15	Tempo	X (cel/mL)			Viab	GLC	GLN	GLU	LAC	NH ₄	GPV	GPV _E
16	(d)	X _{VIÁVEL}	X _{MORTA}	X _{TOTAL}	(%)	(g/L)	(g/L)	(g/L)	(g/L)	(g/L)	(ug/L)	(ug/10 ⁷ cel)
17	0,00	7,93E+05	6,25E+03	7,99E+05	99,2	9,81	3,01	-1,00	0,01943	0,01250	38,31	483,40
18	1,06	1,36E+06	3,25E+04	1,39E+06	97,7	9,86	2,74	-1,00	0,01943	0,02390	93,94	693,30
19	1,84	2,44E+06	6,00E+04	2,50E+06	97,6	-1,00	-1,00	-1,00	-1,00000	0,04214	-1,00	-1,00
20	2,13	3,19E+06	1,20E+05	3,31E+06	96,4	8,94	2,14	-1,00	0,01943	0,04047	374,31	1173,40
21	2,72	4,91E+06	1,65E+05	5,08E+06	96,8	-1,00	-1,00	-1,00	-1,00000	0,06069	-1,00	-1,00
22	3,10	7,17E+06	2,60E+05	7,43E+06	96,5	7,41	0,99	-1,00	0,08650	0,05374	846,28	1180,30
23	3,88	1,09E+07	2,06E+05	1,11E+07	98,1	7,17	0,69	-1,00	0,14870	0,07740	1000,70	916,60
24	4,09	1,12E+07	3,19E+05	1,15E+07	97,2	6,77	0,63	-1,00	0,15716	0,07137	1251,40	1121,70

Figura 136 - Planilha Excel com os dados experimentais.

APÊNDICE F – LISTAGEM DO SOFTWARE AjustaPar

1. Arquivos

Este software é composto pelos seguintes arquivos:

- AjustaPar.m
- Dados.m
- Dados.xls
- equacoes.m
- funcao.m
- Graficos.m
- Graficos2.m
- poliedros.m

A listagem de cada um desses arquivos é mostrada a seguir.

2. AjustaPar.m

```
% *****
% Módulo: AjustaPar.m
% Descrição: Faz ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

function AjustaPar

% Variáveis globais
clear all
global formula normaliza inicial inferior superior
global Residuo Residuo_Anterior epsilon FatorPrecisao ODE

% Inicializa resíduo
Residuo = realmax; % realmax = 1.7977e+308
FatorPrecisao = 1E6;

% Escolha do método de otimização
fprintf('\n\nEscolha o método de otimização:\n');
fprintf('1 = Minimização com restrição (fmincon)\n');
fprintf('2 = Poliedros Flexíveis sem restrição (fminsearch)\n');
fprintf('3 = Poliedros Flexíveis com restrição (pamboukian)\n');
fprintf('4 = Minimização sem restrição (fminunc)\n');
metodo = input('Qual sua opção? (ENTER = 1) ');
if (isempty(metodo))
    metodo = 1;
end
fprintf('\n');
if (metodo==1) % fmincon
    AlgTipo = input('Tipo (1 = active-set, 2 = interior-point, 3 = SQP)? (ENTER = interior-point) ');
```

```

if (isempty(AlgTipo) || AlgTipo == 2)
    AlgTipo = 'interior-point';
elseif (AlgTipo == 1)
    AlgTipo = 'active-set';
else
    AlgTipo = 'sqp';
end
MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
if (isempty(MaxIteracoes))
    MaxIteracoes = 10000;
end
MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
if (isempty(MaxAvaliacoes))
    MaxAvaliacoes = 100000;
end
Precisao = input('Precisão? (ENTER = 1E-6) ');
if (isempty(Precisao))
    Precisao = 1E-6;
end
elseif (metodo==2) % fminsearch
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==3) % poliedros (livro)
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    Precisao = input('Precisão? (ENTER = 1E-8) ');
    if (isempty(Precisao))
        Precisao = 1E-8;
    end
    Passo = input('Passo? (ENTER = 0.3) ');
    if (isempty(Passo))
        Passo = 0.3;
    end
    % Livro "Computer-aided design of communication networks":
    % Alfa=1.0, Gama=2.0, Beta=0.5, Delta=0.5.
    % Original: Alfa=1.0, Gama=2, Beta=0.5, Delta=0.5.
    % Bonomi: Alfa=1.0, Gama=1.5, Beta=0.8, Delta=0.5.
    Alfa = input('Coeficiente de Reflexão (Alfa)? (ENTER = 1.0) ');
    if (isempty(Alfa))
        Alfa = 1.0;
    end
    Gama = input('Coeficiente de Expansão (Gama)? (ENTER = 1.5) ');
    if (isempty(Gama))
        Gama = 1.5;
    end
    Beta = input('Coeficiente de Contração (Beta)? (ENTER = 0.8) ');
    if (isempty(Beta))
        Beta = 0.8;
    end
    Delta = input('Coeficiente de Redução (Delta)? (ENTER = 0.5) ');
    if (isempty(Delta))
        Delta = 0.5;
    end
elseif (metodo==4) %fminunc
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
else
    fprintf('Opção inválida!\n\n');
    return;
end
% Escolha do método para o cálculo dos resíduos
fprintf('\n\nEscolha o método para o cálculo dos resíduos:\n');
fprintf('1 = Fórmula 1\n');
fprintf('2 = Fórmula 2\n');
fprintf('3 = Fórmula 3\n');
fprintf('4 = Fórmula 4\n');

```

```

if (isempty(AlgTipo) || AlgTipo == 2)
    AlgTipo = 'interior-point';
elseif (AlgTipo == 1)
    AlgTipo = 'active-set';
else
    AlgTipo = 'sqp';
end
MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
if (isempty(MaxIteracoes))
    MaxIteracoes = 10000;
end
MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
if (isempty(MaxAvaliacoes))
    MaxAvaliacoes = 100000;
end
Precisao = input('Precisão? (ENTER = 1E-6) ');
if (isempty(Precisao))
    Precisao = 1E-6;
end
elseif (metodo==2) % fminsearch
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==3) % poliedros (livro)
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    Precisao = input('Precisão? (ENTER = 1E-8) ');
    if (isempty(Precisao))
        Precisao = 1E-8;
    end
    Passo = input('Passo? (ENTER = 0.3) ');
    if (isempty(Passo))
        Passo = 0.3;
    end
    % Livro "Computer-aided design of communication networks":
    % Alfa=1.0, Gama=2.0, Beta=0.5, Delta=0.5.
    % Original: Alfa=1.0, Gama=2, Beta=0.5, Delta=0.5.
    % Bonomi: Alfa=1.0, Gama=1.5, Beta=0.8, Delta=0.5.
    Alfa = input('Coeficiente de Reflexão (Alfa)? (ENTER = 1.0) ');
    if (isempty(Alfa))
        Alfa = 1.0;
    end
    Gama = input('Coeficiente de Expansão (Gama)? (ENTER = 1.5) ');
    if (isempty(Gama))
        Gama = 1.5;
    end
    Beta = input('Coeficiente de Contração (Beta)? (ENTER = 0.8) ');
    if (isempty(Beta))
        Beta = 0.8;
    end
    Delta = input('Coeficiente de Redução (Delta)? (ENTER = 0.5) ');
    if (isempty(Delta))
        Delta = 0.5;
    end
elseif (metodo==4) %fminunc
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
else
    fprintf('Opção inválida!\n\n');
    return;
end
% Escolha do método para o cálculo dos resíduos
fprintf('\n\nEscolha o método para o cálculo dos resíduos:\n');
fprintf('1 = Fórmula 1\n');
fprintf('2 = Fórmula 2\n');
fprintf('3 = Fórmula 3\n');
fprintf('4 = Fórmula 4\n');

```



```

% Calculo dos residuo final (sem normalizar as variáveis)
normaliza = 1;
Residuo = funcao(x)/FatorPrecisao;
fprintf('Residuo = %f\n',Residuo);

tElapsed = toc(tStart);
fprintf('Tempo = %d segundos\n', round(tElapsed));

% Exibe resultados
k = x;
fprintf('\nParâmetros calculados:\n');
fprintf('k = [ ');
for i=1:length(k)
    fprintf('%4e ',k(i));
    if (mod(i,5)==0)
        fprintf('...\n      ');
    end
end
fprintf(']\n\n');

% Construção dos gráficos
Graficos;

% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');

end

```

3. Dados.m

```

% *****
% Script: Dados.m
% Descrição: Valores experimentais e limites para o ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global a1 a2 Equacoes Titulos inicial inferior superior pontosEnsaio ensaiosAjuste

fprintf('\n\nAguarde! Lendo os dados.\n');

% Lê titulos
[tmp1 Titulos] = xlsread('Dados.xls','Alisados','B1:Z1');

% Lê valores alisados
a1 = xlsread('Dados.xls','Alisados');

% Lê valores experimentais
a2 = xlsread('Dados.xls','Experimentais');

% pontos inicial e final de cada ensaio
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A2:A2');
eval(cell2mat(tmp2));

% quantidade de ensaios
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A5:A5');
eval(cell2mat(tmp2));

% Valores iniciais dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A8:A8');
eval(cell2mat(tmp2));

% Limite inferior dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A11:A11');
eval(cell2mat(tmp2));

% Limite superior dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A14:A14');
eval(cell2mat(tmp2));

% Equações
[tmp1 Equacoes] = xlsread('Dados.xls','Equacoes');

```

```

% Calculo dos residuo final (sem normalizar as variáveis)
normaliza = 1;
Residuo = funcao(x)/FatorPrecisao;
fprintf('Residuo = %f\n',Residuo);

tElapsed = toc(tStart);
fprintf('Tempo = %d segundos\n', round(tElapsed));

% Exibe resultados
k = x;
fprintf('\nParâmetros calculados:\n');
fprintf('k = [ ');
for i=1:length(k)
    fprintf('%4e ',k(i));
    if (mod(i,5)==0)
        fprintf('...\n      ');
    end
end
fprintf(']\n\n');

% Construção dos gráficos
Graficos;

% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');

end

```

3. Dados.m

```

% *****
% Script: Dados.m
% Descrição: Valores experimentais e limites para o ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global a1 a2 Equacoes Titulos inicial inferior superior pontosEnsaio ensaiosAjuste

fprintf('\n\nAguarde! Lendo os dados.\n');

% Lê titulos
[tmp1 Titulos] = xlsread('Dados.xls','Alisados','B1:Z1');

% Lê valores alisados
a1 = xlsread('Dados.xls','Alisados');

% Lê valores experimentais
a2 = xlsread('Dados.xls','Experimentais');

% pontos inicial e final de cada ensaio
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A2:A2');
eval(cell2mat(tmp2));

% quantidade de ensaios
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A5:A5');
eval(cell2mat(tmp2));

% Valores iniciais dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A8:A8');
eval(cell2mat(tmp2));

% Limite inferior dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A11:A11');
eval(cell2mat(tmp2));

% Limite superior dos parâmetros
[tmp1 tmp2] = xlsread('Dados.xls','Ajuste','A14:A14');
eval(cell2mat(tmp2));

% Equações
[tmp1 Equacoes] = xlsread('Dados.xls','Equacoes');

```

```

    a2 = ali;
end
if (normaliza==2) % Normalização por coluna juntando as tabelas
    juntas = [ali;res1]; % Linhas de ali seguidas das linhas de res1
    norm = normc(juntas); % Normaliza colunas
    tam_ali = size(ali);
    a2 = norm(1:tam_ali(1),:); % a2 recebe da linha 1 ao tamanho de ali
    res2 = norm((tam_ali(1)+1):(tam_ali(1)*2),:); % res2 recebe as linhas restantes
end

% Calculo do residuo
if (formula==1)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + (res2(i,j)-a2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==2)
    max_a2 = max(a2); % maior valor experimental com normalização
    max_res2 = max(res2); % maior valor calculado com normalização
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + (res2(i,j)/max_res2(j)-a2(i,j)/max_a2(j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==3)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./res2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==4)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./a2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==5)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==6)
    max_ali = max(ali); % maior valor experimental sem normalização
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j) > epsilon * max_ali(j)) % não precisa verificar se (ali(i,j)~= -1)
                s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==7)
    tam = size(a2);

```

```

    a2 = ali;
end
if (normaliza==2) % Normalização por coluna juntando as tabelas
    juntas = [ali;res1]; % Linhas de ali seguidas das linhas de res1
    norm = normc(juntas); % Normaliza colunas
    tam_ali = size(ali);
    a2 = norm(1:tam_ali(1),:); % a2 recebe da linha 1 ao tamanho de ali
    res2 = norm((tam_ali(1)+1):(tam_ali(1)*2),:); % res2 recebe as linhas restantes
end

% Calculo do residuo
if (formula==1)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + (res2(i,j)-a2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==2)
    max_a2 = max(a2); % maior valor experimental com normalização
    max_res2 = max(res2); % maior valor calculado com normalização
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + (res2(i,j)/max_res2(j)-a2(i,j)/max_a2(j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==3)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./res2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==4)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./a2(i,j)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==5)
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j)~= -1) % existe valor experimental
                s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==6)
    max_ali = max(ali); % maior valor experimental sem normalização
    s = 0;
    tam = size(a2);
    for i=1:tam(1)
        for j=1:tam(2)
            if (ali(i,j) > epsilon * max_ali(j)) % não precisa verificar se (ali(i,j)~= -1)
                s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
            end
        end
    end
    erro = s;
end
if (formula==7)
    tam = size(a2);
end

```

```

sr = 0; sa = 0;
for j=1:tam(2)
    % retira pontos inexistentes (experimental=-1)
    aux_a2 = [];
    aux_res2 = [];
    cnt=1;
    for i=1:tam(1)
        if (ali(i,j)~= -1)
            aux_a2(cnt) = a2(i,j);
            aux_res2(cnt) = res2(i,j);
            cnt = cnt + 1;
        end
    end
    % Calcula coeficientes
    % calcula coeficiente de correlação
    C = corrccoef(aux_res2,aux_a2); r = abs(C(1,2)); sr = sr + abs(r-1);
    % calcula coeficiente angular
    P = polyfit(aux_res2,aux_a2,1); a = P(1); sa = sa + abs(a-1);
end
erro = sr + sa;
end
if (formula==8) % obrigatório valores normalizados
    tam = size(a2);
    % forma vetor coluna com todos os valores
    res3=[]; a3 = [];
    for j=1:tam(2)
        % retira pontos inexistentes (experimental=-1)
        aux_a2 = [];
        aux_res2 = [];
        cnt=1;
        for i=1:tam(1)
            if (ali(i,j)~= -1)
                aux_a2(cnt) = a2(i,j);
                aux_res2(cnt) = res2(i,j);
                cnt = cnt + 1;
            end
        end
        % junta colunas
        res3 = [ res3; aux_res2' ]; %ok<AGROW>
        a3 = [ a3; aux_a2' ]; %ok<AGROW>
    end
    C = corrccoef(res3,a3); r = abs(C(1,2)); % calcula coeficiente de correlação
    P = polyfit(res3,a3,1); a = P(1); % calcula coeficiente angular
    erro = abs(r-1) + abs(a-1);
end
ErroTotal = ErroTotal + erro;
end

Residuo_Anterior = Residuo;
Residuo = ErroTotal;

ErroTotal = ErroTotal * FatorPrecisao;

end

```

6. Graficos.m

```

% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

function Graficos
% Construção dos gráficos
global a2 res1 ensaiosAjuste pontosEnsaio Ensaio
a2aux = a2;
for Ensaio=ensaiosAjuste
    % Copia para a1 apenas um ensaio de alaux
    a2 = a2aux(pontosEnsaio(Ensaio,1):pontosEnsaio(Ensaio,2),:);
    % Condições iniciais
    tam = size(a2); % tamanho (linhas,colunas) de ali
    y0 = a2(1,2:tam(2)); % primeira linha da coluna 2 em diante
    % Valores de tempo
    tspan = a2(:,1); % primeira coluna

```

```

% Resolve Equações Diferenciais Ordinárias
[t y] = ode45('equacoes', tspan, y0);
res1=[t y];
% Cria nova figura
figure();
Graficos2;
end
end

```

7. Graficos2.m

```

% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Construção dos gráficos
global a2 res1 Titulos
tam = size(a2);
n = tam(2)-1;
x = ceil(sqrt(n));
y = ceil(n/x);
for i=1:n
    subplot(x,y,i),plot(a2(:,1),a2(:,i+1),'b*',res1(:,1),res1(:,i+1),'b-')
    ylabel(Titulos(i));
    xlabel('tempo (h)');
end
end

```

8. poliedros.m

```

% *****
% Módulo: poliedros.m
% Descrição: Minimização utilizando Poliédro Flexíveis
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% *****
% Implementação do Algoritmo dos POLIÉDROS FLEXÍVEIS
% Baseado na implementação em FORTRAN encontrada em:
% "Computer-aided design of communication networks"
% Yi-Sheng Zhu & Wai-Kai Chen
% Publicado por World Scientific, 2000
% ISBN 981022351X 9789610223519
% *****

% *****
% Sintaxe
% [XF,PRECISAO,QUANTITER,QUANTEVAL] = poliedros(FO,Step,X0,XI,XS,MaxIter,Precision);
% Exemplo de chamada
% [XF,PRECISAO,QUANTITER,QUANTEVAL] = poliedros(@funcao,0.3,[1 1 1 1],[0 0 0 0],[2 2 2 2],2000,1e-8);
% *****

% *****
% Modificações feitas por Marilena
% 1) Verificar se os valores sorteados estão dentro dos limites (inferior/superior)
% 2) Contar quantas vezes a FO é avaliada (QUANTEVAL)
% 3) Contar a Quantidade de Iterações feitas (QUANTITER)
% 4) Exibir a matriz inicial
% 5) Permitir a passagem dos coeficientes como parâmetros
% *****

```

```

function [XF,PRECISAO,QUANTITER,QUANTEVAL] =
poliedros(FO,Step0,X0,XI,XS,MaxIter,Precision,Alfa,Gama,Beta,Delta)
% Quantidade de avaliações da FO
QUANTEVAL = 0; % Linha acrescentada por Marilena
% Nx is the number of independent variables
Nx = length(X0);
% Step is the initial step size
Step = Step0;
% Inicializa vetores e matrizes
X = zeros(Nx,1);
X1 = zeros(Nx,Nx);
Sum = zeros(Nx,1);
% X(I) is the array of initial guesses
X = X0;
% Inicializa variáveis
XNx = Nx;
IN = 1;
% Somatória inicial
Sum(IN) = feval(FO,X);
QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
% Call second(time)
K1 = Nx + 1;
K2 = Nx + 2;
K3 = Nx + 3;
K4 = Nx + 4;
% Start
A = zeros(Nx,Nx);
Vn = Nx;
Step1 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)+Vn-1.0);
Step2 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)-1.0);
for J=1:Nx
    A(1,J) = 0.0;
end
for I=2:K1
    for J=1:Nx
        A(I,J) = Step2;
        L = I - 1;
        A(I,L) = Step1;
    end
end
for I=1:K1
    for J=1:Nx
        X1(I,J) = X(J)+A(I,J);
        % *****
        % Código acrescentado por Marilena
        % *****
        if (X1(I,J) < XI(J))
            X1(I,J) = XI(J);
        end
        if (X1(I,J) > XS(J))
            X1(I,J) = XS(J);
        end
        % *****
    end
end
for I=1:K1
    for J=1:Nx
        X(J) = X1(I,J);
        IN = I;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    end
end
end
% *****
% Código acrescentado por Marilena
% *****
% Exibe matriz inicial
fprintf('\n\nMatriz inicial\n');
tamX1 = size(X1);
for I=1:tamX1(1)
    for J=1:tamX1(2)
        fprintf('%.4e ',X1(I,J));
    end
    fprintf('\n');
end
fprintf('\n\nIterações\n');
% *****

% Inicia looping
Difer = realmax;
IJK = 1;
while (Difer > Precision && IJK <= MaxIter)
    % Select largest Value of Sum(I) in Simplex
    Sumh = Sum(1);
    Index = 1;
    for I=2:K1
        if (Sum(I) >= Sumh)

```

```

else
    Sums = Sum(2);
end
for I=1:K1
    if ((Index-1 ~= 0) && (Sum(I)>=Sums))
        Sums = Sum(I);
    end
end
if (Sum(K3) > Sums)
    if (Sum(K3) <= Sumh)
        for J=1:Nx
            X1(Index,J) = X1(K3,J);
        end
    end
    for J=1:Nx
        X1(K4,J) = Beta .* X1(Index,J) + (1.0-Beta) .* X1(K2,J);
        % *****
        % Código acrescentado por Marilena
        % *****
        if (X1(K4,J) < XI(J))
            X1(K4,J) = XI(J);
        end
        if (X1(K4,J) > XS(J))
            X1(K4,J) = XS(J);
        end
        % *****
        X(J) = X1(K4,J);
    end
    IN = K4;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    if (Sumh > Sum(K4))
        for J=1:Nx
            X1(Index,J) = X1(K4,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    else
        for J=1:Nx
            for I=1:K1
                X1(I,J) = Delta .* (X1(I,J) + X1(Kount,J));
                % *****
                % Código acrescentado por Marilena
                % *****
                if (X1(I,J) < XI(J))
                    X1(I,J) = XI(J);
                end
                if (X1(I,J) > XS(J))
                    X1(I,J) = XS(J);
                end
                % *****
            end
        end
        for I=1:K1
            for J=1:Nx
                X(J) = X1(I,J);
            end
            IN = I;
            Sum(IN) = feval(FO,X);
            QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
        end
    end
else
    for J=1:Nx
        X1(Index,J) = X1(K3,J);
        X(J) = X1(Index,J);
    end
    IN = Index;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
end
end
for J=1:Nx
    X(J) = X1(K2,J);
end
IN = K2;
Sum(IN) = feval(FO,X);
QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
Difer = 0.0;
for I=1:K1
    Difer = Difer + (Sum(I)-Sum(K2)).^2;
end
Difer = 1.0 ./ XNx .* sqrt(Difer);
IJK = IJK + 1;
fprintf('%d\t%.8e\n',IJK-1,Difer);
PRECISAO = Difer;

```



```

else
    Sums = Sum(2);
end
for I=1:K1
    if ((Index-1 ~= 0) && (Sum(I)>=Sums))
        Sums = Sum(I);
    end
end
if (Sum(K3) > Sums)
    if (Sum(K3) <= Sumh)
        for J=1:Nx
            X1(Index,J) = X1(K3,J);
        end
    end
    for J=1:Nx
        X1(K4,J) = Beta .* X1(Index,J) + (1.0-Beta) .* X1(K2,J);
        % *****
        % Código acrescentado por Marilena
        % *****
        if (X1(K4,J) < XI(J))
            X1(K4,J) = XI(J);
        end
        if (X1(K4,J) > XS(J))
            X1(K4,J) = XS(J);
        end
        % *****
        X(J) = X1(K4,J);
    end
    IN = K4;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    if (Sumh > Sum(K4))
        for J=1:Nx
            X1(Index,J) = X1(K4,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    else
        for J=1:Nx
            for I=1:K1
                X1(I,J) = Delta .* (X1(I,J) + X1(Kount,J));
                % *****
                % Código acrescentado por Marilena
                % *****
                if (X1(I,J) < XI(J))
                    X1(I,J) = XI(J);
                end
                if (X1(I,J) > XS(J))
                    X1(I,J) = XS(J);
                end
                % *****
            end
        end
        for I=1:K1
            for J=1:Nx
                X(J) = X1(I,J);
            end
            IN = I;
            Sum(IN) = feval(FO,X);
            QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
        end
    end
else
    for J=1:Nx
        X1(Index,J) = X1(K3,J);
        X(J) = X1(Index,J);
    end
    IN = Index;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
end
end
for J=1:Nx
    X(J) = X1(K2,J);
end
IN = K2;
Sum(IN) = feval(FO,X);
QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
Difer = 0.0;
for I=1:K1
    Difer = Difer + (Sum(I)-Sum(K2)).^2;
end
Difer = 1.0 ./ XNx .* sqrt(Difer);
IJK = IJK + 1;
fprintf('%d\t%.8e\n',IJK-1,Difer);
PRECISAO = Difer;

```

```

    if (IJK > MaxIter)
        disp('Quantidade maxima de iteracoes excedida!');
    end
end %while
XF = zeros(Nx,1);
for J=1:Nx
    X(J) = Xl(Kount,J);
    XF(J) = X(J);
end
QUANTITER = IJK-1; % Linha acrescentada por Marilena
end

```

9. Dados.xls

Este é o arquivo Excel que contém os dados alisados, os dados experimentais, as condições de ajuste e as equações cinéticas do processo (Figuras 137 a 140). Maiores detalhes podem ser vistos no Manual do AjustaPar.

	A	B	C	D	E	F	G
1	Tempo (h)	1E-06.Xv (cel/mL)	GLC (g/L)	GLN (mg/L)	P (ug/mL)	LAC (g/L)	NH4 (mg/L)
2	0,1	1,01E-01	4,50E+00	6,00E+02	1,51E+00	1,51E-01	5,03E+00
3	10,1	1,65E-01	4,34E+00	5,46E+02	2,57E+00	2,24E-01	8,38E+00
4	20,1	2,64E-01	4,09E+00	4,63E+02	4,25E+00	3,36E-01	1,35E+01
5	30,1	4,07E-01	3,73E+00	3,43E+02	6,79E+00	4,99E-01	2,10E+01
6	40,1	6,02E-01	3,24E+00	1,81E+02	1,04E+01	7,21E-01	3,11E+01
7	50,1	8,18E-01	2,70E+00	6,08E-03	1,51E+01	9,67E-01	4,23E+01
8	60,1	8,18E-01	2,70E+00	1,00E-07	1,83E+01	9,70E-01	4,23E+01
9	70,1	8,18E-01	2,70E+00	1,00E-07	2,14E+01	9,72E-01	4,24E+01
10							

Figura 137 - Dados alisados de cada ensaio.

	A	B	C	D	E	F	G
1	Tempo (h)	1E-06.Xv (cel/mL)	GLC (g/L)	GLN (mg/L)	P (ug/mL)	LAC (g/L)	NH4 (mg/L)
2	0,1	1,01E-01	4,50E+00	6,00E+02	1,51E+00	1,51E-01	5,03E+00
3	10,1	1,65E-01	4,34E+00	5,46E+02	2,57E+00	2,24E-01	8,38E+00
4	20,1	2,64E-01	4,09E+00	4,63E+02	4,25E+00	3,36E-01	1,35E+01
5	30,1	4,07E-01	3,73E+00	3,43E+02	6,79E+00	4,99E-01	2,10E+01
6	40,1	6,02E-01	3,24E+00	1,81E+02	1,04E+01	7,21E-01	3,11E+01
7	50,1	8,18E-01	2,70E+00	6,08E-03	1,51E+01	9,67E-01	4,23E+01
8	60,1	8,18E-01	2,70E+00	1,00E-07	1,83E+01	9,70E-01	4,23E+01
9	70,1	8,18E-01	2,70E+00	1,00E-07	2,14E+01	9,72E-01	4,24E+01
10							

Figura 138 - Dados experimentais de cada ensaio.

```

A
1 % Pontos inicial e final de cada ensaio - pontosEnsaio = [ i1 f1; i2 f2; i3 f3; ... ];
2 pontosEnsaio = [ 1 8 ];
3
4 % Quantidade de ensaios - ensaiosAjuste = [ e1 e2 e3 ... ];
5 ensaiosAjuste = [ 1 ];
6
7 % Valores iniciais dos parâmetros
8 inicial = [ 0.035 2.55 150 2.9 0.45 0.65 0.0019 0.020 8.57 0.379 1.03 0.302E-03 51 0.49E-03 0.51 6.95 ]';
9
10 % Limite inferior dos parâmetros
11 inferior = [ 0.035 2.55 150 2.9 0.45 0.65 0.0019 0.020 8.57 0.379 1.03 0.302E-03 51 0.49E-03 0.51 6.95 ]';
12
13 % Limite superior dos parâmetros
14 superior = [ 0.085 2.75 170 3.1 0.65 0.85 0.0021 0.022 8.77 0.399 1.23 0.322E-03 53 0.51E-03 0.53 7.15 ]';
15

```

Figura 139 - Informações de ajuste.

```

A1  fx  MIX = k(1)*(y(3)/(y(3)+k(2)))*(k(3)/(k(3)+y(6)))*(k(4)/(k(4)+y(5)));
A
1 MIX = k(1)*(y(3)/(y(3)+k(2)))*(k(3)/(k(3)+y(6)))*(k(4)/(k(4)+y(5)));
2 resp(1) = MIX*y(1);
3 resp(2) = -(1/k(5))*MIX*y(1)-(1/k(6))*k(15)*MIX*y(1);
4 resp(3) = -(1/k(7))*MIX*y(1)-(1/k(8))*k(16)*MIX*y(1);
5 resp(4) = k(9)*MIX*y(1)+k(10)*y(1);
6 resp(5) = k(11)*MIX*y(1)+k(12)*y(1);
7 resp(6) = k(13)*MIX*y(1)+k(14)*y(1);
8

```

Figura 140 - Equações cinéticas.

APÊNDICE G – LISTAGEM GENÉRICA DO SOFTWARE AjustaPar UTILIZADO NO MODELO 5

1. Arquivos

Este software é composto pelos seguintes arquivos:

- AjustaPar.m
- Dados.m
- Dados2.m
- equacoes.m
- funcao.m
- Graficos.m
- Graficos2.m
- poliedros.m
- ValoresGrafico.m

A listagem de cada um desses arquivos é mostrada a seguir.

2. AjustaPar.m

```
% *****
% Módulo: AjustaPar.m
% Descrição: Faz ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

function AjustaPar

% Variáveis globais
clear all
global formula normaliza inicial inferior superior
global Residuo Residuo_Anterior epsilon FatorPrecisao ODE

% Inicializa resíduo
Residuo = realmax; % realmax = 1.7977e+308
FatorPrecisao = 1E6;

% Escolha do método de otimização
fprintf('\n\nEscolha o método de otimização:\n');
fprintf('1 = Minimização com restrição (fmincon)\n');
fprintf('2 = Poliedros Flexíveis sem restrição (fminsearch)\n');
fprintf('3 = Poliedros Flexíveis com restrição (pamboukian)\n');
fprintf('4 = Minimização sem restrição (fminunc)\n');
metodo = input('Qual sua opção? (ENTER = 1) ');
if (isempty(metodo))
```

```

    metodo = 1;
end
fprintf('\n');
if (metodo==1) % fmincon
    AlgTipo = input('Tipo (1 = active-set, 2 = interior-point, 3 = SQP)? (ENTER = interior-point) ');
    if (isempty(AlgTipo) || AlgTipo == 2)
        AlgTipo = 'interior-point';
    elseif (AlgTipo == 1)
        AlgTipo = 'active-set';
    else
        AlgTipo = 'sqp';
    end
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==2) % fminsearch
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==3) % poliedros (livro)
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    Precisao = input('Precisão? (ENTER = 1E-8) ');
    if (isempty(Precisao))
        Precisao = 1E-8;
    end
    Passo = input('Passo? (ENTER = 0.3) ');
    if (isempty(Passo))
        Passo = 0.3;
    end
    % Livro "Computer-aided design of communication networks":
    %     Alfa=1.0, Gama=2.0, Beta=0.5, Delta=0.5.
    % Original: Alfa=1.0, Gama=2, Beta=0.5, Delta=0.5.
    % Bonomi: Alfa=1.0, Gama=1.5, Beta=0.8, Delta=0.5.
    Alfa = input('Coeficiente de Reflexão (Alfa)? (ENTER = 1.0) ');
    if (isempty(Alfa))
        Alfa = 1.0;
    end
    Gama = input('Coeficiente de Expansão (Gama)? (ENTER = 1.5) ');
    if (isempty(Gama))
        Gama = 1.5;
    end
    Beta = input('Coeficiente de Contração (Beta)? (ENTER = 0.8) ');
    if (isempty(Beta))
        Beta = 0.8;
    end
    Delta = input('Coeficiente de Redução (Delta)? (ENTER = 0.5) ');
    if (isempty(Delta))
        Delta = 0.5;
    end
elseif (metodo==4) %fminunc
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
end
else
    fprintf('Opção inválida!\n\n');
    return;
end
% Escolha do método para o cálculo dos resíduos

```

```

    metodo = 1;
end
fprintf('\n');
if (metodo==1) % fmincon
    AlgTipo = input('Tipo (1 = active-set, 2 = interior-point, 3 = SQP)? (ENTER = interior-point) ');
    if (isempty(AlgTipo) || AlgTipo == 2)
        AlgTipo = 'interior-point';
    elseif (AlgTipo == 1)
        AlgTipo = 'active-set';
    else
        AlgTipo = 'sqp';
    end
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==2) % fminsearch
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
elseif (metodo==3) % poliedros (livro)
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    Precisao = input('Precisão? (ENTER = 1E-8) ');
    if (isempty(Precisao))
        Precisao = 1E-8;
    end
    Passo = input('Passo? (ENTER = 0.3) ');
    if (isempty(Passo))
        Passo = 0.3;
    end
    % Livro "Computer-aided design of communication networks":
    %     Alfa=1.0, Gama=2.0, Beta=0.5, Delta=0.5.
    % Original: Alfa=1.0, Gama=2, Beta=0.5, Delta=0.5.
    % Bonomi: Alfa=1.0, Gama=1.5, Beta=0.8, Delta=0.5.
    Alfa = input('Coeficiente de Reflexão (Alfa)? (ENTER = 1.0) ');
    if (isempty(Alfa))
        Alfa = 1.0;
    end
    Gama = input('Coeficiente de Expansão (Gama)? (ENTER = 1.5) ');
    if (isempty(Gama))
        Gama = 1.5;
    end
    Beta = input('Coeficiente de Contração (Beta)? (ENTER = 0.8) ');
    if (isempty(Beta))
        Beta = 0.8;
    end
    Delta = input('Coeficiente de Redução (Delta)? (ENTER = 0.5) ');
    if (isempty(Delta))
        Delta = 0.5;
    end
elseif (metodo==4) %fminunc
    MaxIteracoes = input('Quantidade máxima de iterações? (ENTER = 10000) ');
    if (isempty(MaxIteracoes))
        MaxIteracoes = 10000;
    end
    MaxAvaliacoes = input('Quantidade máxima de avaliações da função objetivo? (ENTER = 100000) ');
    if (isempty(MaxAvaliacoes))
        MaxAvaliacoes = 100000;
    end
    Precisao = input('Precisão? (ENTER = 1E-6) ');
    if (isempty(Precisao))
        Precisao = 1E-6;
    end
end
else
    fprintf('Opção inválida!\n\n');
    return;
end
% Escolha do método para o cálculo dos resíduos

```

```

    fprintf('Quantidade de avaliações da função objetivo: %d\n',output.funcCount);
    fprintf('Método matemático utilizado: %s\n',output.algorithm);
    fprintf('Precisão = %e\n',abs(Residuo_Anterior-Residuo));
end

% Calculo dos residuo final (sem normalizar as variáveis)
normaliza = 1;
Residuo = funcao(x)/FatorPrecisao;
fprintf('Residuo = %f\n',Residuo);

tElapsed = toc(tStart);
fprintf('Tempo = %d segundos\n', round(tElapsed));

% Exibe resultados
k = x;
fprintf('\nParâmetros calculados:\n');
fprintf('k = [ ');
for i=1:length(k)
    fprintf('%4e ',k(i));
    if (mod(i,6)==0)
        fprintf(' ...\n ');
    end
end
end
fprintf(']' * 1E-12;\n\n');

% Ler Dados originais
Dados2;
% Construção dos gráficos
Graficos;

% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');
end

```

3. Dados.m

```

% *****
% Script: Dados.m
% Descrição: Valores experimentais alisados e limites para o ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global al inicial inferior superior pontosEnsaio ensaiosAjuste todos_kp

% PARÂMETROS                VARIÁVEIS DE ESTADO
% k(1)= a1                  y(1)= X
% k(2)= a2                  y(2)= GLC
% k(3)= a3                  y(3)= GLN
% k(4)= a4                  y(4)= LAC
% k(5)= a5                  y(5)= NH4
% k(6)= a6                  y(6)= ALA
% k(7)= a7                  y(7)= GLU
% k(8)= a8                  y(8)= GPVe
% k(9)= a9
% ...
% k(19)= a19

% *****
%                               ATENÇÃO
% OS VALORES EXPERIMENTAIS DEVEM COMEÇAR COM O PRIMEIRO PONTO
% DA FASE EXPONENCIAL para t=0
% *****

al = [
% t (h)      Xv (cel/L)      GLC (mM)      GLN (mM)      LAC(mM)      NH4 (mM)      ALA (mM)      GLU (mM)      GPVe (ng/cel)
% Ma-07
22.742      821829638      53.189      26.750      2.1      0.715      5.117      6.026      1.81E-4;
45.485      1473728759      49.726      21.449      -1.000      1.629      6.981      5.738      1.81E-4;
68.227      2686818641      45.664      16.233      -1.000      2.391      9.506      5.405      1.56E-4;
90.970      4876781528      42.325      13.073      2.930      2.550      12.862      5.289      1.33E-4;
113.712     9115856990      36.515      9.893      3.321      2.639      16.430      4.774      1.15E-4;
136.455     14033050892      28.685      5.497      3.617      3.145      19.755      3.938      6.49E-5;
159.197     15833464460      23.592      3.391      4.201      3.503      24.238      3.549      4.07E-5;
181.940     17683238111      15.473      2.801      5.066      3.153      28.647      3.354      4.68E-5;
204.682     19378237572      7.766      2.712      5.838      3.114      31.487      3.171      3.79E-5;
227.425     19772362505      1.801      2.717      6.288      2.541      33.582      3.045      3.58E-5;

```

```

    fprintf('Quantidade de avaliações da função objetivo: %d\n',output.funcCount);
    fprintf('Método matemático utilizado: %s\n',output.algorithm);
    fprintf('Precisão = %e\n',abs(Residuo_Anterior-Residuo));
end

% Calculo dos residuo final (sem normalizar as variáveis)
normaliza = 1;
Residuo = funcao(x)/FatorPrecisao;
fprintf('Residuo = %f\n',Residuo);

tElapsed = toc(tStart);
fprintf('Tempo = %d segundos\n', round(tElapsed));

% Exibe resultados
k = x;
fprintf('\nParâmetros calculados:\n');
fprintf('k = [ ');
for i=1:length(k)
    fprintf('%4e ',k(i));
    if (mod(i,6)==0)
        fprintf(' ...\n ');
    end
end
end
fprintf(']' * 1E-12;\n\n');

% Ler Dados originais
Dados2;
% Construção dos gráficos
Graficos;

% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');
end

```

3. Dados.m

```

% *****
% Script: Dados.m
% Descrição: Valores experimentais alisados e limites para o ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global al inicial inferior superior pontosEnsaio ensaiosAjuste todos_kp

% PARÂMETROS                VARIÁVEIS DE ESTADO
% k(1)= a1                  y(1)= X
% k(2)= a2                  y(2)= GLC
% k(3)= a3                  y(3)= GLN
% k(4)= a4                  y(4)= LAC
% k(5)= a5                  y(5)= NH4
% k(6)= a6                  y(6)= ALA
% k(7)= a7                  y(7)= GLU
% k(8)= a8                  y(8)= GPVe
% k(9)= a9
% ...
% k(19)= a19

% *****
%                               ATENÇÃO
% OS VALORES EXPERIMENTAIS DEVEM COMEÇAR COM O PRIMEIRO PONTO
% DA FASE EXPONENCIAL para t=0
% *****

al = [
% t (h)      Xv (cel/L)      GLC (mM)      GLN (mM)      LAC(mM)      NH4 (mM)      ALA (mM)      GLU (mM)      GPVe (ng/cel)
% Ma-07
22.742      821829638      53.189      26.750      2.1      0.715      5.117      6.026      1.81E-4;
45.485      1473728759      49.726      21.449      -1.000      1.629      6.981      5.738      1.81E-4;
68.227      2686818641      45.664      16.233      -1.000      2.391      9.506      5.405      1.56E-4;
90.970      4876781528      42.325      13.073      2.930      2.550      12.862      5.289      1.33E-4;
113.712     9115856990      36.515      9.893      3.321      2.639      16.430      4.774      1.15E-4;
136.455     14033050892      28.685      5.497      3.617      3.145      19.755      3.938      6.49E-5;
159.197     15833464460      23.592      3.391      4.201      3.503      24.238      3.549      4.07E-5;
181.940     17683238111      15.473      2.801      5.066      3.153      28.647      3.354      4.68E-5;
204.682     19378237572      7.766      2.712      5.838      3.114      31.487      3.171      3.79E-5;
227.425     19772362505      1.801      2.717      6.288      2.541      33.582      3.045      3.58E-5;

```


	250.167	19757452976	0.635	2.780	6.491	5.121	35.836	3.013	3.35E-5;
	272.910	20006149653	0.790	-1	6.030	11.152	-1	-1	3.41E-5;
	295.652	19690487102	0.796	-1	4.779	15.618	-1	-1	3.70E-5;
	318.395	16053701949	0.755	-1	3.028	17.908	-1	-1	-1;
	340.000	8900540637	0.756	2.877	1.160	18.379	48.848	2.96	-1;
%	Ma-08								
	0	592014547	65.674	21.531	0.145	0.550	2.781	4.931	1.74E-04;
	12.586	777880201	62.860	20.946	0.126	0.725	4.176	5.226	1.74E-04;
	25.173	1001601044	59.892	20.070	0.131	0.935	5.487	5.419	1.74E-04;
	37.759	1295482054	57.151	18.610	0.146	1.238	6.631	5.411	2.15E-04;
	50.346	1685488866	53.424	16.305	0.150	1.603	7.537	5.110	2.19E-04;
	62.932	2199146755	49.539	13.752	0.156	2.055	8.537	4.719	2.18E-04;
	75.518	2824106863	48.359	12.389	0.149	2.609	10.340	4.726	2.17E-04;
	88.105	3440730114	45.921	12.104	0.131	3.040	12.749	5.059	2.03E-04;
	100.691	3923494422	41.982	11.779	0.131	3.362	15.088	5.327	1.91E-04;
	113.278	4336807273	41.169	11.177	0.122	3.437	16.828	5.401	1.88E-04;
	125.864	4744429762	40.293	10.375	0.122	3.441	18.005	5.318	1.92E-04;
	138.450	5158625454	37.121	9.424	0.137	3.638	18.758	5.115	1.92E-04;
	151.037	5549178201	33.398	8.373	0.132	4.239	19.230	4.834	1.71E-04;
	163.623	5948516635	29.787	7.272	0.117	4.963	19.560	4.512	1.36E-04;
	176.210	6413026311	26.079	-1	0.124	5.346	-1	-1	1.06E-04;
	188.167	6893590158	22.483	4.795	0.146	5.487	19.312	3.552	8.25E-05;
%	Ma-09								
	0	686563567	52.299	20.656	2.333	0.802	2.991	6.000	7.48E-05;
	12.464	717494854	51.337	20.513	-1	0.982	4.039	6.015	1.31E-04;
	24.928	749937849	51.144	20.308	-1	1.063	5.046	6.025	1.58E-04;
	37.391	783317993	51.620	19.981	-1	1.174	5.968	6.022	1.70E-04;
	49.855	811905030	52.071	19.471	-1	1.344	6.764	6.002	1.78E-04;
	62.319	833333572	52.180	18.705	-1	1.518	7.377	5.940	1.79E-04;
	74.783	852137511	52.060	17.601	-1	1.694	7.734	5.784	1.77E-04;
	87.246	873142750	51.850	16.177	-1	1.843	7.817	5.512	1.75E-04;
	99.710	895254758	51.488	14.673	-1	1.930	7.734	5.176	1.75E-04;
	112.174	896544124	50.215	13.193	-1	2.172	7.554	4.805	1.75E-04;
	124.638	907381807	49.362	11.730	-1	2.267	7.307	4.411	1.75E-04;
	137.101	928683511	49.035	10.276	-1	2.202	7.023	4.004	1.74E-04;
	149.565	942922141	48.571	-1	-1	2.156	-1	-1	1.73E-04;
	162.029	949757447	47.884	-1	-1	2.183	-1	-1	1.69E-04;
	174.493	957298570	47.172	-1	-1	2.277	-1	-1	1.57E-04;
	186.333	968512371	46.564	4.11	2.5	2.410	5.68	2.37	1.39E-04;
%	Ma-10								
	17.402	821682464	55.211	22.642	1	0.767	4.589	5.074	9.09E-05;
	34.805	1152973025	54.137	21.860	-1	1.241	6.377	5.171	1.27E-04;
	52.207	1832261508	51.836	20.636	-1	1.788	8.493	5.225	1.49E-04;
	69.610	2984279416	49.188	18.749	-1	2.223	11.103	5.215	1.56E-04;
	87.012	4942401618	44.523	16.006	1.298	2.796	14.270	5.173	1.53E-04;
	104.415	8058569257	40.335	12.592	1.574	3.521	17.861	5.181	1.34E-04;
	121.817	11783643667	36.658	9.091	1.954	4.160	22.086	5.214	1.04E-04;
	139.220	14530862358	30.763	6.268	2.359	4.585	27.533	5.197	7.89E-05;
	156.622	15796861889	23.454	4.259	2.770	5.185	33.692	5.120	6.35E-05;
	174.025	16343424604	15.965	2.968	3.354	6.242	39.651	4.998	5.47E-05;
	191.427	16971875563	9.145	2.301	4.119	7.410	44.499	4.846	-1;
	208.829	17746656858	3.451	2.104	4.662	8.045	47.851	4.677	4.7E-05;
	226.232	18191319698	0.359	2	4.546	8.849	48	4.4	4.5E-05;
	243.634	18238977144	-1	-1	3.579	10.724	-1	-1	-1;
	260.167	18210237542	0	2	2.118	13.600	50	4.1	3.3E-05;
%	Ma-11								
	35.084	729778096	5.403	22.926	0.833	1.454	4.584	5.848	1.72E-04;
	52.625	782121399	5.200	21.764	-1	1.803	5.323	5.901	1.67E-04;
	70.167	925164827	4.567	20.586	1.019	2.394	6.217	5.968	1.66E-04;
	87.709	1093786133	3.722	19.394	1.114	2.493	7.307	6.054	1.65E-04;
	105.251	1288431456	2.727	18.289	1.247	2.498	8.509	6.147	1.69E-04;
	122.793	1520627821	1.511	17.440	1.349	3.219	9.651	6.228	1.68E-04;
	140.334	1797294452	1	17.014	1.406	3.001	10.560	6.278	1.65E-04;
	157.876	2076799793	-1	17.168	1.309	3.058	11.068	6.275	1.68E-04;
	175.418	2263842925	0.5	17.624	1.163	3.666	11.170	6.194	1.64E-04;
	192.960	2357048220	-1	17.761	1.032	5.384	11.032	6.019	1.29E-04;
	210.502	2341511352	0.1	17.317	1.018	6.343	10.761	5.855	1.19E-04;
	228.043	2345271090	-1	15.786	0.8	6.627	10.384	6.008	1.04E-04;
	245.585	2242477859	-1	12.617	0.5	9.334	9.921	6.806	9.13E-05;
	262.250	1967779823	0	8.290	0.778	5.240	9.430	8.128	9.46E-05;
%	Ma-12								
	17.358	649506654	31.581	24.235	0.8	0.826	4.184	5.542	1.69E-04;
	34.716	802000883	31.761	22.647	-1	1.175	5.461	5.629	1.81E-04;
	52.074	1146126994	30.186	21.078	-1	1.512	7.081	5.725	1.77E-04;
	69.431	1615677869	28.580	19.537	-1	1.855	9.218	5.835	1.77E-04;
	86.789	2274472956	25.677	17.904	-1	2.261	11.749	5.915	1.77E-04;
	104.147	3196606246	22.409	15.771	-1	2.421	13.715	5.805	1.78E-04;
	121.505	4533866874	19.729	13.823	1.281	2.719	15.837	5.707	1.65E-04;
	138.863	6426745512	15.431	12.127	1.329	3.334	19.174	5.736	1.20E-04;
	156.221	8294465718	10.869	10.314	1.498	3.771	23.450	5.752	9.50E-05;
	173.579	9320968487	5.311	8.461	1.732	3.998	27.570	5.656	1.03E-04;
	190.936	9646959429	1.008	7.183	1.911	4.382	29.814	5.491	1.10E-04;
	208.294	9816755360	0.080	6.891	1.896	5.867	29.126	5.348	1.05E-04;
	225.652	9863371957	0.265	6.943	1.693	9.930	26.716	5.219	9.45E-05;
	243.010	9519607015	0.291	-1	-1	15.076	-1	-1	8.85E-05;
	259.500	8710459788	0.284	6.9	1.12	18.828	19.3	5.03	9.02E-05;
%	Ma-14								
	22.174	949152856	80.351	25.547	0.222	1.457	5.883	6.180	1.82E-04;
	44.348	1623314041	76.668	21.923	-1	2.335	8.553	6.145	1.75E-04;
	66.522	2704677781	71.443	17.436	-1	3.745	11.905	5.940	1.75E-04;
	88.696	3337126506	68.941	14.132	0.510	5.136	15.998	5.875	1.73E-04;
	110.870	3641072881	64.995	11.883	0.988	6.897	20.125	5.775	1.70E-04;
	133.043	3849140891	60.756	10.118	1.121	7.646	24.758	5.700	1.56E-04;
	155.217	3931014934	55.195	8.916	1.201	8.069	32.037	5.883	1.37E-04;
	177.391	3820221924	52.618	8.169	1.410	8.650	41.043	6.261	1.24E-04;
	199.565	3703760253	51.633	7.553	1.690	9.135	48.656	6.555	1.19E-04;
	221.739	3769355319	48.222	6.741	1.941	9.594	51.725	6.487	1.19E-04;
	243.913	3941547541	43.652	5.410	1.771	10.461	47.145	5.780	1.19E-04;
	266.087	4093853084	38.948	3.939	1.589	11.253	39.188	4.788	1.17E-04;
	288.261	4093172132	34.437	3.341	2.411	10.892	39.155	4.467	1.14E-04;
	310.435	3931417562	31.078	3.219	3.373	10.628	44.281	4.579	-1;
	331.500	3662190903	29.310	3.144	3.792	11.287	50.501	4.795	-1;
%	Ma-15								

```

83.166      647292734      55.024      3.134      0.333      2.046      5.886      5.769      1.07E-04;
124.749      921511080      53.400      2.476      -1      2.473      7.433      5.768      1.06E-04;
166.332      1320462870      52.555      1.904      -1      2.507      9.267      5.826      1.05E-04;
207.915      1891882177      50.048      1.444      -1      2.700      11.454      5.954      1.07E-04;
249.498      2710470012      48.694      1.058      -1      3.175      13.055      5.910      1.04E-04;
291.081      3899129231      44.256      0.733      0.951      3.813      13.206      5.505      8.24E-05;
332.664      5076434882      41.794      0.509      1.661      3.752      13.556      5.221      5.78E-05;
374.247      5784699235      36.318      0.571      1.905      2.286      15.403      5.386      4.95E-05;
415.831      5949970391      31.162      0.815      2.635      1.453      17.013      5.387      5.01E-05;
457.414      6125568912      25.451      1.092      2.924      0.777      18.129      5.164      4.98E-05;
498.997      6130858084      19.898      1.123      3.245      0.384      19.062      4.854      5.66E-05;
540.580      5896545782      12.075      0.709      3.308      0.306      20.263      4.636      7.01E-05;
582.163      5932842332      6.816      0.418      3.810      0.389      21.398      4.384      7.26E-05;
621.667      4764471396      0.570      0.410      4.522      0.514      22.297      4.072      -1;

%
Ma-16
0      658089850      53.386      10.917      0.333      0.734      3.030      5.353      3.30E-05;
17.642      1069878624      58.180      9.348      -1      1.094      4.151      5.334      3.87E-05;
35.284      1694319426      58.593      7.973      -1      1.744      5.855      5.328      6.16E-05;
52.926      2691853705      52.259      6.974      1      2.519      8.698      5.345      7.16E-05;
70.569      4329971538      45.344      6.161      -1      3.514      12.486      5.383      7.21E-05;
88.211      7058325248      41.031      4.888      -1      4.743      16.108      5.414      7.06E-05;
105.853      10221043652      36.839      2.731      1.185      5.802      18.480      5.425      6.29E-05;
123.495      11611519455      30.567      0.943      2.148      7.152      20.103      5.363      4.23E-05;
141.137      12003361415      24.554      0.330      3.178      8.726      22.006      5.240      3.69E-05;
158.779      12199806345      16.921      0.832      4.369      8.637      20.910      5.066      4.03E-05;
176.421      12498725684      8.714      1.674      6.873      8.368      18.076      4.823      4.39E-05;
194.064      12530639874      2.908      2.349      8.395      10.214      15.390      4.529      4.39E-05;
211.706      11897913011      0.973      2.910      6.577      14.528      12.914      4.205      3.99E-05;
229.348      11382693135      0.808      -1      4.392      20.226      -1      -1      3.12E-05;
246.990      10841073050      0.884      -1      2.964      25.907      -1      -1      2.54E-05;
263.750      9992543513      0.890      4.12      1.774      30.832      7.95      3.49      2.59E-05;

%
Ma-17
22.297      849388939      62.626      53.854      0.333      2.751      4.483      5.023      1.56E-04;
44.593      1398382211      61.789      50.637      -1      4.596      6.669      5.212      1.55E-04;
66.890      2356575042      58.321      42.393      -1      6.322      9.463      5.180      1.51E-04;
89.186      4044960316      52.228      35.086      1.564      8.679      13.133      5.081      9.02E-05;
111.483      5431891004      46.220      31.947      1.614      10.731      17.645      5.037      6.02E-05;
133.779      5483803549      36.407      28.553      1.538      13.314      22.103      5.026      5.84E-05;
156.076      5646227416      29.704      24.842      1.754      15.929      26.292      5.018      6.93E-05;
178.372      5880115238      24.333      21.441      2.029      18.204      30.165      4.993      6.19E-05;

200.669      6113058390      18.448      18.981      2.164      20.071      33.680      4.933      4.18E-05;
222.965      6330483976      11.932      18.073      2.190      22.213      36.788      4.821      4.94E-05;
245.262      6432259740      5.936      18.242      2.240      24.150      39.276      4.727      4.25E-05;
267.559      6327177668      1.950      17.330      2.555      24.579      40.668      4.850      3.81E-05;
289.855      5794843382      1.186      13.040      2.884      26.514      40.468      5.404      4.32E-05;
312.152      5007854692      1.236      6.061      2.149      32.793      38.455      6.256      5.84E-05;
333.333      4861164140      1.252      3.182      1.469      34.555      35.521      6.594      6.09E-05;

];

% pontos inicial e final de cada ensaio
%      Ma07      Ma08      Ma09      Ma10      Ma11      Ma12      Ma14      Ma15      Ma16      Ma17
% pontosEnsaio = [ 1 15; 16 31; 32 47; 48 62; 63 76; 77 91; 92 106; 107 120; 121 136; 137 151 ];

% pontos inicial e final de cada ensaio utilizado neste ajuste
%      Ma07      Ma08      Ma09      Ma10      Ma11      Ma12      Ma14      Ma15      Ma16      Ma17
pontosEnsaio = [ 1 15; 16 31; 32 47; 48 62; 63 76; 77 91; 92 97; 107 120; 121 136; 137 151 ];

% ensaios utilizados no ajuste
ensaiosAjuste = [ 7 ]; %ensaiosAjuste = [ 1 2 3 4 5 6 7 8 9 10 ];

% Todos os Kps (1e-10)
todos_kp = [ 1.3548 1.6439 10.000 1.2015 1.1804 1.7185 5.0279 0.1000 0.6 3.7 ];

% Valor inicial que determinou o Kp da faixa
inicial = [ 3.1226e+001 1.0000e-003 4.0000e+001 1.0000e-003 4.0000e+001 1.0000e-003 4.0000e+001 1.0000e-003 ...
1.0000e-003 1.5662e+001 1.0000e-003 4.0000e+001 1.1628e+001 1.0000e-003 ...
1.4759e+001 4.0000e+001 1.0000e-003 4.0000e+001 4.0000e+001 1.0000e-003 ...
1.0000e-003 1.0000e-003 1.0000e-003 4.0000e+001 1.0000e-003 4.0000e+001 ...
4.0000e+001 1.0000e-003 4.8912e-001 1.0000e-003 2.0117e+001 1.0000e-003 ...
6.2345e+000 1.5904e+001 1.0000e-003 1.0000e-003 1.0000e-003 1.0000e-003 ...
1.0000e-003 1.0000e-002 2.0000e+001 1.0000e-002 2.4565e+000 2.0000e+001 ...
7.8031e+000 ]' ;

% Limite inferior dos parâmetros
inferior = [ ones(37,1) * 1E-3; ones(6,1)*0.01]; % k, ks

% Limite superior dos parâmetros
superior = [ ones(37,1) * 40; ones(6,1)*20]; % k, ks

```

4. Dados2.m

```

% *****
% Script: Dados2.m
% Descrição: Valores experimentais

```

```

% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global a1 pontosEnsaio

% PARÂMETROS                VARIÁVEIS DE ESTADO
% k(1)= a1                  y(1)= X
% k(2)= a2                  y(2)= GLC
% k(3)= a3                  y(3)= GLN
% k(4)= a4                  y(4)= LAC
% k(5)= a5                  y(5)= NH4
% k(6)= a6                  y(6)= ALA
% k(7)= a7                  y(7)= GLU
% k(8)= a8                  y(8)= GPVe
% k(9)= a9
% ...
% k(19)= a19

% *****
%                               ATENÇÃO
% OS VALORES EXPERIMENTAIS DEVEM COMEÇAR COM O PRIMEIRO PONTO
% DA FASE EXPONENCIAL para t=0
% *****

a1 = [
% t (h)      Xv (cel/L)  GLC (mM)  GLN (mM)  LAC (mM)  NH4 (mM)  ALA (mM)  GLU (mM)  GPVe (ng/cel)
% Ma-07
0.00      651250000    54.31    30.99    -1.00    0.39     3.48     6.07     1.34E-04;
21.33     822500000    53.21    -1.00    -1.00    1.21     -1.00    -1.00     2.14E-04;
28.33     946250000    52.76    -1.00    -1.00    0.02     -1.00    -1.00     1.72E-04;
45.75     1425000000    54.54    22.68    -1.00    1.90     7.18     6.00     1.76E-04;
52.33     1727500000    42.70    18.21    -1.00    1.90     7.39     5.30     1.85E-04;
69.33     2512500000    45.54    -1.00    -1.00    2.53     -1.00    -1.00     1.53E-04;
76.50     3450000000    45.23    -1.00    -1.00    2.68     -1.00    -1.00     1.39E-04;
93.50     5565000000    44.64    -1.00    2.98     2.63     -1.00    -1.00     1.10E-04;
100.83    6660000000    43.60    -1.00    -1.00    2.63     -1.00    -1.00     1.37E-04;
118.33    10662500000    30.88    10.40     3.25     2.63     17.92     5.14     1.04E-04;
126.08    11265625000    31.61     6.19     3.85     2.85     12.04     3.56     9.40E-05;
142.00    13343750000    26.46    -1.00     3.36     3.23     -1.00    -1.00     5.05E-05;
148.17    12928125000    27.79    -1.00     4.00     3.64     -1.00    -1.00     4.45E-05;
165.00    14656250000    21.62    -1.00    -1.00     3.36     -1.00    -1.00     4.08E-05;
173.92    15312500000    17.34    -1.00    -1.00     3.36     -1.00    -1.00     5.07E-05;
189.25    18250000000    13.80     4.32     5.34     2.97     32.50     4.20     4.29E-05;
197.00    19175000000    10.33    -1.00    -1.00     3.10     -1.00    -1.00     4.05E-05;
213.08    17690625000     5.06     2.70     6.11     3.23     24.71     3.09     3.40E-05;
221.17    19687500000     2.79    -1.00     6.11     2.53     -1.00    -1.00     4.36E-05;
237.33    21825000000     0.76    -1.00    -1.00     2.97     -1.00    -1.00     2.58E-05;
244.50    17859375000     0.76    -1.00    -1.00     3.64     -1.00    -1.00     3.64E-05;
261.75    16637500000     0.76     4.48     6.40     8.44     37.02     4.17     3.32E-05;
313.50    18525000000     0.76    -1.00     3.43    17.65     -1.00    -1.00     4.05E-05;
340.00    8831250000     0.76    -1.00     1.15    18.35     -1.00    -1.00     -1.00;
% Ma-08
0.00      638750000    65.66    21.48     0.15     0.55     2.77     4.91     -1;
21.50     893750000    61.09    -1.00     0.11     0.88     -1.00    -1.00     1.36E-04;
27.42     1005000000    58.80    -1.00     0.15     0.95     -1.00    -1.00     2.06E-04;
44.50     1452500000    56.18    17.78     0.22     1.46     7.28     5.39     2.80E-04;
54.33     1834375000    51.79    -1.00     0.42     1.71     -1.00    -1.00     2.68E-04;
68.42     2327500000    47.00    12.44     0.22     2.24     8.89     4.47     2.16E-04;
76.17     3110625000    49.90    12.43     0.15     2.72    10.55     4.78     2.23E-04;
91.92     3643750000    45.89    12.29     0.11     3.05    13.43     5.22     1.95E-04;
98.67     3831250000    40.51    11.80     0.15     3.43    14.92     5.30     1.43E-04;
121.50    4592500000    41.64    -1.00     0.11     3.43     -1.00    -1.00     1.21E-04;
142.75    5307500000    35.57    -1.00     0.15     3.71     -1.00    -1.00     1.93E-04;
163.67    5677500000    29.81     7.26     0.11     5.05    19.54     4.50     1.33E-04;
188.17    6898125000    22.47    -1.00     0.15     5.46     -1.00    -1.00     8.31E-05;
% Ma-09
0.00      686250000    52.43    20.58    -1.00     0.79     2.94     5.27     6.98E-05;
19.33     730000000    47.22    -1.00    -1.00     1.16     -1.00    -1.00     1.71E-04;
25.67     758750000    53.76    -1.00    -1.00     0.95     -1.00    -1.00     1.42E-04;
45.58     802500000    52.07    19.64    -1.00     1.30     6.47     5.95     1.81E-04;
72.25     848125000    52.09    18.31    -1.00     1.64     7.94     5.97     2.09E-04;
91.42     797500000    51.13    15.31    -1.00     2.07     7.68     5.34     1.60E-04;
97.33     917500000    52.60    -1.00    -1.00     1.64     -1.00    -1.00     1.84E-04;
115.00    886250000    49.50    -1.00    -1.00     2.33     -1.00    -1.00     1.82E-04;
140.00    937500000    49.10     9.95    -1.00     2.15     6.92     3.90     1.14E-04;
171.00    952500000    47.32    -1.00    -1.00     2.24     -1.00    -1.00     1.63E-04;
186.33    969375000    46.58    -1.00    -1.00     2.42     -1.00    -1.00     1.37E-04;
% Ma-10
0.00      817500000    53.13    23.14    -1.00     0.39     3.01     4.95     6.19E-05;
19.50     785000000    56.40    -1.00    -1.00     0.88     -1.00    -1.00     7.86E-05;
26.83     856250000    55.50    -1.00    -1.00     0.88     -1.00    -1.00     1.21E-04;
45.42     1545625000    51.23    -1.00    -1.00     1.49     -1.00    -1.00     1.51E-04;
54.00     2050000000    51.48    -1.00    -1.00     2.15     -1.00    -1.00     9.31E-05;
69.50     3291250000    49.20    18.74    -1.00     2.15    10.86     5.81     1.58E-04;
75.00     3712500000    50.36    18.21     1.19     2.23    12.18     6.00     1.72E-04;
93.92     5917500000    40.13    -1.00     1.38     2.97     -1.00    -1.00     1.33E-04;
100.25    6730000000    40.35    13.59     1.45     3.50    17.24     5.76     1.50E-04;
117.17    10587500000    40.06    -1.00     1.84     3.80     -1.00    -1.00     1.03E-04;
123.75    11890625000    35.75     8.44     2.02     4.65    22.11     5.25     1.01E-04;
141.67    14118750000    29.89    -1.00     2.49     4.47     -1.00    -1.00     7.39E-05;
169.33    14600000000    17.51    -1.00     2.97     5.48     -1.00    -1.00     5.73E-05;
188.50    15360000000    10.77     2.31     3.99     7.91     44.38     4.87     4.99E-05;
218.33    18375000000     0.20     2.18     4.86    -1.00     49.04     4.58     -1;
238.17    18106875000     0.20    -1.00     4.13     9.31     -1.00    -1.00     -1;

```

```

% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Dados
global a1 pontosEnsaio

% PARÂMETROS                VARIÁVEIS DE ESTADO
% k(1)= a1                  y(1)= X
% k(2)= a2                  y(2)= GLC
% k(3)= a3                  y(3)= GLN
% k(4)= a4                  y(4)= LAC
% k(5)= a5                  y(5)= NH4
% k(6)= a6                  y(6)= ALA
% k(7)= a7                  y(7)= GLU
% k(8)= a8                  y(8)= GPVe
% k(9)= a9
% ...
% k(19)= a19

% *****
%                               ATENÇÃO
% OS VALORES EXPERIMENTAIS DEVEM COMEÇAR COM O PRIMEIRO PONTO
% DA FASE EXPONENCIAL para t=0
% *****

a1 = [
% t (h)      Xv (cel/L)  GLC (mM)  GLN (mM)  LAC (mM)  NH4 (mM)  ALA (mM)  GLU (mM)  GPVe (ng/cel)
% Ma-07
0.00      651250000    54.31    30.99    -1.00    0.39     3.48     6.07     1.34E-04;
21.33     822500000    53.21    -1.00    -1.00    1.21     -1.00    -1.00    2.14E-04;
28.33     946250000    52.76    -1.00    -1.00    0.02     -1.00    -1.00    1.72E-04;
45.75     1425000000    54.54    22.68    -1.00    1.90     7.18     6.00    1.76E-04;
52.33     1727500000    42.70    18.21    -1.00    1.90     7.39     5.30    1.85E-04;
69.33     2512500000    45.54    -1.00    -1.00    2.53     -1.00    -1.00    1.53E-04;
76.50     3450000000    45.23    -1.00    -1.00    2.68     -1.00    -1.00    1.39E-04;
93.50     5565000000    44.64    -1.00    2.98     2.63     -1.00    -1.00    1.10E-04;
100.83    6660000000    43.60    -1.00    -1.00    2.63     -1.00    -1.00    1.37E-04;
118.33    10662500000    30.88    10.40     3.25     2.63     17.92     5.14    1.04E-04;
126.08    11265625000    31.61     6.19     3.85     2.85     12.04     3.56    9.40E-05;
142.00    13343750000    26.46    -1.00     3.36     3.23     -1.00    -1.00    5.05E-05;
148.17    12928125000    27.79    -1.00     4.00     3.64     -1.00    -1.00    4.45E-05;
165.00    14656250000    21.62    -1.00    -1.00     3.36     -1.00    -1.00    4.08E-05;
173.92    15312500000    17.34    -1.00    -1.00     3.36     -1.00    -1.00    5.07E-05;
189.25    18250000000    13.80     4.32     5.34     2.97     32.50     4.20    4.29E-05;
197.00    19175000000    10.33    -1.00    -1.00     3.10     -1.00    -1.00    4.05E-05;
213.08    17690625000     5.06     2.70     6.11     3.23     24.71     3.09    3.40E-05;
221.17    19687500000     2.79    -1.00     6.11     2.53     -1.00    -1.00    4.36E-05;
237.33    21825000000     0.76    -1.00    -1.00     2.97     -1.00    -1.00    2.58E-05;
244.50    17859375000     0.76    -1.00    -1.00     3.64     -1.00    -1.00    3.64E-05;
261.75    16637500000     0.76     4.48     6.40     8.44     37.02     4.17    3.32E-05;
313.50    18525000000     0.76    -1.00     3.43    17.65     -1.00    -1.00    4.05E-05;
340.00    8831250000     0.76    -1.00     1.15    18.35     -1.00    -1.00    -1.00;
% Ma-08
0.00      638750000    65.66    21.48     0.15     0.55     2.77     4.91     -1;
21.50     893750000    61.09    -1.00     0.11     0.88     -1.00    -1.00    1.36E-04;
27.42     1005000000    58.80    -1.00     0.15     0.95     -1.00    -1.00    2.06E-04;
44.50     1452500000    56.18    17.78     0.22     1.46     7.28     5.39    2.80E-04;
54.33     1834375000    51.79    -1.00     0.42     1.71     -1.00    -1.00    2.68E-04;
68.42     2327500000    47.00    12.44     0.22     2.24     8.89     4.47    2.16E-04;
76.17     3110625000    49.90    12.43     0.15     2.72    10.55     4.78    2.23E-04;
91.92     3643750000    45.89    12.29     0.11     3.05    13.43     5.22    1.95E-04;
98.67     3831250000    40.51    11.80     0.15     3.43    14.92     5.30    1.43E-04;
121.50    4592500000    41.64    -1.00     0.11     3.43     -1.00    -1.00    1.21E-04;
142.75    5307500000    35.57    -1.00     0.15     3.71     -1.00    -1.00    1.93E-04;
163.67    5677500000    29.81     7.26     0.11     5.05    19.54     4.50    1.33E-04;
188.17    6898125000    22.47    -1.00     0.15     5.46     -1.00    -1.00    8.31E-05;
% Ma-09
0.00      686250000    52.43    20.58    -1.00     0.79     2.94     5.27     6.98E-05;
19.33     730000000    47.22    -1.00    -1.00     1.16     -1.00    -1.00    1.71E-04;
25.67     758750000    53.76    -1.00    -1.00     0.95     -1.00    -1.00    1.42E-04;
45.58     802500000    52.07    19.64    -1.00     1.30     6.47     5.95    1.81E-04;
72.25     848125000    52.09    18.31    -1.00     1.64     7.94     5.97    2.09E-04;
91.42     797500000    51.13    15.31    -1.00     2.07     7.68     5.34    1.60E-04;
97.33     917500000    52.60    -1.00    -1.00     1.64     -1.00    -1.00    1.84E-04;
115.00    886250000    49.50    -1.00    -1.00     2.33     -1.00    -1.00    1.82E-04;
140.00    937500000    49.10     9.95    -1.00     2.15     6.92     3.90    1.14E-04;
171.00    952500000    47.32    -1.00    -1.00     2.24     -1.00    -1.00    1.63E-04;
186.33    969375000    46.58    -1.00    -1.00     2.42     -1.00    -1.00    1.37E-04;
% Ma-10
0.00      817500000    53.13    23.14    -1.00     0.39     3.01     4.95     6.19E-05;
19.50     785000000    56.40    -1.00    -1.00     0.88     -1.00    -1.00    7.86E-05;
26.83     856250000    55.50    -1.00    -1.00     0.88     -1.00    -1.00    1.21E-04;
45.42     1545625000    51.23    -1.00    -1.00     1.49     -1.00    -1.00    1.51E-04;
54.00     2050000000    51.48    -1.00    -1.00     2.15     -1.00    -1.00    9.31E-05;
69.50     3291250000    49.20    18.74    -1.00     2.15    10.86     5.81    1.58E-04;
75.00     3712500000    50.36    18.21     1.19     2.23    12.18     6.00    1.72E-04;
93.92     5917500000    40.13    -1.00     1.38     2.97     -1.00    -1.00    1.33E-04;
100.25    6730000000    40.35    13.59     1.45     3.50    17.24     5.76    1.50E-04;
117.17    10587500000    40.06    -1.00     1.84     3.80     -1.00    -1.00    1.03E-04;
123.75    11890625000    35.75     8.44     2.02     4.65    22.11     5.25    1.01E-04;
141.67    14118750000    29.89    -1.00     2.49     4.47     -1.00    -1.00    7.39E-05;
169.33    14600000000    17.51    -1.00     2.97     5.48     -1.00    -1.00    5.73E-05;
188.50    15360000000    10.77     2.31     3.99     7.91     44.38     4.87    4.99E-05;
218.33    18375000000     0.20     2.18     4.86    -1.00     49.04     4.58     -1;
238.17    18106875000     0.20    -1.00     4.13     9.31     -1.00    -1.00     -1;

```

260.17	18193750000	0.20	-1.00	1.93	13.99	-1.00	-1.00	-1;
Ma-11								
0.00	737500000	5.86	25.22	-1.00	0.36	3.40	5.77	1.77E-04;
20.58	808125000	6.05	-1.00	-1.00	0.88	-1.00	-1.00	1.40E-04;
26.67	795000000	5.35	-1.00	-1.00	1.17	-1.00	-1.00	1.73E-04;
45.08	762500000	5.36	-1.00	-1.00	1.84	-1.00	-1.00	1.91E-04;
52.25	767500000	5.23	-1.00	-1.00	1.63	-1.00	-1.00	1.48E-04;
69.15	894375000	4.60	-1.00	1.02	2.35	-1.00	-1.00	1.75E-04;
76.33	952500000	-1.00	20.28	-1.00	2.65	6.43	5.98	1.32E-04;
95.33	1230000000	3.30	-1.00	1.16	2.35	-1.00	-1.00	1.47E-04;
100.08	1218125000	3.06	-1.00	1.20	2.45	-1.00	-1.00	1.70E-04;
117.33	1557500000	1.92	-1.00	1.35	2.65	-1.00	-1.00	1.60E-04;
124.83	1390000000	1.35	-1.00	1.33	3.68	-1.00	-1.00	1.54E-04;
142.58	1915000000	-1.00	-1.00	1.42	2.88	-1.00	-1.00	1.35E-04;
150.08	1937500000	-1.00	16.37	1.36	2.65	11.14	6.28	1.53E-04;
166.75	2193750000	-1.00	-1.00	1.22	3.83	-1.00	-1.00	1.53E-04;
174.25	2243750000	-1.00	18.12	1.19	3.13	11.12	6.26	1.73E-04;
190.58	2362500000	-1.00	17.41	1.02	5.32	10.99	6.24	1.27E-04;
214.25	2325000000	-1.00	-1.00	1.05	6.52	-1.00	-1.00	1.19E-04;
239.75	2317500000	-1.00	15.31	-1.00	7.68	10.11	5.77	9.17E-05;
262.25	1961250000	-1.00	7.23	-1.00	15.36	9.39	8.58	9.51E-05;
Ma-12								
0.00	748750000	29.99	25.83	-1.00	0.43	3.11	5.46	1.58E-04;
20.92	702500000	31.79	-1.00	-1.00	0.91	-1.00	-1.00	1.68E-04;
27.58	746250000	31.96	-1.00	-1.00	1.02	-1.00	-1.00	1.96E-04;
43.58	820000000	31.31	-1.00	-1.00	1.35	-1.00	-1.00	1.79E-04;
50.92	1116250000	30.07	-1.00	-1.00	1.50	-1.00	-1.00	1.44E-04;
68.50	1681250000	28.42	19.51	-1.00	1.80	8.85	5.79	1.56E-04;
77.83	1927500000	28.16	-1.00	-1.00	2.07	-1.00	-1.00	2.01E-04;
92.67	2625000000	23.90	17.76	-1.00	2.39	13.07	6.05	1.29E-04;
101.33	3460000000	22.90	15.70	-1.00	2.47	13.36	5.75	1.67E-04;
116.92	4072500000	20.40	14.24	1.36	2.30	14.83	5.66	1.83E-04;
122.92	4515000000	19.80	-1.00	1.21	2.95	-1.00	-1.00	1.61E-04;
141.42	6290625000	14.15	-1.00	-1.00	-1.00	-1.00	-1.00	1.21E-04;
145.67	7200000000	14.24	11.58	1.39	3.53	20.57	5.77	9.29E-05;
163.50	8250000000	8.70	9.57	1.57	3.92	25.25	5.76	9.94E-05;
189.17	9600000000	1.06	7.00	1.93	4.37	30.28	5.48	1.10E-04;
215.50	9845000000	0.28	7.06	1.84	6.93	28.10	5.30	-1;
239.33	9680000000	0.28	6.97	1.44	14.35	24.25	5.12	8.83E-05;
259.50	8690000000	0.28	-1.00	-1.00	18.75	-1.00	-1.00	9.06E-05;
Ma-14								
0.00	848750000	81.34	28.44	0.2	0.66	3.57	6.07	-1.00;
19.17	896250000	81.15	-1.00	-1.00	1.38	-1.00	-1.00	1.85E-04;
42.67	1470625000	77.53	-1.00	-1.00	2.30	-1.00	-1.00	1.62E-04;
48.67	1757500000	74.85	21.66	-1.00	2.39	9.19	6.23	-1.00;
66.75	2855000000	70.42	16.68	0.34	3.99	11.40	5.67	1.47E-04;
75.25	3090000000	71.29	16.07	0.39	4.14	13.99	6.08	1.70E-04;
90.50	3268125000	68.80	-1.00	0.71	5.24	-1.00	-1.00	1.51E-04;
97.50	3172500000	67.56	13.17	0.23	5.67	17.58	5.84	-1.00;
111.50	3661875000	64.62	11.93	1.29	7.17	20.61	5.80	1.81E-04;
139.92	3881250000	59.67	9.61	1.02	7.66	25.95	5.65	-1.00;
163.83	3937500000	52.76	-1.00	1.32	8.29	-1.00	-1.00	1.29E-04;
189.33	3712500000	52.83	-1.00	1.51	8.97	-1.00	-1.00	-1.00;
239.00	3893750000	44.61	6.04	2.03	10.08	51.80	6.25	1.19E-04;
260.17	4075000000	40.27	3.95	1.32	11.34	37.64	4.74	-1.00;
308.00	3962500000	31.24	3.29	3.41	10.49	43.99	4.59	1.11E-04;
331.50	3656250000	29.39	3.14	3.75	11.34	50.62	4.80	-1.00;
Ma-15								
0.00	671250000	55.78	4.60	-1.00	0.71	3.28	5.87	1.30E-04;
22.83	656250000	57.45	-1.00	-1.00	1.31	-1.00	-1.00	-1.00;
48.17	637500000	56.54	-1.00	-1.00	1.53	-1.00	-1.00	-1.00;
71.00	716250000	51.85	-1.00	-1.00	1.92	-1.00	-1.00	1.08E-04;
93.92	753750000	51.71	-1.00	-1.00	2.15	-1.00	-1.00	-1.00;
118.50	898750000	51.86	-1.00	-1.00	2.41	-1.00	-1.00	8.60E-05;
125.17	1008750000	54.93	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
143.83	1200000000	52.09	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
149.00	1035000000	52.13	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
166.17	1297500000	53.62	-1.00	-1.00	2.50	-1.00	-1.00	1.01E-04;
171.92	1362500000	51.49	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
190.00	1350000000	51.16	1.61	-1.00	2.70	10.21	5.83	9.56E-05;
215.17	1790000000	49.63	-1.00	-1.00	2.70	-1.00	-1.00	-1.00;
243.42	2337500000	52.33	1.17	-1.00	3.03	13.57	6.13	8.29E-05;
261.92	3268750000	47.06	-1.00	0.72	3.40	-1.00	-1.00	-1.00;
268.33	3255000000	46.90	0.79	0.78	3.67	12.73	5.58	1.12E-04;
285.17	4100000000	43.07	-1.00	0.82	3.36	-1.00	-1.00	-1.00;
292.42	3887500000	44.81	0.88	0.92	4.06	14.12	5.79	8.40E-05;
310.50	4822500000	45.26	-1.00	1.53	4.06	-1.00	-1.00	6.33E-05;
316.42	4807500000	43.00	0.45	1.56	4.06	11.92	4.84	6.37E-05;
333.08	4812500000	47.10	-1.00	1.63	3.77	-1.00	-1.00	-1.00;
340.42	5171250000	39.81	-1.00	1.72	3.49	-1.00	-1.00	5.73E-05;
357.17	5320000000	37.24	0.52	1.70	2.79	15.02	5.47	-1.00;
384.50	6037500000	35.90	-1.00	2.07	-1.00	-1.00	-1.00	4.81E-05;
408.67	6010000000	28.89	-1.00	2.34	1.59	-1.00	-1.00	-1.00;
430.17	5850000000	32.53	0.89	3.11	-1.00	17.54	5.36	5.18E-05;
455.17	6630000000	25.91	-1.00	2.89	0.81	-1.00	-1.00	-1.00;
482.67	6090000000	21.49	-1.00	2.98	-1.00	-1.00	-1.00	5.03E-05;
502.33	6150000000	19.82	1.25	3.37	0.37	19.01	4.78	-1.00;
527.00	5780000000	13.73	-1.00	3.06	-1.00	-1.00	-1.00	6.79E-05;
555.50	5320000000	10.61	0.42	3.63	0.33	20.80	4.59	-1.00;
597.42	6090000000	4.38	-1.00	3.94	-1.00	-1.00	-1.00	7.19E-05;
621.67	4740000000	0.57	0.45	4.54	0.51	22.27	4.06	-1.00;
Ma-16								
0.00	726250000	53.21	10.93	-1.00	0.75	3.08	5.35	3.45E-05;
19.92	996250000	58.36	-1.00	-1.00	1.10	-1.00	-1.00	3.45E-05;
27.33	1232500000	63.84	-1.00	-1.00	1.48	-1.00	-1.00	5.64E-05;
45.42	2325000000	59.12	7.27	-1.00	2.16	7.17	5.01	8.16E-05;
52.08	2900625000	51.40	-1.00	-1.00	2.51	-1.00	-1.00	6.57E-05;
68.33	4252500000	47.30	-1.00	-1.00	3.27	-1.00	-1.00	8.30E-05;
75.83	5212500000	42.07	-1.00	-1.00	3.94	-1.00	-1.00	6.42E-05;
92.42	7728750000	40.70	4.74	0.61	5.13	16.90	6.10	5.07E-05;
101.17	8882500000	38.91	3.23	0.92	5.33	18.26	5.87	7.82E-05;
117.67	11375000000	31.74	1.21	1.78	6.93	19.36	5.51	4.46E-05;
124.17	11287500000	30.64	0.98	2.22	6.86	19.62	5.27	4.17E-05;

260.17	18193750000	0.20	-1.00	1.93	13.99	-1.00	-1.00	-1;
Ma-11								
0.00	737500000	5.86	25.22	-1.00	0.36	3.40	5.77	1.77E-04;
20.58	808125000	6.05	-1.00	-1.00	0.88	-1.00	-1.00	1.40E-04;
26.67	795000000	5.35	-1.00	-1.00	1.17	-1.00	-1.00	1.73E-04;
45.08	762500000	5.36	-1.00	-1.00	1.84	-1.00	-1.00	1.91E-04;
52.25	767500000	5.23	-1.00	-1.00	1.63	-1.00	-1.00	1.48E-04;
69.15	894375000	4.60	-1.00	1.02	2.35	-1.00	-1.00	1.75E-04;
76.33	952500000	-1.00	20.28	-1.00	2.65	6.43	5.98	1.32E-04;
95.33	1230000000	3.30	-1.00	1.16	2.35	-1.00	-1.00	1.47E-04;
100.08	1218125000	3.06	-1.00	1.20	2.45	-1.00	-1.00	1.70E-04;
117.33	1557500000	1.92	-1.00	1.35	2.65	-1.00	-1.00	1.60E-04;
124.83	1390000000	1.35	-1.00	1.33	3.68	-1.00	-1.00	1.54E-04;
142.58	1915000000	-1.00	-1.00	1.42	2.88	-1.00	-1.00	1.35E-04;
150.08	1937500000	-1.00	16.37	1.36	2.65	11.14	6.28	1.53E-04;
166.75	2193750000	-1.00	-1.00	1.22	3.83	-1.00	-1.00	1.53E-04;
174.25	2243750000	-1.00	18.12	1.19	3.13	11.12	6.26	1.73E-04;
190.58	2362500000	-1.00	17.41	1.02	5.32	10.99	6.24	1.27E-04;
214.25	2325000000	-1.00	-1.00	1.05	6.52	-1.00	-1.00	1.19E-04;
239.75	2317500000	-1.00	15.31	-1.00	7.68	10.11	5.77	9.17E-05;
262.25	1961250000	-1.00	7.23	-1.00	15.36	9.39	8.58	9.51E-05;
Ma-12								
0.00	748750000	29.99	25.83	-1.00	0.43	3.11	5.46	1.58E-04;
20.92	702500000	31.79	-1.00	-1.00	0.91	-1.00	-1.00	1.68E-04;
27.58	746250000	31.96	-1.00	-1.00	1.02	-1.00	-1.00	1.96E-04;
43.58	820000000	31.31	-1.00	-1.00	1.35	-1.00	-1.00	1.79E-04;
50.92	1116250000	30.07	-1.00	-1.00	1.50	-1.00	-1.00	1.44E-04;
68.50	1681250000	28.42	19.51	-1.00	1.80	8.85	5.79	1.56E-04;
77.83	1927500000	28.16	-1.00	-1.00	2.07	-1.00	-1.00	2.01E-04;
92.67	2625000000	23.90	17.76	-1.00	2.39	13.07	6.05	1.29E-04;
101.33	3460000000	22.90	15.70	-1.00	2.47	13.36	5.75	1.67E-04;
116.92	4072500000	20.40	14.24	1.36	2.30	14.83	5.66	1.83E-04;
122.92	4515000000	19.80	-1.00	1.21	2.95	-1.00	-1.00	1.61E-04;
141.42	6290625000	14.15	-1.00	-1.00	-1.00	-1.00	-1.00	1.21E-04;
145.67	7200000000	14.24	11.58	1.39	3.53	20.57	5.77	9.29E-05;
163.50	8250000000	8.70	9.57	1.57	3.92	25.25	5.76	9.94E-05;
189.17	9600000000	1.06	7.00	1.93	4.37	30.28	5.48	1.10E-04;
215.50	9845000000	0.28	7.06	1.84	6.93	28.10	5.30	-1;
239.33	9680000000	0.28	6.97	1.44	14.35	24.25	5.12	8.83E-05;
259.50	8690000000	0.28	-1.00	-1.00	18.75	-1.00	-1.00	9.06E-05;
Ma-14								
0.00	848750000	81.34	28.44	0.2	0.66	3.57	6.07	-1.00;
19.17	896250000	81.15	-1.00	-1.00	1.38	-1.00	-1.00	1.85E-04;
42.67	1470625000	77.53	-1.00	-1.00	2.30	-1.00	-1.00	1.62E-04;
48.67	1757500000	74.85	21.66	-1.00	2.39	9.19	6.23	-1.00;
66.75	2855000000	70.42	16.68	0.34	3.99	11.40	5.67	1.47E-04;
75.25	3090000000	71.29	16.07	0.39	4.14	13.99	6.08	1.70E-04;
90.50	3268125000	68.80	-1.00	0.71	5.24	-1.00	-1.00	1.51E-04;
97.50	3172500000	67.56	13.17	0.23	5.67	17.58	5.84	-1.00;
111.50	3661875000	64.62	11.93	1.29	7.17	20.61	5.80	1.81E-04;
139.92	3881250000	59.67	9.61	1.02	7.66	25.95	5.65	-1.00;
163.83	3937500000	52.76	-1.00	1.32	8.29	-1.00	-1.00	1.29E-04;
189.33	3712500000	52.83	-1.00	1.51	8.97	-1.00	-1.00	-1.00;
239.00	3893750000	44.61	6.04	2.03	10.08	51.80	6.25	1.19E-04;
260.17	4075000000	40.27	3.95	1.32	11.34	37.64	4.74	-1.00;
308.00	3962500000	31.24	3.29	3.41	10.49	43.99	4.59	1.11E-04;
331.50	3656250000	29.39	3.14	3.75	11.34	50.62	4.80	-1.00;
Ma-15								
0.00	671250000	55.78	4.60	-1.00	0.71	3.28	5.87	1.30E-04;
22.83	656250000	57.45	-1.00	-1.00	1.31	-1.00	-1.00	-1.00;
48.17	637500000	56.54	-1.00	-1.00	1.53	-1.00	-1.00	-1.00;
71.00	716250000	51.85	-1.00	-1.00	1.92	-1.00	-1.00	1.08E-04;
93.92	753750000	51.71	-1.00	-1.00	2.15	-1.00	-1.00	-1.00;
118.50	898750000	51.86	-1.00	-1.00	2.41	-1.00	-1.00	8.60E-05;
125.17	1008750000	54.93	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
143.83	1200000000	52.09	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
149.00	1035000000	52.13	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
166.17	1297500000	53.62	-1.00	-1.00	2.50	-1.00	-1.00	1.01E-04;
171.92	1362500000	51.49	-1.00	-1.00	2.50	-1.00	-1.00	-1.00;
190.00	1350000000	51.16	1.61	-1.00	2.70	10.21	5.83	9.56E-05;
215.17	1790000000	49.63	-1.00	-1.00	2.70	-1.00	-1.00	-1.00;
243.42	2337500000	52.33	1.17	-1.00	3.03	13.57	6.13	8.29E-05;
261.92	3268750000	47.06	-1.00	0.72	3.40	-1.00	-1.00	-1.00;
268.33	3255000000	46.90	0.79	0.78	3.67	12.73	5.58	1.12E-04;
285.17	4100000000	43.07	-1.00	0.82	3.36	-1.00	-1.00	-1.00;
292.42	3887500000	44.81	0.88	0.92	4.06	14.12	5.79	8.40E-05;
310.50	4822500000	45.26	-1.00	1.53	4.06	-1.00	-1.00	6.33E-05;
316.42	4807500000	43.00	0.45	1.56	4.06	11.92	4.84	6.37E-05;
333.08	4812500000	47.10	-1.00	1.63	3.77	-1.00	-1.00	-1.00;
340.42	5171250000	39.81	-1.00	1.72	3.49	-1.00	-1.00	5.73E-05;
357.17	5320000000	37.24	0.52	1.70	2.79	15.02	5.47	-1.00;
384.50	6037500000	35.90	-1.00	2.07	-1.00	-1.00	-1.00	4.81E-05;
408.67	6010000000	28.89	-1.00	2.34	1.59	-1.00	-1.00	-1.00;
430.17	5850000000	32.53	0.89	3.11	-1.00	17.54	5.36	5.18E-05;
455.17	6630000000	25.91	-1.00	2.89	0.81	-1.00	-1.00	-1.00;
482.67	6090000000	21.49	-1.00	2.98	-1.00	-1.00	-1.00	5.03E-05;
502.33	6150000000	19.82	1.25	3.37	0.37	19.01	4.78	-1.00;
527.00	5780000000	13.73	-1.00	3.06	-1.00	-1.00	-1.00	6.79E-05;
555.50	5320000000	10.61	0.42	3.63	0.33	20.80	4.59	-1.00;
597.42	6090000000	4.38	-1.00	3.94	-1.00	-1.00	-1.00	7.19E-05;
621.67	4740000000	0.57	0.45	4.54	0.51	22.27	4.06	-1.00;
Ma-16								
0.00	726250000	53.21	10.93	-1.00	0.75	3.08	5.35	3.45E-05;
19.92	996250000	58.36	-1.00	-1.00	1.10	-1.00	-1.00	3.45E-05;
27.33	1232500000	63.84	-1.00	-1.00	1.48	-1.00	-1.00	5.64E-05;
45.42	2325000000	59.12	7.27	-1.00	2.16	7.17	5.01	8.16E-05;
52.08	2900625000	51.40	-1.00	-1.00	2.51	-1.00	-1.00	6.57E-05;
68.33	4252500000	47.30	-1.00	-1.00	3.27	-1.00	-1.00	8.30E-05;
75.83	5212500000	42.07	-1.00	-1.00	3.94	-1.00	-1.00	6.42E-05;
92.42	7728750000	40.70	4.74	0.61	5.13	16.90	6.10	5.07E-05;
101.17	8882500000	38.91	3.23	0.92	5.33	18.26	5.87	7.82E-05;
117.67	11375000000	31.74	1.21	1.78	6.93	19.36	5.51	4.46E-05;
124.17	11287500000	30.64	0.98	2.22	6.86	19.62	5.27	4.17E-05;

```

        end
        erro = s;
    end
    if (formula==5)
        s = 0;
        tam = size(a2);
        for i=1:tam(1)
            for j=1:tam(2)
                if (ali(i,j)~= -1) % existe valor experimental
                    s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
                end
            end
        end
        erro = s;
    end
    if (formula==6)
        max_ali = max(ali); % maior valor experimental sem normalização
        s = 0;
        tam = size(a2);
        for i=1:tam(1)
            for j=1:tam(2)
                if (ali(i,j) > epsilon * max_ali(j)) % não precisa verificar se (ali(i,j)~= -1)
                    s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
                end
            end
        end
        erro = s;
    end
    if (formula==7)
        tam = size(a2);
        sr = 0; sa = 0;
        for j=1:tam(2)
            % retira pontos inexistentes (experimental=-1)
            aux_a2 = [];
            aux_res2 = [];
            cnt=1;
            for i=1:tam(1)
                if (ali(i,j)~= -1)
                    aux_a2(cnt) = a2(i,j);
                    aux_res2(cnt) = res2(i,j);
                    cnt = cnt + 1;
                end
            end
            % Calcula coeficientes
            % calcula coeficiente de correlação
            C = corrcoef(aux_res2,aux_a2); r = abs(C(1,2)); sr = sr + abs(r-1);
            % calcula coeficiente angular
            P = polyfit(aux_res2,aux_a2,1); a = P(1); sa = sa + abs(a-1);
        end
        erro = sr + sa;
    end
    if (formula==8) % obrigatório valores normalizados
        tam = size(a2);
        % forma vetor coluna com todos os valores
        res3 = []; a3 = [];
        for j=1:tam(2)
            % retira pontos inexistentes (experimental=-1)
            aux_a2 = [];
            aux_res2 = [];
            cnt=1;
            for i=1:tam(1)
                if (ali(i,j)~= -1)
                    aux_a2(cnt) = a2(i,j);
                    aux_res2(cnt) = res2(i,j);
                    cnt = cnt + 1;
                end
            end
            % junta colunas
            res3 = [ res3; aux_res2' ]; %#ok<AGROW>
            a3 = [ a3; aux_a2' ]; %#ok<AGROW>
        end
        C = corrcoef(res3,a3); r = abs(C(1,2)); % calcula coeficiente de correlação
        P = polyfit(res3,a3,1); a = P(1); % calcula coeficiente angular
        erro = abs(r-1) + abs(a-1);
    end
    ErroTotal = ErroTotal + erro;

end % for

Residuo_Anterior = Residuo;
Residuo = ErroTotal;

ErroTotal = ErroTotal * FatorPrecisao;

end

```

```

        end
        erro = s;
    end
    if (formula==5)
        s = 0;
        tam = size(a2);
        for i=1:tam(1)
            for j=1:tam(2)
                if (ali(i,j)~= -1) % existe valor experimental
                    s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
                end
            end
        end
        erro = s;
    end
    if (formula==6)
        max_ali = max(ali); % maior valor experimental sem normalização
        s = 0;
        tam = size(a2);
        for i=1:tam(1)
            for j=1:tam(2)
                if (ali(i,j) > epsilon * max_ali(j)) % não precisa verificar se (ali(i,j)~= -1)
                    s = s + ((res2(i,j)-a2(i,j))./((res2(i,j)+a2(i,j))/2)) .^2;
                end
            end
        end
        erro = s;
    end
    if (formula==7)
        tam = size(a2);
        sr = 0; sa = 0;
        for j=1:tam(2)
            % retira pontos inexistentes (experimental=-1)
            aux_a2 = [];
            aux_res2 = [];
            cnt=1;
            for i=1:tam(1)
                if (ali(i,j)~= -1)
                    aux_a2(cnt) = a2(i,j);
                    aux_res2(cnt) = res2(i,j);
                    cnt = cnt + 1;
                end
            end
            % Calcula coeficientes
            % calcula coeficiente de correlação
            C = corrcoef(aux_res2,aux_a2); r = abs(C(1,2)); sr = sr + abs(r-1);
            % calcula coeficiente angular
            P = polyfit(aux_res2,aux_a2,1); a = P(1); sa = sa + abs(a-1);
        end
        erro = sr + sa;
    end
    if (formula==8) % obrigatório valores normalizados
        tam = size(a2);
        % forma vetor coluna com todos os valores
        res3 = []; a3 = [];
        for j=1:tam(2)
            % retira pontos inexistentes (experimental=-1)
            aux_a2 = [];
            aux_res2 = [];
            cnt=1;
            for i=1:tam(1)
                if (ali(i,j)~= -1)
                    aux_a2(cnt) = a2(i,j);
                    aux_res2(cnt) = res2(i,j);
                    cnt = cnt + 1;
                end
            end
            % junta colunas
            res3 = [ res3; aux_res2' ]; %#ok<AGROW>
            a3 = [ a3; aux_a2' ]; %#ok<AGROW>
        end
        C = corrcoef(res3,a3); r = abs(C(1,2)); % calcula coeficiente de correlação
        P = polyfit(res3,a3,1); a = P(1); % calcula coeficiente angular
        erro = abs(r-1) + abs(a-1);
    end
    ErroTotal = ErroTotal + erro;
end % for

Residuo_Anterior = Residuo;
Residuo = ErroTotal;

ErroTotal = ErroTotal * FatorPrecisao;

end

```


7. Graficos.m

```
% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

function Graficos

% Construção dos gráficos
global al res1 ensaiosAjuste pontosEnsaio Ensaio
global Kp todos_kp

Dados2;
alaux = al;

for Ensaio=ensaiosAjuste
    % Cópia para ali apenas um ensaio de al
    al = alaux(pontosEnsaio(Ensaio,1):pontosEnsaio(Ensaio,2),:);

    % Condições iniciais
    tam = size(al);          % tamanho (linhas,colunas) de ali
    X0 = al(1,2);
    GLC0 = al(1,3);
    GLN0 = al(1,4);
    LAC0 = al(1,5);
    NH40 = al(1,6);
    ALA0 = al(1,7);
    GLU0 = al(1,8);

    Kp = todos_kp(Ensaio) * 1E-10;
    PURTOT0 = X0 * Kp;

    %          X(cel/L)  GLC (mM)  GLN (mM)  LAC (mM)  NH4 (mM)  ALA (mM)  GLU (nM)  PURItotal (mM)
    y0 = [  X0      GLC0      GLN0      LAC0      NH40      ALA0      GLU0      PURTOT0      ];

    % Valores de tempo
    tspan = al(:,1); % primeira coluna

    % Resolve Equações Diferenciais Ordinárias
    [t y] = ode45('equacoes', tspan, y0);
    res1=[t y];

    % Cria nova figura e gera gráfico
    figure();
    Graficos2;

end

end
```

8. Graficos2.m

```
% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Construção dos gráficos
global al res1

%subplot(<N. de linhas>, <N. colunas>, <Posição da Fig.>)
%plot(x1,y1,<"cor1_símbolo1">, x2,y2,<"cor2_símbolo2">, ...)
%Dados Experimentais
```

7. Graficos.m

```
% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

function Graficos

% Construção dos gráficos
global al res1 ensaiosAjuste pontosEnsaio Ensaio
global Kp todos_kp

Dados2;
alaux = al;

for Ensaio=ensaiosAjuste
    % Copia para ali apenas um ensaio de al
    al = alaux(pontosEnsaio(Ensaio,1):pontosEnsaio(Ensaio,2),:);

    % Condições iniciais
    tam = size(al);          % tamanho (linhas,colunas) de ali
    X0 = al(1,2);
    GLC0 = al(1,3);
    GLN0 = al(1,4);
    LAC0 = al(1,5);
    NH40 = al(1,6);
    ALA0 = al(1,7);
    GLU0 = al(1,8);

    Kp = todos_kp(Ensaio) * 1E-10;
    PURTOT0 = X0 * Kp;

    %      X(cel/L)  GLC (mM)  GLN (mM)  LAC (mM)  NH4 (mM)  ALA (mM)  GLU (nM)  PURItotal (mM)
    y0 = [  X0      GLC0    GLN0     LAC0    NH40    ALA0    GLU0    PURTOT0      ];

    % Valores de tempo
    tspan = al(:,1); % primeira coluna

    % Resolve Equações Diferenciais Ordinárias
    [t y] = ode45('equacoes', tspan, y0);
    res1=[t y];

    % Cria nova figura e gera gráfico
    figure();
    Graficos2;

end

end
```

8. Graficos2.m

```
% *****
% Script: Graficos.m
% Descrição: Gráficos comparando dados experimentais e dados gerados com o
%            uso dos parâmetros ajustados
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% Construção dos gráficos
global al res1

%subplot(<N. de linhas>, <N. colunas>, <Posição da Fig.>)
%plot(x1,y1,<"cor1_símbolo1">, x2,y2,<"cor2_símbolo2">, ...)
%Dados Experimentais
```

```

%x: ai(:,1) - toda a 1a coluna da matriz "ai" (dados exp. do Ensaio_i)
%y: ai(:,2) - toda a 2a coluna da matriz "ai" (dados exp. do Ensaio_i)

%Dados Simulados
%x: resi(:,1) - toda a 1a coluna da matriz "resi" (dados simulados do Ensaio_i)
%y: resi(:,2) - toda a 2a coluna da matriz "resi" (dados simulados do Ensaio_i)

% Tamanho
tamanho = size(res1);

% Gráfico Xv
yexp = a1(:,2);
ycalc = res1(:,2);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,1),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('Xv (cel/L)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico GLC
yexp = a1(:,3);
ycalc = res1(:,3);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,2),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('GLC (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico GLN
yexp = a1(:,4);
ycalc = res1(:,4);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,3),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('GLN (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico LAC
yexp = a1(:,5);
ycalc = res1(:,5);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,4),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('LAC (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico NH4
yexp = a1(:,6);
ycalc = res1(:,6);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,5),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('NH4 (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico ALA
yexp = a1(:,7);
ycalc = res1(:,7);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,6),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('ALA (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

```

```

% Gráfico GLU
yexp = a1(:,8);
ycalc = res1(:,8);
[xg,yg] = ValoresGrafico(a1(:,1),yexp,res1(:,1),ycalc);
subplot(3,3,7),plot(a1(:,1),yexp,'b*',xg,yg,'b-')
ylabel('GLU (mM)')
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

% Gráfico PUR
subplot(3,3,8),plot(res1(:,1),res1(:,9),'b-')
ylabel('PURI (mM)');
xlabel('tempo (h)');
xlim([0 res1(tamanho(1),1)*1.1]);
ymax = max(res1(:,9)); % maior valor de y (calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end

```

9. poliedros.m

```

% *****
% Módulo: poliedros.m
% Descrição: Minimização utilizando Poliédro Flexíveis
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****

% *****
% Implementação do Algoritmo dos POLIÉDROS FLEXÍVEIS
% Baseado na implementação em FORTRAN encontrada em:
% "Computer-aided design of communication networks"
% Yi-Sheng Zhu & Wai-Kai Chen
% Publicado por World Scientific, 2000
% ISBN 981022351X 9789610223519
% *****

% *****
% Sintaxe
% [XF,PRECISAO,QUANTITER,QUANTEVAL] = poliedros(FO,Step,X0,XI,XS,MaxIter,Precision);
% Exemplo de chamada
% [XF,PRECISAO,QUANTITER,QUANTEVAL] = poliedros(@funcao,0.3,[1 1 1 1],[0 0 0 0],[2 2 2 2],2000,1e-8);
% *****

% *****
% Modificações feitas por Marilena
% 1) Verificar se os valores sorteados estão dentro dos limites (inferior/superior)
% 2) Contar quantas vezes a FO é avaliada (QUANTEVAL)
% 3) Contar a Quantidade de Iterações feitas (QUANTITER)
% 4) Exibir a matriz inicial
% 5) Permitir a passagem dos coeficientes como parâmetros
% *****

function [XF,PRECISAO,QUANTITER,QUANTEVAL] = poliedros(FO,Step0,X0,XI,XS,MaxIter,Precision,Alfa,Gama,Beta,Delta)
% Quantidade de avaliações da FO
QUANTEVAL = 0; % Linha acrescentada por Marilena
% Nx is the number of independent variables
Nx = length(X0);
% Step is the initial step size
Step = Step0;
% Inicializa vetores e matrizes
X = zeros(Nx,1);
X1 = zeros(Nx,Nx);
Sum = zeros(Nx,1);
% X(I) is the array of initial guesses
X = X0;
% Inicializa variáveis
XNx = Nx;
IN = 1;
% Somatória inicial
Sum(IN) = feval(FO,X);

```

```

QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
% Call second(time)
K1 = Nx + 1;
K2 = Nx + 2;
K3 = Nx + 3;
K4 = Nx + 4;
% Start
A = zeros(Nx,Nx);
Vn = Nx;
Step1 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)+Vn-1.0);
Step2 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)-1.0);
for J=1:Nx
    A(1,J) = 0.0;
end
for I=2:K1
    for J=1:Nx
        A(I,J) = Step2;
        L = I - 1;
        A(I,L) = Step1;
    end
end
for I=1:K1
    for J=1:Nx
        X1(I,J) = X(J)+A(I,J);
        % *****
        % Código acrescentado por Marilena
        % *****
        if (X1(I,J) < XI(J))
            X1(I,J) = XI(J);
        end
        if (X1(I,J) > XS(J))
            X1(I,J) = XS(J);
        end
        % *****
    end
end
for I=1:K1
    for J=1:Nx
        X(J) = X1(I,J);
        IN = I;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    end
end

% *****
% Código acrescentado por Marilena
% *****
% Exibe matriz inicial
fprintf('\n\nMatriz inicial\n');
tamX1 = size(X1);
for I=1:tamX1(1)
    for J=1:tamX1(2)
        fprintf('%4e ',X1(I,J));
    end
    fprintf('\n');
end
fprintf('\n\nIterações\n');
% *****

% Inicia looping
Difer = realmax;
IJK = 1;
while (Difer > Precision && IJK <= MaxIter)
    % Select largest Value of Sum(I) in Simplex
    Sumh = Sum(1);
    Index = 1;
    for I=2:K1
        if (Sum(I) >= Sumh)
            Sumh = Sum(I);
            Index = I;
        end
    end
    % Select minimum Value of Sum(I) in Simplex
    Suml = Sum(1);
    Kount = 1;
    for I=2:K1
        if (Sum(I) <= Suml)
            Suml = Sum(I);
            Kount = I;
        end
    end
    % Find centroid of points with I different from Index
    for J=1:Nx
        Sum2 = 0.0;
        for I=1:K1
            Sum2 = Sum2 + X1(I,J);
        end

```

```

QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
% Call second(time)
K1 = Nx + 1;
K2 = Nx + 2;
K3 = Nx + 3;
K4 = Nx + 4;
% Start
A = zeros(Nx,Nx);
Vn = Nx;
Step1 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)+Vn-1.0);
Step2 = Step ./ (Vn.*sqrt(2.0)) .* (sqrt(Vn+1.0)-1.0);
for J=1:Nx
    A(1,J) = 0.0;
end
for I=2:K1
    for J=1:Nx
        A(I,J) = Step2;
        L = I - 1;
        A(I,L) = Step1;
    end
end
for I=1:K1
    for J=1:Nx
        X1(I,J) = X(J)+A(I,J);
        % *****
        % Código acrescentado por Marilena
        % *****
        if (X1(I,J) < XI(J))
            X1(I,J) = XI(J);
        end
        if (X1(I,J) > XS(J))
            X1(I,J) = XS(J);
        end
        % *****
    end
end
for I=1:K1
    for J=1:Nx
        X(J) = X1(I,J);
        IN = I;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    end
end

% *****
% Código acrescentado por Marilena
% *****
% Exibe matriz inicial
fprintf('\n\nMatriz inicial\n');
tamX1 = size(X1);
for I=1:tamX1(1)
    for J=1:tamX1(2)
        fprintf('%4e ',X1(I,J));
    end
    fprintf('\n');
end
fprintf('\n\nIterações\n');
% *****

% Inicia looping
Difer = realmax;
IJK = 1;
while (Difer > Precision && IJK <= MaxIter)
    % Select largest Value of Sum(I) in Simplex
    Sumh = Sum(1);
    Index = 1;
    for I=2:K1
        if (Sum(I) >= Sumh)
            Sumh = Sum(I);
            Index = I;
        end
    end
    % Select minimum Value of Sum(I) in Simplex
    Suml = Sum(1);
    Kount = 1;
    for I=2:K1
        if (Sum(I) <= Suml)
            Suml = Sum(I);
            Kount = I;
        end
    end
    % Find centroid of points with I different from Index
    for J=1:Nx
        Sum2 = 0.0;
        for I=1:K1
            Sum2 = Sum2 + X1(I,J);
        end

```

```

        if (X1(K4,J) < XI(J))
            X1(K4,J) = XI(J);
        end
        if (X1(K4,J) > XS(J))
            X1(K4,J) = XS(J);
        end
        % *****
        X(J) = X1(K4,J);
    end
    IN = K4;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    if (Sumh > Sum(K4))
        for J=1:Nx
            X1(Index,J) = X1(K4,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    else
        for J=1:Nx
            for I=1:K1
                X1(I,J) = Delta .* (X1(I,J) + X1(Kount,J));
                % *****
                % Código acrescentado por Marilena
                % *****
                if (X1(I,J) < XI(J))
                    X1(I,J) = XI(J);
                end
                if (X1(I,J) > XS(J))
                    X1(I,J) = XS(J);
                end
            end
            % *****
        end
        end
        for I=1:K1
            for J=1:Nx
                X(J) = X1(I,J);
            end
            IN = I;
            Sum(IN) = feval(FO,X);
            QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
        end
        end
    else
        for J=1:Nx
            X1(Index,J) = X1(K3,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    end
end
for J=1:Nx
    X(J) = X1(K2,J);
end
IN = K2;
Sum(IN) = feval(FO,X);
QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
Difer = 0.0;
for I=1:K1
    Difer = Difer + (Sum(I)-Sum(K2)).^2;
end
Difer = 1.0 ./ XNx .* sqrt(Difer);
IJK = IJK + 1;
fprintf('%d\t%.8e\n',IJK-1,Difer);
PRECISAO = Difer;
if (IJK > MaxIter)
    disp('Quantidade maxima de iteracoes excedida!');
end
end %while
XF = zeros(Nx,1);
for J=1:Nx
    X(J) = X1(Kount,J);
    XF(J) = X(J);
end
QUANTITER = IJK-1; % Linha acrescentada por Marilena
end

```

```

        if (X1(K4,J) < XI(J))
            X1(K4,J) = XI(J);
        end
        if (X1(K4,J) > XS(J))
            X1(K4,J) = XS(J);
        end
        % *****
        X(J) = X1(K4,J);
    end
    IN = K4;
    Sum(IN) = feval(FO,X);
    QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    if (Sumh > Sum(K4))
        for J=1:Nx
            X1(Index,J) = X1(K4,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    else
        for J=1:Nx
            for I=1:K1
                X1(I,J) = Delta .* (X1(I,J) + X1(Kount,J));
                % *****
                % Código acrescentado por Marilena
                % *****
                if (X1(I,J) < XI(J))
                    X1(I,J) = XI(J);
                end
                if (X1(I,J) > XS(J))
                    X1(I,J) = XS(J);
                end
            end
            % *****
        end
        end
        for I=1:K1
            for J=1:Nx
                X(J) = X1(I,J);
            end
            IN = I;
            Sum(IN) = feval(FO,X);
            QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
        end
        end
    else
        for J=1:Nx
            X1(Index,J) = X1(K3,J);
            X(J) = X1(Index,J);
        end
        IN = Index;
        Sum(IN) = feval(FO,X);
        QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
    end
end
for J=1:Nx
    X(J) = X1(K2,J);
end
IN = K2;
Sum(IN) = feval(FO,X);
QUANTEVAL = QUANTEVAL + 1; % Linha acrescentada por Marilena
Difer = 0.0;
for I=1:K1
    Difer = Difer + (Sum(I)-Sum(K2)).^2;
end
Difer = 1.0 ./ XNx .* sqrt(Difer);
IJK = IJK + 1;
fprintf('%d\t%.8e\n',IJK-1,Difer);
PRECISAO = Difer;
if (IJK > MaxIter)
    disp('Quantidade maxima de iteracoes excedida!');
end
end %while
XF = zeros(Nx,1);
for J=1:Nx
    X(J) = X1(Kount,J);
    XF(J) = X(J);
end
QUANTITER = IJK-1; % Linha acrescentada por Marilena
end

```


10. ValoresGrafico.m

```
% *****
% Módulo: ValoresGrafico.m
% Descrição: retorna o maior valor de x que não tem y=-1
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/07/2012
% *****
function [xg,yg] = ValoresGrafico(xe,ye,xc,yc)
% Calcula último tempo onde existe valor experimental
tfinal = -1;
for i=1:length(xe)
    if (ye(i)>=0)
        tfinal = xe(i);
    end
end
% Prepara dados para plotar
xg = [];
yg = [];
pos = 1;
for i=1:length(xc)
    if (xc(i)<=tfinal)
        xg = [ xg xc(i) ];
        yg = [ yg yc(i) ];
        pos = i; % posição do último ponto inserido
    end
end
% Insere 10 pontos após tfinal (para ficar alinhado com o último asterisco)
for i=1:10
    if (pos < length(xc))
        xg = [ xg xc(pos+1) ];
        yg = [ yg yc(pos+1) ];
        pos = pos + 1;
    end
end
end
```

APÊNDICE H – LISTAGEM DO SOFTWARE DE SIMULAÇÃO

1. Arquivos

Este *software* é composto pelos seguintes arquivos:

- equacoes.m
- Faixas.m
- Graficos.m
- SimulaS2.m

A listagem de cada um desses arquivos é mostrada a seguir.

2. equacoes.m

```
% *****
% Módulo: equacoes.m
% Descrição: Equações diferenciais utilizadas no ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.03
% Criação: 21/04/2009
% Última modificação: 05/08/2012
% *****

function [xcn]=equacoes(t,y)

    global k Kp simulacao

    X = y(1);
    GLC = y(2);
    GLN = y(3);
    LAC = y(4);
    NH4 = y(5);
    ALA = y(6);
    GLU = y(7);
    PURI = y(8);

    if simulacao == 1
        k = [ 1.0000e-003 4.0000e+001 4.0000e+001 8.9868e+000 2.4462e+001 2.9146e+000 ...
              1.0000e-003 1.0000e-003 1.0000e-003 1.0000e-003 4.0000e+001 1.0000e-003 ...
              1.6316e+001 4.0000e+001 1.0424e+001 4.0000e+001 1.0000e-003 1.0000e-003 ...
              1.0000e-003 1.0000e-003 1.0000e-003 4.0000e+001 1.0000e-003 1.0000e-003 ...
              4.0000e+001 1.9116e+001 4.0000e+001 4.0000e+001 1.0000e-003 1.0000e-003 ...
              4.2120e+000 4.0000e+001 8.9608e+000 1.0000e-003 1.0000e-003 1.0000e-003 ...
              1.0746e+001 2.0000e+001 2.0000e+001 1.0000e-002 1.5397e+000 2.0000e+001 ...
              8.8561e-001 ]';
    end

    if simulacao == 2
        k = [ 1.0000e-003 2.3134e+001 3.1532e+001 1.0000e-003 4.0000e+001 1.0000e-003 ...
              1.0000e-003 2.3575e+001 1.0000e-003 1.0000e-003 1.3486e+000 4.0000e+001 ...
              1.4760e+001 4.0000e+001 1.0000e-003 4.0000e+001 7.3580e+000 1.0000e-003 ...
              1.0000e-003 1.0000e-003 1.0000e-003 1.9708e+001 1.0000e-003 1.0000e-003 ...
              4.0000e+001 4.0000e+001 3.7460e+001 4.0000e+001 1.0000e-003 1.0000e-003 ...
              1.0000e-003 1.0000e-003 6.3411e+000 1.0000e-003 1.0000e-003 1.0000e-003 ...
              1.0000e-003 8.0773e+000 1.6260e+001 1.0000e-002 1.1513e+000 2.0000e+001 ...
              1.4330e+001 ]';
    end
```

```

% Última modificação: 05/08/2012
% *****

% Construção dos gráficos
global al res1 gpv Ensaio

% Tamanhos
tamres = size(res1);
if (Ensaio~=0) % comparar com experimental
    tama = size(al);
    maiort = max(al(tama(1),1),res1(tamres(1),1)) * 1.1; % maior tempo
else
    maiort = res1(tamres(1),1) * 1.1; % maior tempo
end

% Gráfico xv
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,1),plot(al(:,1),al(:,2),'b*',res1(:,1),res1(:,2),'b-')
    ymax = max(max(al(:,2)),max(res1(:,2))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,1),plot(res1(:,1),res1(:,2),'b-')
    ymax = max(res1(:,2)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('xv (cel/l)');
xlabel('tempo (h)');

% Gráfico GLC
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,2),plot(al(:,1),al(:,3),'b*',res1(:,1),res1(:,3),'b-')
    ymax = max(max(al(:,3)),max(res1(:,3))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,2),plot(res1(:,1),res1(:,3),'b-')
    ymax = max(res1(:,3)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('GLC (mM)');
xlabel('tempo (h)');

% Gráfico GLN
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,3),plot(al(:,1),al(:,4),'b*',res1(:,1),res1(:,4),'b-')
    ymax = max(max(al(:,4)),max(res1(:,4))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,3),plot(res1(:,1),res1(:,4),'b-')
    ymax = max(res1(:,4)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('GLN (mM)');
xlabel('tempo (h)');

% Gráfico LAC
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,4),plot(al(:,1),al(:,5),'b*',res1(:,1),res1(:,5),'b-')
    ymax = max(max(al(:,5)),max(res1(:,5))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,4),plot(res1(:,1),res1(:,5),'b-')
    ymax = max(res1(:,5)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('LAC (mM)');
xlabel('tempo (h)');

% Gráfico NH4
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,5),plot(al(:,1),al(:,6),'b*',res1(:,1),res1(:,6),'b-')
    ymax = max(max(al(:,6)),max(res1(:,6))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,5),plot(res1(:,1),res1(:,6),'b-')
    ymax = max(res1(:,6)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('NH4 (mM)');

```

```

% Última modificação: 05/08/2012
% *****

% Construção dos gráficos
global al res1 gpv Ensaio

% Tamanhos
tamres = size(res1);
if (Ensaio~=0) % comparar com experimental
    tama = size(al);
    maiort = max(al(tama(1),1),res1(tamres(1),1)) * 1.1; % maior tempo
else
    maiort = res1(tamres(1),1) * 1.1; % maior tempo
end

% Gráfico xv
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,1),plot(al(:,1),al(:,2),'b*',res1(:,1),res1(:,2),'b-')
    ymax = max(max(al(:,2)),max(res1(:,2))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,1),plot(res1(:,1),res1(:,2),'b-')
    ymax = max(res1(:,2)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('xv (cel/l)');
xlabel('tempo (h)');

% Gráfico GLC
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,2),plot(al(:,1),al(:,3),'b*',res1(:,1),res1(:,3),'b-')
    ymax = max(max(al(:,3)),max(res1(:,3))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,2),plot(res1(:,1),res1(:,3),'b-')
    ymax = max(res1(:,3)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('GLC (mM)');
xlabel('tempo (h)');

% Gráfico GLN
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,3),plot(al(:,1),al(:,4),'b*',res1(:,1),res1(:,4),'b-')
    ymax = max(max(al(:,4)),max(res1(:,4))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,3),plot(res1(:,1),res1(:,4),'b-')
    ymax = max(res1(:,4)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('GLN (mM)');
xlabel('tempo (h)');

% Gráfico LAC
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,4),plot(al(:,1),al(:,5),'b*',res1(:,1),res1(:,5),'b-')
    ymax = max(max(al(:,5)),max(res1(:,5))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,4),plot(res1(:,1),res1(:,5),'b-')
    ymax = max(res1(:,5)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('LAC (mM)');
xlabel('tempo (h)');

% Gráfico NH4
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,5),plot(al(:,1),al(:,6),'b*',res1(:,1),res1(:,6),'b-')
    ymax = max(max(al(:,6)),max(res1(:,6))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,5),plot(res1(:,1),res1(:,6),'b-')
    ymax = max(res1(:,6)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maiort]);
ylabel('NH4 (mM)');

```

```

xlabel('tempo (h)');

% Gráfico ALA
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,6),plot(al(:,1),al(:,7),'b*',res1(:,1),res1(:,7),'b-')
    ymax = max(max(al(:,7)),max(res1(:,7))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,6),plot(res1(:,1),res1(:,7),'b-')
    ymax = max(res1(:,7)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('ALA (mM)');
xlabel('tempo (h)');

% Gráfico GLU
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,7),plot(al(:,1),al(:,8),'b*',res1(:,1),res1(:,8),'b-')
    ymax = max(max(al(:,8)),max(res1(:,8))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,7),plot(res1(:,1),res1(:,8),'b-')
    ymax = max(res1(:,8)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('GLU (mM)');
xlabel('tempo (h)');

% Gráfico PUR
subplot(3,3,8),plot(res1(:,1),res1(:,9),'b-')
ymax = max(res1(:,9)); % maior valor de y (calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('PUR (mM)');
xlabel('tempo (h)');

% Gráfico GPVe
ycalc = gpv * 1E7;
if (Ensaio~=0) % comparar com experimental
    yexp = al(:,9) * 1E7;
    subplot(3,3,9),plot(al(:,1),yexp,'b*',res1(:,1),ycalc,'b-')
    ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,9),plot(res1(:,1),ycalc,'b-')
    ymax = max(ycalc); % maior valor de y (calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('GPVe (ng/1E7cel)');
xlabel('tempo (h)');

```

5. SimulaS2.m

```

% *****
% Módulo: SimulaS2.m
% Descrição: Simulação dos resultados obtidos pelo ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/08/2012
% *****

% Simulação
function SimulaS2
clear all;
clc
global al res1 gpv mi Ensaio Kp

% Ler valores iniciais
[filename, caminho] = uigetfile('*.xls', 'Selecione o arquivo de dados','Valores_01_Ma07.xls');
if isequal(filename,0) || isequal(caminho,0)

```

```

xlabel('tempo (h)');

% Gráfico ALA
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,6),plot(al(:,1),al(:,7),'b*',res1(:,1),res1(:,7),'b-')
    ymax = max(max(al(:,7)),max(res1(:,7))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,6),plot(res1(:,1),res1(:,7),'b-')
    ymax = max(res1(:,7)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('ALA (mM)');
xlabel('tempo (h)');

% Gráfico GLU
if (Ensaio~=0) % comparar com experimental
    subplot(3,3,7),plot(al(:,1),al(:,8),'b*',res1(:,1),res1(:,8),'b-')
    ymax = max(max(al(:,8)),max(res1(:,8))); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,7),plot(res1(:,1),res1(:,8),'b-')
    ymax = max(res1(:,8)); % maior valor de y (experimental ou calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('GLU (mM)');
xlabel('tempo (h)');

% Gráfico PUR
subplot(3,3,8),plot(res1(:,1),res1(:,9),'b-')
ymax = max(res1(:,9)); % maior valor de y (calculado)
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('PUR (mM)');
xlabel('tempo (h)');

% Gráfico GPVe
ycalc = gpv * 1E7;
if (Ensaio~=0) % comparar com experimental
    yexp = al(:,9) * 1E7;
    subplot(3,3,9),plot(al(:,1),yexp,'b*',res1(:,1),ycalc,'b-')
    ymax = max(max(yexp),max(ycalc)); % maior valor de y (experimental ou calculado)
else
    subplot(3,3,9),plot(res1(:,1),ycalc,'b-')
    ymax = max(ycalc); % maior valor de y (calculado)
end
if ( ymax > 0 )
    ylim([0 ymax*1.1]);
end
xlim([0 maior]);
ylabel('GPVe (ng/1E7cel)');
xlabel('tempo (h)');

```

5. SimulaS2.m

```

% *****
% Módulo: SimulaS2.m
% Descrição: Simulação dos resultados obtidos pelo ajuste de parâmetros
% Autor: Marilena Martins Pamboukian
% Versão: 2.0
% Criação: 21/04/2009
% Última modificação: 05/08/2012
% *****

% Simulação
function SimulaS2
clear all;
clc
global al res1 gpv mi Ensaio Kp

% Ler valores iniciais
[filename, caminho] = uigetfile('*.xls', 'Selecione o arquivo de dados','Valores_01_Ma07.xls');
if isequal(filename,0) || isequal(caminho,0)

```

```

tamanho = size(res1);
for i=1:tamanho(1)
    if (res1(i,3) > valorGLC) && (res1(i,4) > valorGLN)
        gpv(i) = -2.19E-8 * res1(i,1).^2 + 3.16E-6 * res1(i,1) + 4.29E-5 + gpvs1;
        ultimo = gpv(i);
        prim = 1;
    else
        if (prim == 1) % primeira vez nesta fase
            prim = 0;
            gpv2 = -1.01E-7 * res1(i,1) + 6.09E-5; % valor do primeiro ponto
            gpvs2 = ultimo - gpv2; % GPV anterior - valor do primeiro ponto
        end
        gpv(i) = -1.01E-7 * res1(i,1) + 6.09E-5 + gpvs2;
    end
end

% Cálculo do resíduo usando Fórmula 6
if (Ensaio~=0) % Comparar com experimental
    % Descobre primeiro ponto experimental após Tini
    p1 = 1;
    while (al(p1,1)<Tini && p1<tam(1))
        p1 = p1 + 1;
    end
    % Descobre último ponto experimental antes de Tfin
    p2 = p1;
    while (al(p2,1)<Tfin && p2<tam(1))
        p2 = p2 + 1;
    end
    p2 = p2 - 1;
    % Calcula valores para comparar com experimentais
    tspan=al(p1:p2,1);
    Faixas(Tini,y0,[],[]);
    [t,y]=ode45('equacoes',tspan,y0,options);
    res2=[t y];
    % GPV experimental
    gpvexp=[];
    ultimo = GPVe0;
    prim = 1;
    tamanho2 = size(res2);
    for i=1:tamanho2(1)
        if (res2(i,3) > valorGLC) && (res2(i,4) > valorGLN)
            gpvexp(i) = -2.19E-8 * res2(i,1).^2 + 3.16E-6 * res2(i,1) + 4.29E-5 + gpvs1;
            ultimo = gpvexp(i);
            prim = 1;
        else
            if (prim == 1) % primeira vez nesta fase
                prim = 0;
                gpv2 = -1.01E-7 * res2(i,1) + 6.09E-5; % valor do primeiro ponto
                gpvs2 = ultimo - gpv2; % GPV anterior - valor do primeiro ponto
            end
            gpvexp(i) = -1.01E-7 * res2(i,1) + 6.09E-5 + gpvs2;
        end
    end
    % Calcula resíduos pela fórmula 6
    fprintf('Resíduo - fórmula 6\n');
    Soma_X = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,2),res2(:,2));
    Soma_GLC = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,3),res2(:,3));
    Soma_GLN = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,4),res2(:,4));
    Soma_LAC = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,5),res2(:,5));
    Soma_NH4 = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,6),res2(:,6));
    Soma_ALA = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,7),res2(:,7));
    Soma_GLU = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,8),res2(:,8));
    Soma_GPVe = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,9) * 1E7, gpvexp * 1E7);
    fprintf('Resíduo X = %f\n',Soma_X);
    fprintf('Resíduo GLC = %f\n',Soma_GLC);
    fprintf('Resíduo GLN = %f\n',Soma_GLN);
    fprintf('Resíduo LAC = %f\n',Soma_LAC);
    fprintf('Resíduo NH4 = %f\n',Soma_NH4);
    fprintf('Resíduo ALA = %f\n',Soma_ALA);
    fprintf('Resíduo GLU = %f\n',Soma_GLU);
    fprintf('Resíduo GPVe= %f\n\n',Soma_GPVe);
end
% Construção do gráfico
figure;
Graficos;
% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');
end

% Cálculo de resíduo usando fórmula 6
function R = calcula_residuo2(te,tc,e,c) % al(:,1), res1(:,1), al(:,x), res1(:,x)
    % Normalização
    m = max([ e; c ] );
    e = e ./ m;
    c = c ./ m;
    % Cálculo do resíduo
    epsilon = 1E-6;
    maxve = max(e); % maior valor experimental normalizado

```

```

tamanho = size(res1);
for i=1:tamanho(1)
    if (res1(i,3) > valorGLC) && (res1(i,4) > valorGLN)
        gpv(i) = -2.19E-8 * res1(i,1).^2 + 3.16E-6 * res1(i,1) + 4.29E-5 + gpvs1;
        ultimo = gpv(i);
        prim = 1;
    else
        if (prim == 1) % primeira vez nesta fase
            prim = 0;
            gpv2 = -1.01E-7 * res1(i,1) + 6.09E-5; % valor do primeiro ponto
            gpvs2 = ultimo - gpv2; % GPV anterior - valor do primeiro ponto
        end
        gpv(i) = -1.01E-7 * res1(i,1) + 6.09E-5 + gpvs2;
    end
end

% Cálculo do resíduo usando Fórmula 6
if (Ensaio~=0) % Comparar com experimental
    % Descobre primeiro ponto experimental após Tini
    p1 = 1;
    while (al(p1,1)<Tini && p1<tam(1))
        p1 = p1 + 1;
    end
    % Descobre último ponto experimental antes de Tfin
    p2 = p1;
    while (al(p2,1)<Tfin && p2<tam(1))
        p2 = p2 + 1;
    end
    p2 = p2 - 1;
    % Calcula valores para comparar com experimentais
    tspan=al(p1:p2,1);
    Faixas(Tini,y0',[]);
    [t,y]=ode45('equacoes',tspan,y0,options);
    res2=[t y];
    % GPV experimental
    gpvexp=[];
    ultimo = GPVe0;
    prim = 1;
    tamanho2 = size(res2);
    for i=1:tamanho2(1)
        if (res2(i,3) > valorGLC) && (res2(i,4) > valorGLN)
            gpvexp(i) = -2.19E-8 * res2(i,1).^2 + 3.16E-6 * res2(i,1) + 4.29E-5 + gpvs1;
            ultimo = gpvexp(i);
            prim = 1;
        else
            if (prim == 1) % primeira vez nesta fase
                prim = 0;
                gpv2 = -1.01E-7 * res2(i,1) + 6.09E-5; % valor do primeiro ponto
                gpvs2 = ultimo - gpv2; % GPV anterior - valor do primeiro ponto
            end
            gpvexp(i) = -1.01E-7 * res2(i,1) + 6.09E-5 + gpvs2;
        end
    end
    % Calcula resíduos pela fórmula 6
    fprintf('Resíduo - fórmula 6\n');
    Soma_X = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,2),res2(:,2));
    Soma_GLC = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,3),res2(:,3));
    Soma_GLN = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,4),res2(:,4));
    Soma_LAC = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,5),res2(:,5));
    Soma_NH4 = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,6),res2(:,6));
    Soma_ALA = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,7),res2(:,7));
    Soma_GLU = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,8),res2(:,8));
    Soma_GPVe = calcula_residuo2(al(p1:p2,1),res2(:,1),al(p1:p2,9) * 1E7, gpvexp * 1E7);
    fprintf('Resíduo X = %f\n',Soma_X);
    fprintf('Resíduo GLC = %f\n',Soma_GLC);
    fprintf('Resíduo GLN = %f\n',Soma_GLN);
    fprintf('Resíduo LAC = %f\n',Soma_LAC);
    fprintf('Resíduo NH4 = %f\n',Soma_NH4);
    fprintf('Resíduo ALA = %f\n',Soma_ALA);
    fprintf('Resíduo GLU = %f\n',Soma_GLU);
    fprintf('Resíduo GPVe= %f\n\n',Soma_GPVe);
end
% Construção do gráfico
figure;
Graficos;
% Finaliza
disp('© 2009-2012 - Marilena Martins Pamboukian');
end

% Cálculo de resíduo usando fórmula 6
function R = calcula_residuo2(te,tc,e,c) % al(:,1), res1(:,1), al(:,x), res1(:,x)
    % Normalização
    m = max([ e; c ] );
    e = e ./ m;
    c = c ./ m;
    % Cálculo do resíduo
    epsilon = 1E-6;
    maxve = max(e); % maior valor experimental normalizado

```



```
R = 0;
q = 0;
for i=1:length(te)
    if (e(i) > epsilon * maxve) % não precisa verificar se (e(i)~= -1)
        dist = ((c(i)-e(i))./((c(i)+e(i))/2)).^2;
        R = R + dist;
        q = q + 1;
    end
end
R = R / q;
end
```