### Proposta de novos métodos para a estimação de parâmetros em equações diferenciais ordinárias

André Thomaz Gandolpho de Mello

### Texto apresentado ao Instituto de Matemática e Estatística da Universidade de São Paulo para obtenção do título de Mestre em Ciências

### Programa: Programa de Pós-graduação em Ciência da Computação Orientador: Prof. Dr. Junior Barrera

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, novembro de 2020

### Proposta de novos métodos para a estimação de parâmetros em equações diferenciais ordinárias

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 07/10/2020. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Junior Barrera (orientador) IME-USP
- Prof. Dr. Adilson Simonis IME-USP
- Prof. Dr. Fabrício Martins Lopes UTFPR

# Agradecimentos

O presente trabalho não teria sido possível sem o apoio de diversas pessoas ao longo de seu desenvolvimento. Ressalto brevemente aqui algumas que contribuíram, de uma forma ou de outra, para que ele se tornasse uma realidade.

Agradeço em primeiro lugar a meus pais, Silvia e Goar, por todo o apoio nesta e em todas as fases de minha vida. Agradeço por me encorajarem a seguir esse caminho acadêmico e a enfrentar suas dificuldades. Agradeço igualmente a meu irmão, Luiz, pelos momentos de amizade e companheirismo. Também foram parte de minha formação meus avós maternos, Maria e Onofre, e paternos, Aparecida e Plínio, e a eles serei sempre grato. Me sinto privilegiado por poder contar com uma família tão especial.

Agradeço à minha companheira Elena por estar ao meu lado durante todos esses anos, durante os bons e maus momentos. Agradeço por todo o seu carinho e compreensão. Agradeço também aos meus amigos, novos ou de longa data, Eduardo, Gustavo, Mariana, Victor e Sarah, pelas conversas profundas e pelos momentos de descontração. Agradeço também especialmente a Anna, por estar presente em minha vida por tanto tempo, durante diversas fases, tendo inclusive influenciado em minha escolha de área de pesquisa. Deixo, por fim, meus agradecimentos a todos os amigos que não foram aqui citados, mas que de alguma forma impactaram minha jornada.

Meus profundos agradecimentos ao professor Junior Barrera, por acreditar em meu potencial, pelas ricas discussões, e por todo o apoio e incentivo durante minha formação no mestrado. Agradeço também ao professor Marcelo Reis, por sua sempre valiosa e entusiasmada participação e contribuições para este trabalho. Também sou grato pelo direcionamento dado pelo professor Walter Mascarenhas durante minha qualificação. Agradeço ao IME, ao programa de pós-graduação em Ciência da Computação, e todos os professores e funcionários, por auxiliarem em meu amadurecimento como pesquisador. Agradeço também à CAPES pelo seu papel fundamental na pesquisa brasileira e por ter financiado este trabalho.

## Resumo

MELLO, A. T. G. **Proposta de novos métodos para a estimação de parâmetros em equações diferenciais ordinárias**. 2020. 83 f. Dissertação de Mestrado - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2020.

O ramo da cinética química estuda modelos que descrevem o comportamento de reacões químicas. Para que o modelo se comporte como esperado e seja válido, é preciso que seus coeficientes de taxa de reação estejam de acordo com a realidade do fenômeno. Nesse contexto, o trabalho apresenta novas formas de realizar a estimação desses valores. Problemas de estimação de parâmetros, também denominados Problemas Inversos, são, especialmente nessa área, notoriamente difíceis por conta do mal condicionamento e não convexidade da superfície a ser otimizada. Propomos um novo método que leva em consideração propriedades das Equações Diferenciais Ordinárias, incorporando técnicas de integração numérica ao estimador. Essa abordagem visa suavizar a superfície de otimização, facilitando a convergência a seu mínimo global. Intitulado Data Shooting, verificamos que seu custo computacional supera em até 4 ordens de magnitude a alternativa clássica, denominada Single Shooting. Sua acurácia, por outro lado, mostra-se inferior para os casos mais complexos. O custo de suavizar a superfície de otimização é a introdução de um viés ao modelo, uma troca conhecida na área de aprendizado de máquina como bias-variance tradeoff. Propomos então a utilização deste método como o primeiro passo em um processo de regularização de duas etapas, desenvolvido na literatura com o objetivo de lidar com o problema de mal condicionamento de Problemas Inversos. Os proponentes deste método mostram que a regularização traz grandes benefícios. Contudo, ressaltam que para fazer uso desse método os pesquisadores devem ter a priori bons valores de referência para os parâmetros que desejam estimar. Infelizmente isso nem sempre é possível na prática. Desse modo, o segundo método desenvolvido, intitulado Data to Single Shooting, consiste em utilizar os valores estimados através do método Data Shooting, em uma primeira etapa, como os valores de referência na regularização de Tikhonov do método Single Shooting, na segunda. Mesmo não obtendo resultados tão precisos quanto a alternativa clássica em alguns dos casos de teste selecionados, o Data Shooting consegue evitar ficar preso em mínimos locais presentes na superfície gerada pelo Single Shooting, gerando portanto boas estimativas iniciais e de forma eficiente. Os resultados obtidos apontam que o método Data to Single Shooting possui uma boa performance na maioria dos casos, sendo aproximadamente 10% mais rápido que o método clássico no geral. Além disso, esse método proposto consegue solucionar problemas de maior complexidade para os quais o método clássico falha em encontrar uma resposta.

Palavras-chave: estimação de parâmetros, problemas inversos, equações diferenciais ordinárias, mal condicionamento, não convexidade, regularização

## Abstract

MELLO, A. T. G. A proposal of new methods for parameter estimation in ordinary differential equations. 2020. 83 f. Masters Dissertation - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2020.

Chemical kinetics studies models which describe the behavior of chemical reactions. For the model to act as expected and be valid, its reaction rate coefficients must be in accordance with the reality of the phenomenon. In this context, this work presents new ways of estimating these values. Parameter estimation problems, also known as Inverse Problems, are, especially in this context, notoriously difficult to deal with due to ill conditioning and the non-convexity of the optimization surface. We propose a new method that takes into account properties of Ordinary Differential Equations, incorporating techniques from numerical integration to the estimator. This approach aims at smoothing out the optimization surface, allowing for easier convergence to its global minimum. Coined Data Shooting, we have found that its computational cost are up to 4 orders of magnitude lower than the classic alternative, Single Shooting. Its accuracy, on the other hand, is inferior for more complex cases. The cost of smoothing the optimization surface is the injection of a bias to the model, an exchange known in the machine learning field as *bias-variance tradeoff*. Given such characteristics, we propose the use of this method as the first step in a two-step regularization process, developed in the literature with the purpose of dealing with the ill conditioning arising in Inverse Problems. The creators of this method have demonstrated that regularization brings notable benefits when dealing with parameter estimation problems. However, they note that in order to use this method researchers must have good a priori reference values for the parameters they wish to estimate. Unfortunately, this is not always possible in practice. Thus, the second method we present, entitled Data to Single Shooting, consists of using the estimated values from Data Shooting, in a first step, as such reference values in a Tikhonov regularization of the Single Shooting method, on the second step. Even though Data Shooting does not generate as accurate results as the alternative classical method in some of the selected test cases, it is generally able to avoid getting stuck in local minima present on the optimization surface generated by Single Shooting. With that, Data Shooting is able to provide good initial estimates, in an efficient manner. The results we have obtained show that the Data to Single Shooting method has a good performance in most cases, being approximately 10% faster than the classic method overall. Moreover, this new method is able to solve more complex problems for which the classic method fails to find an answer.

**Keywords:** parameter estimation, inverse problems, ordinary differential equations, ill conditioning, non convexity, regularization

# Conteúdo

Li	sta d	le Figuras	ix			
Li	sta d	le Tabelas	xi			
1	Intr	rodução	1			
	1.1	Motivação	1			
	1.2	Objetivos	2			
	1.3	Organização do Trabalho	2			
2	Fun	Fundamentos Teóricos				
	2.1	Equações Diferenciais Ordinárias	5			
		2.1.1 Equações Diferenciais de Primeira Ordem	5			
		2.1.2 Integração Numérica	$\overline{7}$			
	2.2	Cinética Química	13			
		2.2.1 Reações Elementares	13			
		2.2.2 Reações Complexas	14			
	2.3	Estimação de Parâmetros	17			
		2.3.1 Mínimos Quadrados	17			
		2.3.2 Otimização	18			
		2.3.3 Regularização	21			
3	Métodos Propostos					
	3.1	Métodos de Shooting				
	3.2	Data Shooting	29			
	3.3	Data to Single Shooting	34			
4	Materiais e Implementação 37					
	4.1	Detalhes de Implementação	37			
		4.1.1 Simulação de Experimentos	37			
		4.1.2 Seleção de Hiperparâmetros	38			
		4.1.3 Avaliação de Modelos	40			
	4.2	Casos de Teste	45			
		4.2.1 First Order Irreversible Chain Reaction (FIC)	45			
		4.2.2 First Order Reversible Chain Reaction (FRC)	46			
		4.2.3 Catalytic Cracking of Gas Oil (CCGO)	47			

	4.2.4 Bellman's Problem (BEP) $\ldots$					
		4.2.5	Methanol-to-Hydrocarbons Process (MHP)	49		
		4.2.6	Lotka-Volterra's Problem (LVP)	50		
		4.2.7	Biomass Batch Growth (BBG)	51		
		4.2.8	FitzHugh-Nagumo (FHN)	52		
		4.2.9	Mitogen Activated Protein Kinases (MAPK)	53		
	4.3	Tecno	logias	55		
		4.3.1	Software e Linguagem	55		
		4.3.2	Hardware e Sistema Operacional	55		
5	$\mathbf{Res}$	ultado	s e Discussão	57		
	5.1	Comp	aração dos Métodos	57		
		5.1.1	Convexidade	57		
		5.1.2	Estimação	64		
	5.2	Códig	o Fonte	77		
6	Cor	Conclusões				
	6.1	Síntes	e do Trabalho	79		
	6.2	Contri	buições	81		
	6.3	Traba	lhos Futuros	82		
A	Mét	tricas		83		
Bi	Bibliografia 8					

# Lista de Figuras

2.1	Ilustração do Método de Euler.	8
2.2	Impacto do passo no Método de Euler	9
2.3	Sistema de Robertson.	12
2.4	Exemplo de superfície de otimização	19
2.5	Exemplos de superfícies de otimização em três dimensões	21
9.1	Ilustra são do mátodo Cingle Chosting	97
ა.1 ვე	Ilustração do método Single Shooting.	21
ა.⊿ ეე	Instração do método Multiple Shooting.	20 91
ა.ა ე_4	Ilustração do metodo Data Shooting	01 91
5.4 2.5	Dis manuel de méter de Data En Chasting	91 97
3.5	Diagrama do metodo Data to Single Snooting	35
4.1	Processo de obtenção dos dados.	38
4.2	Processo de LOOCV	39
4.3	Ilustração do critério de sucesso segundo literatura	41
4.4	Ilustração do critério de sucesso adotado	42
4.5	Gráfico de Success Rate e tempo computacional.	43
4.6	Solução do PVI para o problema FIC.	45
4.7	Solução do PVI para o problema FRC	46
4.8	Solução do PVI para o problema CCGO	47
4.9	Solução do PVI para o problema BEP	48
4.10	Solução do PVI para o problema MHP	49
4.11	Solução do PVI para o problema LVP.	50
4.12	Solução do PVI para o problema BBG	51
4.13	Solução do PVI para o problema FHN	52
4.14	Solução do PVI para o problema MAPK	54
F 1		<b>F</b> 0
5.1 5.9	Comparação de curvas de otimização para problema exponenciai	50
5.2 5.2	Comparação de curvas de otimização para problema FIC.	09 61
5.3	Comparação de curvas de otimização para problema LVP	61
5.4	Comparação de curvas de otimização para problema LVP com mais dados	62
5.5	Comparação de curvas de otimização para problema FHN.	63
5.6	Comparação entre curvas estimadas e dados experimentais para o problema CCGO.	65
5.7	Comparação entre curvas estimadas e dados experimentais para o problema MHP.	65
5.8	Avaliação de modelos para o problema FIC	66

### x LISTA DE FIGURAS

5.9	Avaliação de modelos para o problema FRC.	67
5.10	Avaliação de modelos para o problema CCGO.	67
5.11	Avaliação de modelos para o problema BEP	68
5.12	Avaliação de modelos para o problema MHP	68
5.13	Avaliação de modelos para o problema LVP	69
5.14	Avaliação de modelos para o problema BBG.	70
5.15	Avaliação de modelos para o problema FHN	70
5.16	Avaliação de modelos para o problema MAPK	71
5.17	Comparação entre curvas estimadas e dados experimentais para o problema FHN $\ .$	72
5.18	Métricas.	76

# Lista de Tabelas

2.1	Relação entre ordem e velocidade de reação		14
4.1	l Parâmetros do caso de teste FIC		45
4.2	2 Parâmetros do caso de teste FRC		46
4.3	B Parâmetros do caso de teste CCGO		47
4.4	4 Parâmetros do caso de teste BEP		48
4.5	5 Parâmetros do caso de teste MHP		49
4.6	6 Parâmetros do caso de teste LVP		50
4.7	7 Parâmetros do caso de teste BBG		51
4.8	8 Parâmetros do caso de teste FHN		52
4.9	9 Parâmetros do caso de teste MAPK		54
A.1	1 Tabela com as principais métricas avaliadas para todos os ca	asos de teste	84

### xii LISTA DE TABELAS

## Capítulo 1

# Introdução

Observar, compreender e predizer fenômenos da natureza são tarefas de grande interesse em diversos ramos da ciência, desde a biologia à economia, da química à sociologia. É comum que pesquisadores desenvolvam modelos capazes de descrever esses fenômenos com o objetivo de melhor compreendêlos. Muitas vezes esses modelos são criados com base em princípios teóricos mas dependem de parâmetros desconhecidos a priori. Determinar bons valores para esses parâmetros é essencial para que o modelo se comporte corretamente. É possível capturar dados do fenômeno estudado para que sejam utilizados em algoritmos capazes de estimar tais parâmetros.

No ramo da bioquímica, o estudo do comportamento de elementos químicos em organismos vivos é de central importância. Um dos focos de cientistas nessa área é, por exemplo, verificar o resultado de interações entre sistemas biológicos e medicamentos, buscando verificar sua eficácia e possíveis efeitos colaterais. Essas interações podem ser modeladas como cadeias de reações provenientes da área da cinética química. Essas reações, por sua vez, podem então ser descritas matematicamente através de sistemas de Equações Diferenciais Ordinárias (EDOs).

O presente trabalho introduz dois novos métodos para a estimação de parâmetros. Ambos foram desenvolvidos especificamente para a utilização em problemas envolvendo EDOs, tendo reações da cinética química como contexto. Nossa abordagem se ancora na teoria dos métodos numéricos para solução de EDOs, em conjunto com a teoria dos Mínimos Quadrados. Desse modo, atacamos o problema de estimação por uma ótica numérica e determinística.

O primeiro método, denominado Data Shooting (DS), possibilita a geração de problemas de otimização com uma superfície suave ou até mesmo convexa em casos particulares. Em contrapartida, ele acrescenta um considerável viés ao modelo. Isso pode levar o otimizador a não encontrar os parâmetros ideais para o problema em alguns casos.

Contudo, mostramos que DS pode ser utilizado em conjunto com modelos clássicos, dentro de um esquema de regularização do problema original. Essa abordagem dá origem ao método intitulado Data to Single Shooting. Mostramos que esse segundo método pode trazer melhorias significativas em questão de performance e poder de generalização em comparação com o método clássico de Single Shooting [Ham11, HLDVI12, SB13].

### 1.1 Motivação

A estimação de parâmetros é um problema também conhecido como Problema Inverso (PI). Diz-se inverso pois mais comumente temos um modelo conhecido e, partir deste, dados artificiais podem ser gerados. Já em um PI, temos os dados, coletados empiricamente, e a partir destes desejamos encontrar o modelo que melhor descreve o fenômeno que os produziu.

Para se tratar de Problemas Inversos na cinética química é preciso, em primeiro lugar, definir

um bom modelo matemático capaz de capturar o comportamento do fenômeno estudado. Uma das formas de se fazer isso é através de sistemas de Equações Diferenciais Ordinárias (EDOs). Uma vez obtido esse modelo, é possível utilizar técnicas de integração numérica para simular o comportamento das variáveis de estados que descrevem o fenômeno. Esse problema é comumente chamado de Problema de Valor Inicial (PVI).

A solução do PVI, portanto, nos permite construir uma função objetivo capaz de contrastar esses resultados obtidos pelo modelo com os dados adquiridos experimentalmente. Então, essa função é minimizada através de algum algoritmo de otimização, com o objetivo de encontrar os parâmetros desejados. De maneira geral, esses são os passos tomados pelo método clássico de estimação de parâmetros para EDOs denominado Single Shooting.

Apesar de muito utilizado, o método Single Shooting enfrenta dificuldades em diferentes âmbitos. Existem dificuldades não apenas em cada um dos passos descritos acima, mas também é preciso lidar com aquelas inerentes aos PIs, que os tornam notoriamente complexos. PIs sofrem especialmente com os problemas de mal condicionamento e a não convexidade.

O mal condicionamento está relacionado à não identificabilidade dos parâmetros no espaço de busca ou então a dados de baixa qualidade. Isso acarreta na instabilidade dos métodos numéricos. Já a não convexidade pode levar algoritmos de otimização a ficarem presos em mínimos locais. Esse conjunto de fatores acaba impossibilitando a identificação dos melhores parâmetros possíveis em muitos dos casos [GB15].

Para a cinética química, a obtenção de um bom modelo matemático para reações químicas é essencial. Para isso, duas etapas são cruciais. Na primeira, é necessário estudar o fenômeno afim de identificar todos os elementos químicos envolvidos e então estabelecer as relações entre eles. Isso origina a estrutura do modelo matemático. Já na segunda etapa, que é o foco deste trabalho, é preciso ajustar eventuais parâmetros desconhecidos do modelo, comumente relacionados aos coeficientes de taxa de reação.

A estimação incorreta dos parâmetros de um dado modelo pode prejudicar toda a análise do fenômeno observado. Sistemas dinâmicos, provenientes das EDOs, podem apresentar comportamentos não apenas quantitativos como também qualitativos completamente distintos na presença de parâmetros ligeiramente diferentes.

### 1.2 Objetivos

O presente trabalho visa introduzir dois novos métodos para a estimação de parâmetros em Equações Diferenciais Ordinárias, com foco em problemas da cinética química.

Denominados Data Shooting e Data to Single Shooting, esses métodos têm como objetivo contornar especificamente as principais dificuldades inerentes aos Problemas Inversos: mal condicionamento e não convexidade.

Também é objetivo deste trabalho a implementação destes métodos, bem como a realização de experimentos que comprovem empiricamente a sua eficácia e viabilidade prática.

### 1.3 Organização do Trabalho

Além da atual introdução e da bibliografia, essa dissertação conta com mais cinco capítulos. São eles: Fundamentos Teóricos (2), Métodos Propostos (3), Materiais e Implementação (4), Resultados e Discussão (5) e Conclusões (6).

O capítulo 2 apresenta conceitos fundamentais para o trabalho, tocando em pontos chave das Equações Diferenciais Ordinárias, cinética química e estimação de parâmetros. Já o capítulo seguinte,

3, trata dos novos métodos propostos, seu racional, suas propriedades e comparações com métodos clássicos, sendo o capítulo central do trabalho.

Em seguida, o capítulo 4 introduz o material e métodos utilizados ao longo do desenvolvimento deste trabalho, crucial para o entendimento dos resultados obtidos. Apresentamos os detalhes de implementação, os casos de teste selecionados e as tecnologias empregadas.

Por sua vez, o capítulo 5 trata dos resultados obtidos e realizamos uma discussão acerca deles, ressaltando as vantagens e desvantagens dos métodos propostos em comparação com modelos clássicos. O capítulo 6 finaliza essa dissertação com as conclusões, passando por uma síntese do trabalho e evidenciando as suas contribuições.

### 4 INTRODUÇÃO

### Capítulo 2

# Fundamentos Teóricos

Antes de apresentar a nova proposta, trazemos neste capítulo a revisão de conceitos utilizados como base para o desenvolvimento do trabalho. Começamos tratando do principal modelo matemático utilizado neste trabalho, as Equações Diferenciais Ordinárias. Sua estrutura bem como estratégias para solução desse tipo de equações são tratados na seção 2.1.

Dado que o projeto possui como contexto reações bioquímicas, tratamos em seguida, na seção 2.2, sobre a conexão entre a cinética química e as Equações Diferenciais Ordinárias. Fechamos o capítulo introduzindo fundamentos da estimação de parâmetros na seção 2.3, passando pela teoria dos Mínimos Quadrados, otimização e regularização.

### 2.1 Equações Diferenciais Ordinárias

Equações Diferencias Ordinárias, ou EDOs, são frequentemente utilizadas como ferramentas matemáticas para descrever fenômenos de diversas áreas da ciência. Mais especificamente, esse tipo especial de equação surge naturalmente na modelagem matemática de fenômenos químicos, como veremos na seção 2.2. Apresentamos, portanto, nessa seção seus principais conceitos e propriedades.

Trataremos destas equações focando em Equações Diferenciais Ordinárias de Primeira Ordem. Isso porque EDOs de ordem Dois e superior podem ser transformadas em equações de Primeira Ordem por meio de manipulações algébricas. Além disso, equações de Primeira Ordem são as de maior relevância para o trabalho, visto que essas originam naturalmente de equações químicas.

Em seguida tratamos de técnicas de integração numérica. Dependendo de sua estrutura, EDOs podem ser resolvidas analiticamente. Contudo, isso nem sempre é possível. Assim, métodos numéricos de aproximações precisam ser utilizados. Além disso, o conceito de integração numérica é essencial para a compreensão dos novos método de estimação proposto nesse trabalho. Aqui nos restringimos a abordar apenas a solução numérica de Problemas de Valor Inicial.

#### 2.1.1 Equações Diferenciais de Primeira Ordem

Uma Equação Diferencial é um tipo especial de equação. Ela descreve uma relação entre uma função desconhecida e uma ou mais de suas derivadas. Equações Diferenciais podem ser classificadas como Ordinárias ou Parciais. Equações Diferenciais Ordinárias são equações onde há dependência da derivada de apenas uma variável independente [Dav13]. Um exemplo ilustrativo de uma Equação Diferencial Ordinária é:

$$\frac{dx}{dt} = ax.$$
(2.1)

Nesse exemplo,  $x \equiv x(t)$  é a função real desconhecida que possui  $t \in \mathbb{R}$  como variável independente. Apesar de ser uma função de t, é comum que x seja referido como uma variável de estado, ou vetor de estado  $\mathbf{x}$ , quando representando um conjunto de variáveis de estado x. Por sua vez,  $\frac{dx}{dt}$  representa a derivada da função x com relação a t, muitas vezes denotada também como x', x'(t) ou  $\dot{x}$ . Por fim, a é uma constante da equação.

A equação 2.1 é dita uma Equação Diferencial Ordinária de Primeira Ordem. A Ordem de uma EDO é determinada pelo maior número das derivadas envolvidas. Como a equação possui apenas uma primeira derivada, então ela é dita de Primeira Ordem. Um exemplo de EDO de Terceira Ordem seria:

$$a\frac{d^{3}x}{dt^{3}} - b\frac{dx}{dt} + cx = 0.$$
(2.2)

Em muitos casos práticos, Equações Diferenciais de Primeira Ordem recebem maior atenção que EDOs de ordem maior pois é possível transformá-las em Equações de Primeira Ordem. Tomemos como exemplo a equação 2.2. É possível introduzir novas variáveis  $y \in z$  de modo a se chegar ao sistema:

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} = z$$

$$\frac{dz}{dt} = \frac{b}{a}y - \frac{c}{a}x.$$
(2.3)

Desse modo transformamos uma EDO de Terceira Ordem em um sistema de EDOs de Primeira Ordem. Um sistema de Equações Diferenciais Ordinárias possui o mesmo comportamento e propriedade de um sistema de equações convencional. Apenas agora é acrescentada a dificuldade das Equações Diferenciais, onde as próprias funções são incógnitas.

É possível resolver uma EDO analiticamente em certos casos. Voltemos ao exemplo inicial 2.1. Multiplicando ambos os lados da equação por dt e integrando, obtemos:

$$x = \int (ax)dt \tag{2.4a}$$

$$= ce^{at}.$$
 (2.4b)

Temos, portanto, uma integral indefinida em 2.4a, que pode ser calculada. Ela resulta em 2.4b, na qual c representa a constante de integração. Dessa forma, 2.4b não representa a solução da equação 2.1, mas sim uma família de funções que respeitam as propriedades dessa equação. De fato essa é uma família que admite infinitas soluções, uma vez que admite infinitos valores para a constante c. Para que seja possível encontrar uma única solução para a Equação Diferencial original precisamos de mais informações.

Para se obter uma solução única para uma EDO ou sistema de EDOs, é necessário o que se compreende por condições auxiliares. Um tipo de condição auxiliar é denominado Valor Inicial. Voltando à equação 2.4, se temos o Valor Inicial x(t) = 10 para t = 0, podemos facilmente encontrar

$$x = 10e^{at}. (2.5)$$

Com o Valor Inicial, definimos o valor que a função desconhecida, nesse caso x(t), assume em um

momento inicial de sua variável dependente, nesse caso t = 0. Desse modo, em posse de uma EDO e um Valor Inicial, temos o que se chama de Problema de Valor Inicial (PVI), ou *Initial Value Problem* (IVP) [Pre07, SB13].

Contudo, esse é apenas um exemplo didático de uma EDO linear, estando longe de cobrir todas as formas que EDOs podem assumir. As equações contendo as funções desconhecidas podem apresentar todo o tipo de não-linearidade. Isso inviabiliza a solução analítica da equação. Por conta disso, métodos numéricos precisam ser empregados para se obter ao menos aproximações da solução real [Pre07, SB13].

### 2.1.2 Integração Numérica

A integração numérica de funções é uma ferramenta amplamente utilizada atualmente em diferentes áreas, graças ao avanço do poder computacional. Diversos métodos para integração numérica já foram propostos. É possível agrupar tais métodos principalmente em duas categorias: os Métodos de Passo Único (*one-step method*), e os Métodos de Multipasso (*multistep method*) [Pre07, SB13]. O presente trabalho explora o uso de Métodos de Passo Único para a solução de PVIs e, portanto, apenas esses serão abordados neste capítulo.

Nos chamados Métodos de Passo Único, temos a propagação de uma solução ao longo de um intervalo, combinando diversos passos ao estilo do Método de Euler. Os conhecidos Métodos de Runge-Kutta fazem parte dessa categoria. O próprio Método de Euler pode ser classificado como um método de Runge-Kutta de Primeira Ordem. Essencialmente, o próximo ponto da função é calculado a partir do ponto calculado anteriormente. O Método de Euler não só é utilizado como base para os demais métodos que compõem essa categoria, como também possui uma fácil interpretação gráfica. Por esses motivos, iniciaremos a exploração dos Métodos de Um Passo através de Euler e abordaremos os termos aqui brevemente mencionados.

Dentre os Métodos de Passo Único é possível ainda fazer mais uma distinção. Há os métodos ditos Explícitos e também os Implícitos. Abordaremos primeiro os métodos Explícitos. Eles possuem uma implementação relativamente simples que pode ser traduzida diretamente da teoria. Contudo, Métodos Explícitos não se comportam como esperado quando realizam a integração da família de Equações Diferenciais denominadas rígidas (stiff). Para integrar esses tipos de Equações é necessário o uso de técnicas mais avançadas, como por exemplo Métodos Implícitos. Esse grupo de métodos é computacionalmente mais custoso mas em troca oferece melhores aproximações para Equações Diferenciais Rígidas [Pre07, SB13]. Métodos Implícitos serão abordados logo após os Métodos Explícitos.

#### Métodos Explícitos

Começamos a introdução dos métodos de integração numérica do tipo Passo Único e Explícitos definindo a seguinte equação diferencial:

$$\frac{dx}{dt} = f(x,t). \tag{2.6}$$

No exemplo 2.1, f(x,t) = ax. Aqui utilizamos f(x,t) sem uma definição exata. Isso porque os métodos de integração que serão apresentados são gerais, independentes de f(x,t).

O Método de Euler apresenta uma forma intuitiva de realizar a aproximação da função desconhecida x. A partir de um ponto inicial é possível estimar o próximo ponto na curva x(t) dando um pequeno passo na direção de sua derivada. Isto é

$$NovoValor = ValorAtual + Curvatura \times Passo.$$

$$(2.7)$$

Felizmente, devido às propriedades das Equações Diferenciais Ordinárias, já temos acesso à curvatura dx/dt de x. Essa curvatura é definida justamente pela função f(x,t). Matematicamente escrevemos:

$$x_{i+1} = x_i + f(x_i, t_i)h, (2.8)$$

onde  $x_{i+1} \equiv x_i + h$ , sendo h denominado de passo e  $f(x_i, t_i)$  correspondendo à derivada de x avaliada nos valores  $x_i$  e  $t_i$  [CC15]. Dessa forma o Método de Euler pode prosseguir calculando iterativamente os valores da função x ao longo de um dado intervalo de tempo.



**Figura 2.1:** Ilustração do funcionamento do Método de Euler. Em azul e traço contínuo temos a real função do espaço de estados. Os pontos em azul destacados fazem parte dessa função. O tracejado vermelho representa a direção da estimação e o ponto vermelho representa o valor estimado. Esse exemplo consegue demonstrar, ao mesmo tempo, a simplicidade do método e sua principal fraqueza: o erro proveniente de um passo de integração h maior que o ideal.

A discrepância entre o valor real da função, em azul, e o estimado por Euler, em vermelho, na figura 2.1 é inerente a métodos de integração. Mais especificamente, erros como esse são decompostos em dois tipos. O primeiro deles é chamado de erro de truncamento ou erro de discretização. Esse é um erro causado explicitamente pela natureza do método utilizado para aproximação das variáveis de estado x. O segundo é denominado erro de arredondamento, e surge por conta de limitações no número de dígitos que podem ser armazenados no computador [CC15].

O erro de truncamento, por sua vez, é composto de duas partes. Há o de truncamento local, resultante da aplicação do método em um único passo, e também o de truncamento propagado, o qual surge da aproximação produzida em passos anteriores. O erro de truncamento total ou global é a soma de ambos.

O erro de truncamento local do método de Euler é da ordem  $O(h^2)$  enquanto que o erro global é O(h). Por esse critério podemos dizer que o método de Euler é de Primeira Ordem. Esse padrão se mantém para os demais Métodos de Passo Único. Um método de ordem n terá erro local da ordem de  $O(h^{n+1})$  e erro global  $O(h^n)$  [CC15].

Desse fato podemos concluir que quanto menor o passo h, menor será o erro de truncamento decorrente desses métodos. A figura 2.2 ilustra o efeito da redução do passo na integração numérica utilizando o Método de Euler. A Equação Diferencial Ordinária e o Valor Inicial que descrevem a curva azul é

$$\frac{dx}{dt} = x,$$

$$x(0) = 1,$$
(2.9)

(2.10)



que pode ser facilmente integrada, gerando a solução analítica

**Figura 2.2:** Gráfico que ilustra o impacto do tamanho do passo no Método de Euler. A curva em azul representa a função  $e^t$ , ao passo que as curvas amarela, vermelha e verde representam aproximações dessa função com diferentes valores de passo h. Quanto maior o passo, maior o erro de truncamento.

A família de Métodos de Integração Numérica de Passo Único mais utilizada é a família dos Métodos de Runge-Kutta. É comum encontrar alusões ao "Método" de Runge-Kutta, em referência a uma fórmula única e não à família de possibilidades. Nesse caso, geralmente se referem do método clássico de Runge-Kutta de Quarta Ordem, que assume a forma

$$x_{i+1} = x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h;$$
  

$$k_1 = f(x_i, t_i)$$
  

$$k_2 = f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}k_1h\right)$$
  

$$k_3 = f\left(t_i + \frac{1}{2}h, x_i + \frac{1}{2}k_2h\right)$$
  

$$k_4 = f(t_i + h, x_i + k_3h).$$
  
(2.11)

Esse é apenas um exemplo de método que faz parte da família de Runge-Kutta [Pre07]. Através dos Métodos de Runge-Kutta é possível gerar integradores não apenas de Quarta Ordem, como em 2.11, como também de Primeira, Segunda, *n*-ésima Ordem. O próprio Método de Euler pode ser classificado como um Método de Runge-Kutta de Primeira Ordem. A forma geral que assumem os métodos dessa família é

$$x_{i+1} = x_i + \vartheta(x_i, t_i, h)h, \qquad (2.12)$$

onde

$$\vartheta(x_i, t_i, h) = \sum_{j=1}^n a_j k_j, \qquad (2.13)$$

e por sua vez

$$k_{1} = f(x_{i}, t_{i})$$

$$k_{2} = f(x_{i} + q_{11}k_{1}h, t_{i} + p_{1}h)$$

$$k_{3} = f(x_{i} + q_{21}k_{1}h + q_{22}k_{2}h, t_{i} + p_{2}h)$$

$$\dots$$

$$k_{n} = f(x_{i} + q_{n-1,1}k_{1}h + q_{n-1,2}k_{2}h + \dots + q_{n-1,n-1}k_{n-1}h, t_{i} + p_{n-1}h)$$
(2.14)

Desse modo percebe-se que a família de Métodos de Runge-Kutta pode dar origem a diversos métodos, dependendo de sua Ordem. Mas mais do que isso, uma mesma Ordem pode dar origem a métodos diferentes. Isso decorre dos diferentes possíveis valores que os coeficientes  $p_j e q_{j,k}$  podem assumir.

Tomamos como exemplo o Método de Heun e o Método do Ponto Médio: ambos são classificados como métodos de Runge-Kutta de Segunda Ordem devido ao seu erro global de truncamento  $O(h^2)$ . Contudo, uma diferente combinação de coeficientes dá origem a métodos ligeiramente diferentes [Pre07].

O Método de Heun utiliza os coeficientes  $a_2 = a_1 = 1/2$ ,  $p_1 = q_{11} = 1$  e assume a forma

$$x_{i+1} = x_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h;$$
  

$$k_1 = f(x_i, t_i)$$
  

$$k_2 = f(x_i + k_1h, t_i + h),$$
  
(2.15)

ao passo que o Método do Ponto Médio utiliza  $a_1 = 1, a_2 = 1, p_1 = q_{11} = \frac{1}{2}$  e obtém

$$x_{i+1} = x_i + k_2 h;$$
  

$$k_1 = f(x_i, t_i)$$
  

$$k_2 = f\left(x_i + \frac{1}{2}k_1 h, t_i + \frac{1}{2}h\right).$$
  
(2.16)

É importante ressaltar que esses coeficientes não são escolhidos arbitrariamente. Na realidade eles são obtidos levando-se em conta propriedades da expansão da série de Taylor até o termo correspondente à Ordem desejada. Para determinar os coeficientes de um Método de Runge-Kutta de Ordem Dois, por exemplo, é preciso recorrer à expansão de Taylor até o segundo grau [Dav13].

Note que quanto maior a Ordem desses métodos, maior é a sua demanda computacional, i.e., existe um preço pela redução do erro de truncamento. Cada uma das equações  $k_j$  depende da avaliação da função  $f(x_i, t_i)$ . Desse modo, uma equação de ordem n precisará avaliar a função  $f(x_i, t_i)$  um total de n vezes, tornando o processo cada vez mais dispendioso.

Existem, desse modo, diversos métodos dentro da família dos Métodos de Runge-Kutta. O método clássico de Runge-Kutta de Quarta Ordem é o mais amplamente utilizado por apresentar um bom balanço entre erro de truncamento e custo. Os ganhos em precisão apresentados por métodos de Ordem maior do que quatro, dependendo da aplicação, podem não compensar a complexidade e custo computacional adicionais [CC15].

### Métodos Implícitos

Uma vez cobertos todos os conceitos fundamentais através dos Métodos Explícitos, podemos avançar na definição dos Métodos Implícitos de integração numérica. Mas antes convém melhor definir sua motivação, a solução de Equações Diferenciais Rígidas.

A *rigidez* (*stiffness*) é uma propriedade que aparece com maior frequência quando temos mais de uma Equação Diferencial Ordinária em mãos, ou seja, quando temos um Sistema de Equações Diferenciais. Há componentes de um sistema de EDOs que contribuem para sua rigidez, como os autovalores da matriz Jacobiana, suavidade da solução e intervalo de integração. Ainda assim, essa propriedade é de difícil definição, especialmente em termos matemáticos [HW96].

Hairer *et al.* recordam a primeira utilização e definição do termo, por C.F. Curtiss e J.O. Hirschfelder em 1952: "equações rígidas são equações nas quais certos métodos implícitos, em particular os provenientes da família Backward Differentiation Formula, apresentam resultados tremendamente melhores que aquele obtidos por métodos explícitos" [HW96] (tradução livre).

Felizmente é possível identificar EDOs rígidas através da comparação de suas constantes. Caso se verifique que alguns de seus componentes possuem taxas de variação muito rápidas ao mesmo tempo que outros componentes possuem taxas mais lentas, então provavelmente se trata de um problema rígido. Um exemplo deste cenário é o sistema de Robertson [Gob, HW96]. Robertson define as seguintes reações químicas

$$A \xrightarrow{4 \cdot 10^{-2}} B \tag{2.17a}$$

$$B + B \xrightarrow{10^4} C + B \tag{2.17b}$$

$$B + C \xrightarrow{3 \cdot 10^7} A + C, \qquad (2.17c)$$

que levam ao seguinte sistema de EDOs:

$$\frac{d[A]}{dt} = -4 \cdot 10^{-2} [A] + 10^4 [B] [C]$$
(2.18a)

$$\frac{d[B]}{dt} = 4 \cdot 10^{-2} [A] - 10^4 [B] [C] - 3 \cdot 10^7 [B]^2$$
(2.18b)

$$\frac{d[C]}{dt} = 3 \cdot 10^7 [B]^2.$$
(2.18c)

Isso indica que a reação 2.17c é extremamente rápida, seguida por 2.17b. Por outro lado a reação 2.17a é muito lenta. Essa grande diferença na magnitude das taxas dá origem a um sistema rígido. As imagens da figura 2.3 mostram o comportamento desse sistema.

Uma vez explorada a noção de sistemas rígidos vamos aos métodos implícitos. Toda a discussão feita até então em 2.1.2 trata de métodos explícitos. A diferença entre um método implicito e um explícito se torna evidente observando sua fórmula. Ao passo que o método de Euler explícito possui a forma 2.8, o método implícito assume a forma

$$x_{i+1} = x_i + f(x_{i+1}, t_{i+1})h. (2.19)$$

Nota-se, então, que o termo  $x_{i+1}$  aparece tanto do lado esquerdo quanto do lado direito da equação. Por esse motivo a denominação de método "implícito". Ao passo que o método explícito utiliza apenas o valor de  $x_i$  para calcular  $x_{i+1}$ , o método implícito usa o valor de  $x_{i+1}$  na equação para definir o seu próprio valor.

Para encontrar o valor de  $x_{i+1}$  no caso linear, é necessário isolar a variável e resolver a equação em termos de  $x_i$ . Caso contrário, se f é não-linear, então é preciso recorrer a métodos numéricos para



**Figura 2.3:** Quatro gráficos representando a evolução do sistema proposto por Robertson, cujo objetivo é oferecer um exemplo concreto de sistema rígido. A grande diferença na taxa de variação dos elementos envolvidos fica visível especialmente ao observar a evolução do elemento B em comparação aos demais. Além de apresentar valores em uma escala muito menor comparada a dos demais elementos, ele apresenta um pico acentuado, com uma variação de concentração extremamente rápida, para em seguida prosseguir para um estado estável.

se obter uma solução, como por exemplo o Método de Newton [AL07].

O processo de integração numérica através de um método implícito é, de fato, mais custoso que a alternativa explícita. Isso é especialmente verdade no caso não-linear, no qual a cada passo uma nova equação precisa ser resolvida numericamente. Contudo, métodos implícitos possuem a interessante propriedade de estabilidade incondicional [AL07].

A estabilidade é uma propriedade desejável de métodos numéricos. Por serem em sua maioria algoritmos iterativos baseados em modelos aproximados, a estabilidade está relacionada à sua capacidade de não divergir do resultado real ao longo das iterações. No caso específico de métodos de integração, os métodos implícitos são incondicionalmente estáveis, ao passo que métodos explícitos são condicionalmente estáveis.

Desse modo, a estabilidade dos métodos explícitos é condicionada ao tamanho do passo h utilizado na integração. Isto é, h deve ser menor que um dado limiar para que o algoritmo de integração não desvie do resultado esperado e gere resultados errados. Esse limiar é determinado pelas propriedades da Equação Diferencial sendo integrada [AL07].

Em sistemas rígidos, onde há a presença de elementos que evoluem muito rapidamente em conjunto com elementos de evolução muito lenta, o valor de h é limitado pelo elemento de evolução rápida. Desse modo, h tem de ser tão pequeno para garantir a estabilidade que se torna inviável a avaliação do sistema de equações de interesse no intervalo desejado.

Por outro lado, o passo h não influencia na estabilidade dos métodos implícitos e por isso são denominados incondicionalmente estáveis. Mesmo que sejam computacionalmente custosos, essa propriedade garante que os resultados obtidos por método implícitos não sofram tanto com as aproximações inerentes ao modelo numérico. Além do exemplo de método implícito de Euler em 2.19, é possível estender essa noção para integradores de ordem maior.

### 2.2 Cinética Química

O ramo da cinética química se encarrega do estudo de reações químicas por meio da determinação de taxas de reação, os fatores que as influenciam e seus resultados em um nível molecular [TD]. Essa área trabalha com a união de dados experimentais e modelos matemáticos. Ambos são importantes para melhor explicar ou prever o comportamento de reações.

Não apenas dados de baixa qualidade podem influenciar negativamente os resultados obtidos, como também modelos mal formulados. Desse modo, iremos, nessa seção, apresentar os fundamentos da formalização matemática desenvolvida pela área para descrever fenômenos químicos.

Existem dois paradigmas matemáticos para a modelagem de reações químicas. O primeiro deles adota uma abordagem estocástica e acrescenta mudanças probabilísticas ao consumo e formação de moléculas. O segundo descreve reações através de modelos determinísticos, que levam em conta apenas a variação da concentração dos elementos e um determinado coeficiente [SSP06, Pre07].

Iremos focar no segundo paradigma, uma vez que ele apresenta modelos suficientemente robustos para corretamente descrever cenários nos quais temos concentrações elevadas de substâncias, além de possuir sólido fundamento teórico nas áreas de redes complexas, dinâmica não linear e controle de sistemas [CWLA].

### 2.2.1 Reações Elementares

Detalharemos aqui o modelo matemático que descreve a evolução de fenômenos químicos ao longo do tempo. Iremos apresentar os principais conceitos primeiramente através de reações elementares.

A vasta maioria das reações químicas de interesse não ocorrem em apenas um passo. Ao invés disso, é comum que um reagente sofra uma série de mudanças, passando por outros estados intermediários e interagindo com outros elementos químicos, até chegar ao produto desejado. Cada um desses passos moleculares intermediários são chamados de reações elementares [Car07].

Começamos por analisar uma equação química simples,

$$aA + bB \xrightarrow{k} cC + dD.$$
 (2.20)

Podemos ler essa equação como: os reagentes  $A \in B$  geram os produto  $C \in D$ . Os valores de  $a, b, c \in b$  correspondem aos valores de balanceamento estequiométrico da reação. Estamos interessados em um modelo que descreva como A e B se transformam em C e D ao longo do tempo, isto é, o quão rápido ocorre essa equação e como essa velocidade muda ao longo do tempo.

Experimentos indicam que a velocidade de uma reação pode ser medida através da multiplicação da concentração dos reagentes e uma constante, isto é,

$$\upsilon = k \prod_{l=1}^{m} [R_l]^{\gamma_{R_l}}.$$
(2.21)

Na equação 2.21  $R_l$ , l = 1, 2, ..., m corresponde aos m reagentes da equação, os colchetes em volta

dos elementos químicos indicam o valor da concentração desse elemento,  $\gamma_{R_l}$  é chamada ordem parcial dos reagentes e k é denominada constante ou coeficiente de taxa de reação [TD].

Desse modo, a velocidade da equação química 2.20 pode ser calculada por:

$$v = k[A]^{a}[B]^{b}.$$
 (2.22)

O que determina v são apenas as concentrações dos reagentes. Contudo, a velocidade é a mesma para todos os elementos da reação, determinada por v. Assim, a variação instantânea de cada um dos elementos químicos em 2.20 é:

$$\frac{d[A]}{dt} = -k[A]^{a}[B]^{b}$$

$$\frac{d[B]}{dt} = -k[A]^{a}[B]^{b}$$

$$\frac{d[C]}{dt} = k[A]^{a}[B]^{b}$$

$$\frac{d[D]}{dt} = k[A]^{a}[B]^{b}.$$
(2.23)

Com isso, temos um sistema de Equações Diferenciais Ordinárias (EDOs), o qual descreve o comportamento dinâmico das espécies químicas envolvidas.

A mudança na concentração de cada elemento depende de sua participação na reação. Reagentes tem uma taxa de variação negativa, pois são consumidos ao longo do tempo, ao passo que produtos possuem uma taxa positiva, uma vez que são formados.

Reações elementares podem ser classificadas pela sua ordem de reação. A ordem da reação é sempre igual a molecularidade da reação elementar, que indica a quantidade de moléculas que atuam como reagentes. A tabela 2.1 apresenta como a ordem da reação pode influenciar o cálculo de sua velocidade. Exemplos de reações de ordem zero até dois são fornecidas, uma vez que reações elementares de ordem três ou maior são raras na prática [Car07].

	Equação Química	Equação diferencial	Equação integral
Ordem Zero	$A \longrightarrow Produtos$	d[A]/dt = -k	$[A] = [A]_0 - kt$
Ordem Um	$A \longrightarrow Produtos$	d[A]/dt = -k[A]	$[A] = [A]_0 e^{-kt}$
Ordem Dois	$2 \operatorname{A} \longrightarrow \operatorname{Produtos}$	$d[A]/dt = -k[A]^2$	$\frac{1}{[A]} = \frac{1}{[A]_0} + kt$
	$A + B \longrightarrow Produtos$	d[A]/dt = -k[A][B]	$\frac{1}{[A]_0 - [B]_0} \ln\left(\frac{[A][B]_0}{[B][A]_0}\right) = kt$

Tabela 2.1: Relação entre ordem e velocidade de reação.

### 2.2.2 Reações Complexas

Como mencionado anteriormente, a grande maioria das reações não ocorrem em apenas um passo. Por conta disso apenas uma reação elementar não será suficiente para descrever certos fenômenos de interesse. Muitas vezes se faz necessário o uso de reações complexas. Elas podem ser agrupadas em: reações consecutivas, paralelas ou reversíveis [AFB06].

#### Reações paralelas

Reações paralelas surgem quando mais de uma reação envolvendo os mesmos reagentes e/ou produtos ocorrem simultaneamente em um mesmo meio. Por exemplo:

$$\begin{array}{c} \mathbf{A} \xrightarrow{k_1} \mathbf{B} \\ \mathbf{A} \xrightarrow{k_2} \mathbf{C}. \end{array} \tag{2.24}$$

A velocidade de cada uma das reações é determinada a partir de seus respectivos reagentes, seguindo as regras das reações elementares:

$$\frac{d[A]}{dt} = -k_1[A] - k_2[A]$$

$$\frac{d[B]}{dt} = k_1[A]$$

$$\frac{d[C]}{dt} = k_2[A].$$
(2.25)

Os cálculos das velocidads permanecem inalterados, a diferença se encontra em seu uso para definir a variação do elemento A. Nesse caso o elemento químico A é reagente em ambas as equações que compõem 2.24. Por esse motivo, seu decaimento é influenciado pela velocidade de ambas as reações.

#### Reações consecutivas

Reações consecutivas decorrem da transformação de um reagente em um produto intermediário, que por sua vez é tido como reagente em uma outra equação, da qual outro produto emerge.

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \tag{2.26}$$

As mesmas regras se aplicam para esse caso na determinação das velocidades de reação e seu uso nas Equações Diferenciais Ordinárias que são originadas.

$$\frac{d[A]}{dt} = -k_1[A] 
\frac{d[B]}{dt} = k_1[A] - k_2[B] 
\frac{d[C]}{dt} = k_2[B].$$
(2.27)

A primeira reação de 2.26, especificada pela letra  $k_1$  sobre a flecha da reação, tem o elemento químico A como reagente e B como produto. Desse modo, a velocidade dessa reação é determinada apenas pelo elemento químico A e o consumo de A é determinado por essa taxa. A segunda reação, por sua vez, tem B como reagente e C como produto. Portanto a velocidade da equação é calculada a partir do reagente B, enquanto C é formado com essa velocidade. Por fim, o elemento químico B, ao participar tanto como reagente quanto como produto em diferentes reações, está sujeito à velocidade de produção e de consumo de cada uma das taxas das equações.

#### Reações Reversíveis

Em uma reação reversível temos duas reações elementares nas quais os produtos de uma reação são os reagentes da outra e vice-versa. Tomamos como exemplo

$$A \xrightarrow[]{k_1} B, \qquad (2.28)$$

de onde extraímos as EDOs:

$$\frac{d[A]}{dt} = -k_1[A] + k_2[B]$$

$$\frac{d[B]}{dt} = k_1[A] - k_2[B].$$
(2.29)

O sistema de equações pode ser facilmente obtido seguindo as regras que definem a velocidade de reações. É possível interpretar reações reversíveis como duas reações paralelas, com a peculiaridade dos elementos serem ao mesmo tempo reagentes e produtos.

#### **Reações Enzimáticas Simples**

Enzimas são um tipo especial de catalisadores biológicos de grande interesse na bioquímica. As taxas de reações catalisadas por enzimas chegam a ser de  $10^6$  até  $10^{12}$  vezes maiores que as respectivas reações sem sua presença. Apesar dessa mudança notável, ainda é possível modelar a atuação de enzimas por meio de equações complexas, como proposto por Adrian Brown, em seu estudo de 1902 sobre a cinética de enzimas [VVP13].

Mais especificamente, uma equação química na presença de enzimas pode ser descrita por duas reações consecutivas, onde uma delas é também reversível:

$$E + S \xrightarrow[k_2]{k_2} ES \xrightarrow{k_3} E + P.$$
 (2.30)

Na representação 2.30 E corresponde à enzima, S ao substrato, ES se refere ao complexo enzimasubstrato, e P é o produto gerado. A partir dessa equação é possível extrair o modelo matemático seguindo os conceitos apresentados até então.

$$\frac{d[E]}{dt} = -k_1[E][S] + k_2[ES] + k_3[ES]$$

$$\frac{d[S]}{dt} = -k_1[E][S] + k_2[ES]$$

$$\frac{d[ES]}{dt} = k_1[E][S] - k_2[ES] - k_3[ES]$$

$$\frac{d[P]}{dt} = k_3[ES].$$
(2.31)

Com isso finalizamos a contextualização desse trabalho. Apresentamos nessa seção as principais classificações de reações presentes no estudo da cinética química e sua respectiva modelagem matemática. Passamos por reações elementares, que se dividem principalmente em reações de ordem zero, um ou dois. Em seguida abordamos reações complexas, compostas por equações paralelas, consecutivas e sequenciais, além de equações capazes de descrever a atuação de enzimas.

Em todos esses casos utilizamos a fórmula geral 2.21 para o cálculo da velocidade v da reação. Em suma, a velocidade de uma reação é determinada pela multiplicação entre todos os reagentes da equação química e a sua constante k.

O cálculo da variação instantânea dos elementos químicos também pode ser facilmente descrito. Essencialmente, a variação de um elemento químico corresponde à soma das velocidades de reação das equações onde atua como produto, menos a soma das velocidades de reação onde atua como reagente.

### 2.3 Estimação de Parâmetros

Outra área de grande importância para este trabalho é a dos estudos sobre estimação de parâmetros. Iniciaremos abordando problemas do tipo Mínimos Quadrados (MMQ). Essa área está intrinsecamente relacionada à área de otimização, a qual também iremos cobrir brevemente nessa seção. Por fim, uma subseção se dedica à apresentação do conceito de regularização, por se tratar de uma ferramenta essencial para o entendimento do modelo proposto.

### 2.3.1 Mínimos Quadrados

Compreender fenômenos da natureza é um dos papeis fundamentais da ciência. Muitas disciplinas se debruçam sobre o difícil trabalho de desenvolver modelos que representem a realidade, para então estudar seu comportamento quantitativa e/ou qualitativamente. Esse trabalho se inicia tipicamente por definir interações entre variáveis de interesse, estabelecendo uma relação entre elas, sendo categorizadas entre variáveis dependentes e independentes.

É possível amparar parte da modelagem em princípios teóricos. Como apresentamos na subseção anterior, que introduz a modelagem no contexto da cinética química, podemos, através da teoria, formular boa parte do modelo matemático que descreve fenômenos químicos. Contudo, neste e em muitos outros casos, restam parâmetros que não podem ser determinados apenas por meio de princípios teóricos. Por esse motivo, estudos empíricos e a coleta de dados são essenciais para determinar os parâmetros desconhecidos.

Este é um cenário tão comum a tantas áreas, que diferentes nomes são dados para este mesmo tipo de problema: desde análise de regressão, por parte dos estatísticos, à estimação de parâmetros, filtragem ou identificação de processos por parte de engenheiros. Em última instância, o problema pode ser reduzido à utilização de uma mesma técnica como base, o Método dos Mínimos Quadrados (MMQ) [LH95].

Tendo em mãos um modelo e dados de determinado fenômeno, o primeiro passo para a aplicação do MMQ é a definição da função objetivo. Minimizar essa função corresponde a encontrar os parâmetros que geram o modelo que melhor se ajusta aos dados experimentais. De maneira geral, a função objetivo  $Q_{LS}$  possui a forma exibida na equação 2.32.

Ressaltamos que essa equação possui fortes raízes na estatística. Problemas de minimização de  $Q_{LS}$  em geral podem ser reformulados como problemas de maximização da verossimilhança dos dados (maximum-likelihood estimation). Para que isso seja verdade, é preciso assumir algumas propriedades sobre os dados, como, por exemplo, que os erros de medição sejam independentes e distribuídos de acordo com uma curva normal de desvio padrão constante [Pre07]. Utilizamos essa suposição em nossos estudos. Seguimos, contudo, com a formulação do problema sob o arcabouço dos Mínimos Quadrados por ser uma peça fundamental para os métodos propostos, como ficará claro no capítulo 4.

$$Q_{LS}(\Theta) = \sum_{i=1}^{i=n} w_i r_i^2 = \sum_{i=1}^{i=n} w_i (y_i - \tilde{y}_i)^2 = \sum_{i=1}^{i=n} w_i (y_i - f(x_i; \theta_1, \theta_2, \dots, \theta_n))^2$$
(2.32)

Em 2.32, n representa o número de observações dos dados que possuímos,  $y_i$  corresponde à *i*-ésima observação da nossa variável dependente, enquanto que  $\tilde{y}_i$  simboliza o correspondente *i*-ésimo valor calculado da variável dependente.  $r_i$  por sua vez é denominado resíduo e representa a diferença entre o valor real (obtido empiricamente) e o valor calculado pelo modelo. Já  $x_i$  representa a *i*-ésima observação da variável independente. Pode ser interpretado tanto como um escalar, na presença de apenas uma variável, ou um vetor, no caso de múltiplas variáveis independentes. Chamamos f de função estimadora. Essa é a função que expressa a relação entre as variáveis x e y, ou seja, o modelo que captura o fenômeno. Já  $w_i$  é um escalar que representa o peso dado a *i*-ésima observação.

O vetor  $\Theta$  corresponde aos parâmetros  $\theta_i$ , i = 1, 2, ..., n do modelo que desejamos estimar. Por brevidade, iremos representar a função f que depende de tais parâmetros omitindo esse vetor, de modo que  $f(x) \equiv f(x; \theta_1, \theta_2, ..., \theta_n)$ .

As escolha dos pesos  $w_i$  de cada uma das observações está vinculada à incerteza da variável em questão. Essa incerteza pode ser capturada através do desvio padrão  $\sigma_{x_i} \in \sigma_{y_i}$  das variáveis. Alternativamente, é plausível utilizar  $w_i = 1$  para todo *i*, na ausência de bons valores (ou estimativas) para os desvios padrões das variáveis ou então caso o desvio padrão de todas as variáveis sejam aproximadamente os mesmo. Neste trabalho iremos adotar essa estratégia, simplificando 2.32 para 2.33.

Uma discussão aprofundada sobre o método e mais especificamente acerca da definição destes pesos e sua relação à formulação estatística é feita por Wolberg [Wol06]. Utilizaremos ao longo do trabalho as suposições simplificadores necessárias e seguiremos trabalhando com o problema em sua forma 2.33, focando no problema sob a ótica de métodos numéricos.

$$Q_{LS}(\Theta) = \sum_{i=1}^{i=n} r_i^2 = \sum_{i=1}^{i=n} (y_i - \tilde{y}_i)^2 = \sum_{i=1}^{i=n} (y_i - f(x_i))^2$$
(2.33)

Como já mencionado, nem sempre estaremos trabalhando com apenas uma variável dependente. A fórmula é a mesma caso tenhamos x escalar ou vetor. Contudo, também podemos querer encontrar os melhores parâmetros para um modelo que possua mais de uma variável independente. Nesse, caso a fórmula muda ligeiramente, conforme vemos em 2.34.

$$Q_{LS}(\Theta) = \sum_{l=1}^{l=d} Q_{LS_l} = \sum_{l=1}^{l=d} \sum_{i=1}^{i=n} r_{l_i}^2$$

$$= \sum_{l=1}^{l=d} \sum_{i=1}^{i=n} (y_{l_i} - \tilde{y}_{l_i})^2 = \sum_{l=1}^{l=d} \sum_{i=1}^{i=n} (y_{l_i} - f_l(x_i))^2$$
(2.34)

Podemos adicionar ainda mais um somatório caso queiramos levar em consideração o resíduo de diferentes realizações de um mesmo experimento. Portanto, de modo a simplificar a notação, adotaremos sua representação em matrizes e vetores. Desse modo, a equação 2.34 pode ser representada sucintamente através de vetores  $R(\Theta)$  contendo todos os resíduos.

$$Q_{LS}(\Theta) = R(\Theta)^T R(\Theta) \tag{2.35}$$

Definindo a função objetivo deste modo, temos um problema que se encaixa no contexto dos Mínimos Quadrados. Minimizar essa função é equivalente a minimizar os resíduos e como resultado encontramos os parâmetros que melhor ajustam o modelo aos dados. Para prosseguir na resolução desse problema, a seguir serão apresentados alguns conceitos fundamentais acerca da otimização numérica de funções.

### 2.3.2 Otimização

Quando a função objetivo definida para o problema que desejamos tratar é linear nos parâmetros da equação, temos um problema de Mínimos Quadrados Lineares (MQL). Isso possibilita a solução do problema de forma analítica, bastando resolver o sistema de equações através de técnicas da Álgebra Linear. Contudo, nem sempre esse é o caso.

É possível que o fenômeno observado não possa ser modelado de forma linear nos parâmetros, necessitando da flexibilidade de equações não lineares para corretamente capturar seu comportamento. Nesse cenário, a solução de problemas de Mínimos Quadrados Não Lineares (MQNL) depende da aplicação de métodos de otimização numérica.

Para solucionar um problema de otimização, buscamos encontrar os valores dos parâmetros que minimizam ou maximizam o valor de uma determinada função. No caso de um problema de estimação de parâmetros, desejamos encontrar o mínimo da função objetivo. No cenário de MQNL, contudo, a função objetivo pode contar com diversos mínimos.

É preciso fazer uma distinção entre mínimos locais e mínimos globais. A figura 2.4 ilustra essa diferença. Em geral, mínimos locais são resultados sub ótimos para o problema de otimização. O que desejamos obter de fato é o valor do mínimo global da função, para que tenhamos os melhores parâmetros possíveis.



**Figura 2.4:** Figura ilustrando uma possível superfície de otimização. Adaptado de Antoniou e Lu [AL07]. O pontos A e E correspondem ao limite de busca da otimização, sendo considerados mínimos locais nesse caso. Os pontos B e D representam mínimos locais enquanto que C seria o mínimo local de f(x) nesse espaço de busca.

O pseudoalgoritmo 1 descreve o funcionamento geral de algoritmos de otimização. Em suma, um valor inicial  $\Theta_0$  é iterativamente atualizado, comumente utilizando-se informações do gradiente da função, de modo a levar esse valor inicial em direção ao mínimo da função.

Nesse algoritmo, f se trata da função que desejamos otimizar,  $\Theta$  corresponde ao vetor de parâmetros  $\theta_i, i = 1, 2, \ldots, n$  a serem otimizados, i.e.  $\Delta \Theta_k^T = [\Delta \theta_1, \Delta \theta_2, \ldots, \Delta \theta_n]_k$ . Já  $\varepsilon_{\Theta}$  e  $\varepsilon_F$  são valores de tolerância para o problema de otimização. Eles são utilizados para verificar a convergência da função. Por fim, maxIter, em conjunto com os valores de tolerância, controlam o número de iterações desse processo. Os valores retornados  $\Theta^*$  e  $F^*$  correspondem ao minimizador da função e mínimo valor da função objetivo, respectivamente. Essa dupla de valores constitui a solução do problema de otimização.

Note portanto que o resultado do algoritmos depende do valor inicial  $\Theta_0$ . Dependendo das propriedades da função f, diferentes valores iniciais podem levar a diferentes resultados. Portanto, é possível que o algoritmo convirja para um mínimo local, e não um mínimo global. Isso porque o

Algorithm 1 Algoritmo geral de otimização. Adaptado de Antoniou e Lu [AL07].

```
1: procedure OPTIMIZER(f, \Theta_0, maxIter, \varepsilon_{\Theta}, \varepsilon_F)
           k \leftarrow 0
 2:
 3:
           convergence \leftarrow False
           while k \leq maxIter AND NOT convergence do
 4:
                 k \leftarrow k+1
 5:
                 \Delta \Theta_k \leftarrow computeDelta(f, \Theta_{k-1})
 6:
 7:
                 \Theta_k \leftarrow \Theta_{k-1} + \Delta \Theta_k
                 F_k \leftarrow f(\Theta_k)
 8:
                 \Delta F_k \leftarrow F_{k-1} - F_k
 9:
                 convergence \leftarrow checkConvergence(\Delta \Theta_k, \varepsilon_{\Theta}, \Delta F_k, \varepsilon_F)
10:
           end while
11:
           \Theta^* \leftarrow \Theta_k
12:
           F^* \leftarrow f(\Theta^*)
13:
           return \Theta^*, F^*
14:
15: end procedure
```

pseudoalgoritmo 1 descreve um algoritmo genérico da família dos otimizadores locais. Se desejamos encontrar o mínimo global da função, ao invés de apenas um mínimo local, outras estratégias precisam ser adotadas.

Um das possíveis estratégias para se obter o mínimo global de uma função, ainda fazendo uso de um otimizador local, seria executar esse algoritmos diversas vezes, mas garantindo que cada execução utilize um valor inicial  $\Theta_0$  diferente. Desse modo, espera-se que eventualmente algum desses valores esteja localizado suficientemente próximo da base de atração do mínimo global, fazendo com que o algoritmo convirja para ele. Essa estratégia é denominada Multi-Start (MS), um tipo de otimizador híbrido local-global [VFW<sup>+</sup>19].

No campo dos algoritmos de otimização exclusivamente globais, é possível distinguir aqueles ditos determinísticos dos estocásticos. Algoritmos de otimização global determinísticos apresentam garantias de convergência para o mínimo global da função. Por outro lado, essas garantias vem a um custo computacional alto, o que impede que esses métodos sejam escaláveis [BCPB<sup>+</sup>08].

Já os algoritmos estocásticos, ou probabilísticos, comumente empregam algum tipo de heurística envolvendo um processo aleatório. Recozimento Simulado (RS) ou Simulated Annealing (SA), Otimização por Enxame de Partículas (OEP) ou Particle Swarm Optimization (PSO) e Algoritmos Genéticos (AG) ou Genetic Algorithms (GA) são exemplos de algoritmos de otimização desta categoria. Eles fornecem garantias de convergência global apenas assintoticamente no melhor caso. Ainda assim, apresentam na prática bons resultados e melhor escalabilidade em comparação com as alternativas determinísticas [VFW<sup>+</sup>19].

Desses algoritmos citados, o de particular interesse para esse trabalho é o Recozimento Simulado, método que obtém inspiração da termodinâmica. O algoritmo é motivado pelo resfriamento de metal fundido e detalhes sobre sua implementação pode ser encontrados no Apêndice B do trabalho de Goffe, Ferrier e Rogers [GFR94]. Em essência, esse algoritmo se vale de alguns hiperparâmetros chave: a temperatura inicial T do sistema e a velocidade de resfriamento  $\Delta T$ .

Quando se trabalha com metal fundido, o processo de recozimento (*annealing*) visa permitir que as moléculas do metal em sua forma líquida possam circular livremente e naturalmente atingir o estado de menor energia, em sua forma cristalizada. Quando resfriado rápido demais, o metal atinge um estado amorfo ou policristalino, de maior energia [CC15].

Quanto maior a temperatura inicial T maior será o espaço de busca do algoritmo, e quanto menor a velocidade de resfriamento  $\Delta T$  mais o algoritmo irá demorar para convergir, mas teremos melhores

condições de encontrar o mínimo global. Como no recozimento, a ideia é deixar o processo ocorrer naturalmente, inclusive levando em conta a presença do comportamento estocástico das moléculas.

Desse modo, o RS pode ser entendido como um algoritmo de busca local mas com uma distinção não encontrada em algoritmos locais determinísticos. Sua característica estocástica se manifesta no fato de que ocasionalmente, durante a minimização, o algoritmo pode propositalmente mudar os parâmetros de modo a aumentar o valor da função objetivo. Com isso, espera-se que o algoritmo evite ficar preso em mínimos locais [BT<sup>+</sup>93].

O tipo de otimizador a ser utilizado depende do problema em questão. Para problemas cuja função objetivo possui um número negligenciável de mínimos locais ou é convexa, um algoritmo local é o mais indicado, devido sua garantia de convergência e sua alta performance. Por outro lado, se a função objetivo for originada de um problema altamente não linear e portanto contar com uma superfície de otimização não convexa, então a melhor escolha seria uma alternativa híbrida ou global.



Figura 2.5: Exemplos de superfícies de otimização em três dimensões.

A figura 2.4 mostra a superfície de otimização de uma função em uma dimensão. Na prática, um problema de otimização pode ter *n* dimensões. A figura 2.5 ilustra outros tipos de superfícies, agora em três dimensões. A superfície 2.5a é gerada pela função  $f(x, y) = x^2 + y^2$ , ao passo que 2.5b retrata a função

$$f(x,y) = 3 * (1-x)^2 * exp(-(x^2) - (y+1)^2) - 10 * (x/5 - x^3 - y^5) * exp(-x^2 - y^2) - 1/3 * exp(-(x+1)^2 - y^2),$$

também conhecida como função Peaks.

Apesar de conterem exemplos de funções não convexa em ambas as figuras 2.4 e 2.5, elas ainda são bem comportadas, isto é, a função objetivo que as gera é suave (contínua e diferenciável). Esse não é um cenário comum quando precisamos lidar com Problemas Inversos.

Tratamos a seguir de uma técnica muito empregada na resolução de problemas reais de otimização, denominada Regularização. Ela se torna necessária para viabilizar a convergência do algoritmo de otimização quando a função objetivo é mal condicionada.

### 2.3.3 Regularização

A função objetivo em Mínimos Quadrados Não Lineares é gerada através de equações que envolvem dados capturados experimentalmente. Esses dados frequentemente estão sujeitos a diferentes tipos de ruído, corrompendo o verdadeiro sinal do fenômeno. Esse é um dos fatores que pode dar origem

a uma função objetivo mal condicionada, dificultando a correta convergência de algoritmos de otimização [GB15].

Uma função f(x) = y é dita bem condicionada quando: i) para qualquer  $y \in Y$ , existe uma solução  $x \in X$  para qual f(x) = y; ii) a solução x é única; iii) a solução x depende continuamente do conjunto de dados y. Do contrário, temos um problema mal condicionado. O ramo de estudo de Problemas Inversos (ou estimação de parâmetros) para Equações Diferenciais Ordinárias (EDOs) é um exemplo notável de ocorrência de problemas mal condicionados [Vog86].

É possível ter uma ideia sobre o condicionamento de uma função a partir de seu número de condicionamento (*condition number*). Esse número pode ser calculado através de sua expansão de Taylor de primeira ordem. Números de condicionamento muito grandes são indício de uma função mal condicionada [CC15]. Na prática isso quer dizer que cálculos envolvendo essa função se tornam numericamente instáveis. Isto é, uma pequena mudança nos valores de entrada da função geram uma enorme diferença em sua saída.

Desse modo fica claro como uma função objetivo mal condicionada pode causar problemas para algoritmos de otimização. O algoritmo de otimização visa encontrar os valores de parâmetros que levam ao mínimo da função objetivo de forma iterativa. Por conta disso, esse algoritmo depende da convergência desses valores em sua estratégia de busca. Se a função não é estável, ou apresenta muitas oscilações, o algoritmo de otimização é comprometido em sua busca. Como forma de atenuar esse problema, a técnica de Regularização pode ser aplicada [GB15].

Recordando o problema de Mínimos Quadrados, temos a equação 2.35 que representa a função objetivo a ser minimizada. Em 2.36 temos a definição do problema que de fato desejamos solucionar, o da minimização da função objetivo. Regularizar essa função consiste em acrescentar ao problema de minimização o termo  $\alpha\Gamma(\Theta)$ , obtendo a nova equação 2.37.

$$\underset{\Theta}{\text{minimize}} \qquad Q_{LS}(\Theta) = R(\Theta)^T R(\Theta) \tag{2.36}$$

$$\underset{\Theta}{minimize} \qquad Q_R(\Theta) = Q_{LS}(\Theta) + \alpha \Gamma(\Theta) = R(\Theta)^T R(\Theta) + \alpha \Gamma(\Theta)$$
(2.37)

A função de penalização  $\Gamma$  pode assumir diversas formas. Retomando o contexto de problemas de Mínimos Quadrados Lineares, dois tipos de regressão muito comuns que utilizam regularização são a regressão Lasso e a Ridge. Enquanto o fator  $\Gamma(\Theta)$  assume a forma  $\Gamma(\Theta) = \sum |\theta| = ||\Theta||_1$  na regressão Lasso, ele é definido como  $\Gamma(\Theta) = \sum \theta^2 = ||\Theta||_2^2$  na regressão Ridge [FHT01].

Em ambos os casos a técnica de regularização possui como um dos objetivos controlar a complexidade do modelo linear, sendo o Lasso também utilizado para seleção de parâmetros. Além dessas propriedades, Hastie *et al.* [FHT01] também ressaltam que ambas abordagens são importantes para aliviar a grande variância de modelos na presença de parâmetros altamente correlacionados.

Apesar de simples e comumente substituídas por técnicas mais avançadas especialmente em problemas não lineares, as regularizações por Lasso e Ridge demonstram bem o conceito por trás da abordagem.  $Q_R$  pode ser dividido em dois termos em função de  $\Theta$ :  $Q_{LS}$  e  $\Gamma$ .

O primeiro termo mede o quão bem o modelo se ajusto aos dados. Ao otimizar  $Q_{LS}$  capturamos os padrões exibidos nos dados através do vetor de parâmetros  $\Theta$ . Já  $\Gamma$  é denominado operador de regularização e entra como uma função estabilizadora. Minimizar apenas  $\Gamma(\Theta)$  nos dá uma solução suave, estável, mas que não utiliza de forma alguma os dados experimentais. Por esse motivo  $\alpha$ , denominado parâmetro de regularização, atua como um parâmetro de *tradeoff*.

A medida que  $\alpha$  cresce,  $Q_R$  passa a ser dominado pelo termo  $\Gamma(\Theta)$ . Nesse cenário o problema se torna bem condicionado mas com um alto viés (*high bias*), essencialmente descartando o efeito
dos dados no cálculo de  $Q_R$ . Por outro lado, caso  $\alpha = 0$  temos novamente o problema original, e possivelmente mal condicionado, i.e.  $Q_{LS} = Q_R$ .

Um exemplo de regularização mais poderosa que as apresentadas anteriormente é o da regularização de Tikhonov [GB15]. A regularização Ridge pode ser vista como um caso particular de Tikhonov. A equação 2.38 é uma representação dessa forma de regularização.

$$\Gamma(\Theta) = (\Theta - \Theta^{ref})^T W^T W (\Theta - \Theta^{ref})$$
(2.38)

Na equação 2.38,  $\Theta$  é o vetor de parâmetros que desejamos estimar,  $\Theta^{ref}$  é um vetor de referência e W é uma matriz diagonal, denominada matriz de pesos. No caso específico em que W é a matriz identidade I e o vetor de referência  $\Theta^{ref}$  é o vetor nulo, temos o caso particular da regularização de Ridge.

Note que utilizando essa definição de  $\Gamma$  em nosso problema de otimização, quando  $\alpha \to \infty$ , temos que  $\Theta \to \Theta^{ref}$ . No caso da regressão Ridge, os parâmetros  $\Theta$  tendem à zero. Mais apropriadamente, podemos utilizar  $\Theta^{ref}$ , como o nome sugere, para adicionar conhecimento a priori acerca do problema em questão.

Desse modo, ao invés de  $\Theta$  tender ao vetor nulo, ele tenderá a esse vetor de referência. A matriz de peso W pode ser utilizada para indicar o quão preciso é cada um desses valores que compõem o vetor ou então para ajustar o peso de cada valor de referência de acordo com a sua magnitude.

Finalizamos, assim, o capítulo contendo os principais conceitos utilizados ao longo da tese. Tratamos da estrutura e características de Equações Diferenciais Ordinárias e como solucioná-las através de métodos numéricos. Apresentamos a conexão entre as reações químicas e EDOs e, por fim, introduzimos conceitos relativos à estimação de parâmetros. Na seguinte seção, tendo em mente essa base conceitual, damos início à descrição dos novos métodos propostos.

### 24 FUNDAMENTOS TEÓRICOS

## Capítulo 3

# Métodos Propostos

Esse capítulo é o cerne do presente trabalho. Aqui apresentamos os novos método de estimação de parâmetros para Equações Diferenciais Ordinárias. Primeiramente, na seção 3.1, iremos abordar os métodos clássicos denominados Single Shooting (SS) e Multiple Shooting (MS), começando pelas dificuldades de se estimar parâmetros em EDOs e passando por sua concepção e principais propriedades.

Em seguida, trataremos do primeiro método proposto, Data Shooting (DS), na seção 3.2. Após sua introdução, mostraremos como ele é utilizado para modificar a abordagem de regularização em duas etapas chamada Single Shooting Regularized (SSR). A essa modificação, nosso segundo método proposto, damos o nome de Data to Single Shooting (DSS) e discutimos sua aplicabilidade na seção final 3.3.

### 3.1 Métodos de Shooting

O processo de estimação de parâmetros foi coberto na seção 2.3. Aplicar esse processo quando estamos lidando com Equações Diferenciais Ordinárias (EDOs), contudo, não ocorre de forma tão direta como quando temos em mãos problemas envolvendo funções mais simples, por exemplo polinômios.

Nas Equações Diferenciais, a função que descreve diretamente o comportamento do fenômeno observado ao longo do tempo, x, é justamente uma das incógnitas. Conhecemos apenas a sua taxa de variação  $(\frac{dx}{dt}, \dot{x}(t)$  ou simplesmente  $\dot{x}$ ). Mais do que isso, essa taxa depende da função desconhecida x. Como abordado na seção 2.1, a forma típica que sistemas de EDOs assumem, em notação vetorial, é

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, \Theta). \tag{3.1}$$

Quando conduzimos experimentos, por sua vez, não temos acesso aos valores de  $\mathbf{x}$  diretamente. Na realidade, o vetor de dados coletados pode ser descrito como

$$\mathbf{y}_t = \mathbf{G}(\mathbf{x}_t, \Theta) + \xi_t, \tag{3.2}$$

onde G corresponde a uma função de observação/medição,  $\mathbf{x}_t$  corresponde ao valor observado dos estados no tempo t e  $\xi_t$  corresponde ao ruído de medição no tempo t. Temos portanto uma função discreta, dado que podemos realizar observações ou medições de fenômenos em intervalos  $\Delta t$  discretos de tempo.

O primeiro passo no processo de estimação de parâmetros é o cálculo da diferença entre  $\mathbf{y}$  e os valores estimados pelo modelo. Em outras palavras, o cálculo do resíduo. Uma das formas de se obter o vetor de estados  $\mathbf{x}$  e conseguir verificar o seu comportamento ao longo do tempo é através de métodos de integração numérica.

O problema de se gerar o vetor de estados  $\mathbf{x}$  é denominado por Problema de Valor Inicial ou *Initial Value Problem* (PVI ou IVP): encontrar  $\mathbf{x}$  que, para dado vetor de parâmetros  $\Theta$  e um valor inicial  $\mathbf{x}_0 \equiv \mathbf{x}(t = 0)$ , soluciona o sistema para todo o intervalo de interesse [VTK04]. Os principais conceitos envolvendo integração de EDOs são abordados na subseção 2.1.2.

O método Single Shooting é um dos métodos clássicos desenvolvidos para a estimação de parâmetros em EDOs. Ele consiste em solucionar um novo PVI para cada iteração do processo de otimização, utilizando o valor de  $\Theta$  da atual iteração em suas equações. Desse modo, obtemos um novo vetor de estados, utilizado como estimativa dos dados reais.

Isso é necessário pois em cada iteração do algoritmo de otimização é preciso avaliar a função objetivo, de modo a verificar a sua convergência. Mas essa função objetivo depende do cálculos dos resíduos  $R(\Theta)$ , que por sua vez dependem da diferença entre os valores experimentais e os estimados. Finalmente, a obtenção desses valores estimados, como destacamos, passa por solucionar um PVI. O pseudoalgoritmo 2 visa esclarecer as etapas desse processo.

Algorithm 2 Algoritmo para estimação de parâmetros em Equações Diferenciais Ordinárias.

```
1: procedure Optimizer(f, \Theta_0, maxIter, \varepsilon_{\Theta}, \varepsilon_F)
 2:
          k \leftarrow 0
 3:
          convergence \leftarrow False
          while k \leq maxIter AND NOT convergence do
 4:
               k \leftarrow k + 1
 5:
               \Delta \Theta_k \leftarrow computeDelta(f, \Theta_{k-1})
 6:
               \Theta_k \leftarrow \Theta_{k-1} + \Delta \Theta_k
 7:
               F_k \leftarrow f(\Theta_k)
 8:
               \Delta F_k \leftarrow F_{k-1} - F_k
 9:
               convergence \leftarrow checkConvergence(\Delta \Theta_k, \varepsilon_{\Theta}, \Delta F_k, \varepsilon_F)
10:
          end while
11:
          \Theta^* \leftarrow \Theta_k
12:
          F^* \leftarrow f(\Theta^*)
13:
          return \Theta^*, F^*
14:
     end procedure
15:
16:
    procedure Objective(\Theta)
17:
          estimated \leftarrow NumericalSolver(\Theta, ODEModel, initialValue, tRange)
18:
          sse \leftarrow 0
19:
          i \leftarrow 0
20:
          while i < length(data) do
21:
               sse \leftarrow sse + (data[i] - estimated[i])^2
22:
          end while
23:
          return sse
24:
25: end procedure
26:
27: (\ldots)
28:
    \Theta^{est}, F^{est} \leftarrow Optimizer(Objective, \Theta_0, maxIter, \varepsilon_{\Theta}, \varepsilon_F)
29:
```

Como já mencionado, o laço da linha 4 dentro da função otimizadora, avalia diversas vezes a função f a ser minimizada, na linha 8. Deixamos claro, na linha 29 que utilizamos a *Objective* como função

a ser minimizada. Em suma, ela é encarregada de calcular a soma da diferença quadrática entre dados experimentais e valores estimados. Os valores estimados são calculados na linha 18.

Seguindo a abordagem Single Shooting, os valores estimados são gerados por um integrador numérico, chamado genericamente no código de NumericalSolver. Aqui podemos utilizar, por exemplo, tanto um integrador da família dos Runge-Kutta de Quarta Ordem, quanto métodos implícitos, para lidar com problemas rígidos, dentre outros. Destacamos que para gerar esse vetor de estados estimados, a função depende dos valores dos parâmetros  $\Theta$ . Ela é recebida como entrada da função Objective.

A função NumericalSolver depende também de outras entradas, como o sistema de EDOs que descreve o fenômeno (ODEModel), valor inicial (initialValue) e intervalo de integração (tRange). De modo a simplificar o algoritmo, assumimos que eles se encontram no escopo global. Também assumimos que os dados, na variável data, utilizada na linha 21, também podem ser acessados de dentro da função.

Temos na figura 3.1 um recorte do cálculo feito na linha 22 do algoritmo. Essa etapa é referente ao cálculo do resíduo do problema de otimização. Em vermelho vemos a curva gerada pelo vetor de estados estimados a partir de determinado  $\Theta_k$ , calculado pelo otimizador no laço da linha 4. Esse valores estimados são comparados com os dados experimentais, os pontos em azul.



**Figura 3.1:** Gráfico ilustrando o método Single Shooting. Existe apenas um nó de tiro (representado pelo ponto em vermelho) e ele é o valor inicial dos estados. Representado pela linha vermelha cheia, o PVI é resolvido no intervalo completo das observações,  $[t_0, t_n]$ . A linha tracejada representa o real vetor de estados **x**, e os pontos em azul representam os dados experimentais sujeitos a ruído. As barras em preto representam o resíduo, a diferença entre os dados e os valores obtidos pela solução do PVI.

Esse método clássico é uma extensão natural da solução numérica de PVIs, conta com uma implementação simples e é muito utilizado. Contudo, ele possui suas fraquezas. Hamilton [Ham11] ressalta que essa abordagem pode ser numericamente instável. O autor aponta então para outro método da mesma família, denominado Multiple Shooting (MS), que visa solucionar esse problema.

O nome Single Shooting vem do fato deste método partir de um nó de tiro (*shooting node*) inicial e seu tiro percorrer todo o intervalo de integração, delimitado pelo tempo em que o primeiro dado foi coletado e o último,  $[t_0, t_n]$ . Em contrapartida, ao invés de utilizar apenas um nó de tiro e realizar a integração sobre todo o intervalo  $[t_0, t_n]$ , Multiple Shooting realiza diversos tiros curtos, em partições deste intervalo de tempo. A figura 3.2 representa a ideia por trás desse método.



**Figura 3.2:** Gráfico ilustrando o método Multiple Shooting. Nesse método, diversos nós de tiro (representados pelos ponto em vermelho) são utilizados, um para cada partição do intervalo  $[t_0, t_n]$ . As linhas vermelhas representam a solução do PVI de cada intervalo com seus respectivos valores iniciais. A linha tracejada representa o real vetor de estados  $\mathbf{x}$ , e os pontos em azul representam os dados experimentais sujeitos a ruído. As barras em preto representam o resíduo, a diferença entre os dados e os valores obtidos pela solução do PVI.

Esse método é capaz de atingir uma maior estabilidade em comparação com o Single Shooting, mas a um custo computacional elevado. No exemplo acima, para cada iteração do algoritmo de otimização, o método terá de resolver não um, mas três distintos PVIs. Além disso, é preciso impor restrições adicionais ao problema de otimização. Cada um dos segmentos precisa não apenas ter seus resíduos minimizados como também é preciso garantir que cada um deles forme uma função contínua ao fim do processo. Isto é, o último valor do i - 1-ésimo PVI deve coincidir com o nó de tiro do *i*-ésimo PVI, eliminando as lacunas entre os PVIs como visto na figura 3.2.

Para que seja possível atingir esse objetivo, o problema de otimização agora não se restringe apenas a encontrar os valores ótimos para o vetor de parâmetros  $\Theta$ . Para garantir a continuidade da função  $\mathbf{x}$ , é preciso otimizar também para o vetor  $\mathbf{S}$  contendo os valores de cada um dos nós de tiro utilizados pelos PVI. Teremos um nó para cada partição do intervalo de tempo  $[t_0, t_n]$ . A própria partição do intervalo é algo que pode ser otimizado, sendo um tipo de hiperparâmetro do modelo, cuja escolha não é trivial. Todos esses detalhes de implementação juntamente com seu custo elevado fazem com que esse método não seja tão amplamente utilizado quanto o SS [MHM09].

Existem hoje diversos métodos numéricos para estimação de parâmetros em EDOs. Introduzimos brevemente dois nesse capítulo: Single Shooting (SS) e Multiple Shooting (MS). Dentre esses, optamos por utilizar o método de Single Shooting como base para nossas comparações.

Como qualquer método, SS possui suas vantagens e desvantagem. Diehl e colaboradores [DBDW06] ressaltam que esse método encontra dificuldades ao lidar com sistemas instáveis e fortemente não lineares. Por outro lado, dentre suas vantagens, eles elencam seu baixo número de graus de liberdade a serem otimizados, além de sua simplicidade. Os autores destacam que isso o leva a ser frequentemente utilizado em aplicações da engenharia. Por esse motivo e dado nosso objetivo de verificar a eficácia dos métodos propostos contra um método já bem estabelecido e amplamente utilizado, seguimos com SS como *baseline* para nossas comparações.

Na seção seguinte iremos apresentar uma alternativa a ambos os métodos, fruto dos estudos deste trabalho. Esse novo método, denominado Data Shooting, visa ao mesmo tempo lidar com o problema de instabilidade numérica presente no Single Shooting e também se apresentar como uma alternativa menos custosa em comparação com o Multiple Shooting, mesmo se assemelhando a ele em alguns quesitos.

### 3.2 Data Shooting

Com apresentado na seção anterior, otimizar parâmetros utilizando métodos da família de Shootings passa por solucionar diversas vezes o Problema de Valor Inicial (PVI) posto pelas Equações Diferenciais Ordinárias (EDOs). Vamos então dar início à explicação do conceito principal por trás método proposto, intitulado Data Shooting (DS).

O nome do método faz alusão e se opõem aos método Single Shooting (SS) e Multiple Shooting (MS). A ideia é justamente realizar os tiros (*shootings*) a partir do valor de cada um dos dados coletados durante o experimento, e não a partir de apenas um valor inicial, como no Single Shooting, e nem tentar descobrir os nós ótimos para cada partição do intervalo de integração, como no Multiple Shooting.

Deixaremos esse conceito mais claro a partir do exemplo a seguir. Tomemos a reação química 3.3.

$$A \xrightarrow{\theta} B. \tag{3.3}$$

De onde obtemos a sua taxa de reação e com isso reação 3.3 pode ser descrita pelo seguinte sistema de Equações Diferenciais Ordinárias:

$$\frac{d[A]}{dt} = -\theta[A]$$

$$\frac{d[B]}{dt} = \theta[A].$$
(3.4)

Métodos de integração numérica podem então ser utilizados para resolver esse sistema. Recordando o método de Euler, ele possui a seguinte formulação:

$$x_i = x_{i-1} + f(x_{i-1}, t_{i-1}, \theta)h, (3.5)$$

com  $x_j$  representando a discretização da variável de estado x, f representando a sua derivada, e h representando o tamanho do passo de integração. Nessa equação também colocamos em evidência os parâmetros de nosso interesse  $\theta$ , o que não é comum na fórmula clássica.

Aplicando o método ao exemplo concreto, obtemos a equação 3.6

$$A_{i} = A_{i-1} - \theta A_{i-1}h B_{i} = B_{i-1} + \theta A_{i-1}h.$$
(3.6)

Em 3.6 simplificamos a notação da concentração dos elementos removendo os colchetes, de modo que  $A \equiv [A]$ . Com relação à fórmula original de Euler note que apenas mudamos os índices dos cálculos de  $i \in i + 1$  para  $i - 1 \in i$ , respectivamente. Assim, com base nessa simples correspondência apresentamos o conceito principal do Data Shooting.

O método de Euler indica a existência de uma relação entre a *i*-ésima e a (i - 1)-ésima observação dos estados desse sistema. Desse modo, assumimos ser possível estimar o valor de  $x_i$  através do valor de  $x_{i-1}$  e a relação entre eles via Euler.

Retomando a definição de dados experimentais da equação 3.2, podemos reescrever Euler como nosso estimador da seguinte forma:

$$\tilde{y}_i = y_{i-1} + f(y_{i-1}, t_{i-1}, \theta) \Delta t = \phi(\theta, y_{i-1})$$
(3.7)

Na equação 3.7 o uso do til em  $\tilde{y}_i$  indica que este é o valor estimado de  $y_i$ , ao passo que  $y_i$  se refere ao dado observado, uma função do valor teórico  $x_i$ , como descrito na equação 3.2. Note que aqui o passo h do método de integração corresponde ao intervalo entre as observações  $y_{i-1} \in y_i$ , denominado  $\Delta t_i$ . Utilizamos  $\phi$  como uma notação simplificada desse estimador.

Podemos então escrever a estimação de um dado valor de forma similar ao que temos em 3.6:

$$\tilde{y}_{A_{i}} = \phi(\theta, y_{A_{i-1}}) 
= y_{A_{i-1}} + f(y_{A_{i-1}}, t_{i-1}, \theta) \Delta t_{i} 
= y_{A_{i-1}} - \theta y_{A_{i-1}} \Delta t_{i} 
\tilde{y}_{B_{i}} = \phi(\theta, y_{B_{i-1}}) 
= y_{B_{i-1}} + f(y_{B_{i-1}}, t_{i-1}, \theta) \Delta t_{i} 
= y_{B_{i-1}} + \theta y_{A_{i-1}} \Delta t_{i}$$
(3.8)

Essa é a etapa crucial que diferencia este dos demais métodos da família dos Shootings. Ambos visam solucionar PVIs durante o processo de otimização, podendo utilizar diferentes métodos de integração para isso. O DS utiliza a hipótese simplificadora de que é possível obter boas aproximações realizando apenas um passo do integrador. Seguimos com o exemplo concreto para chegar a mais propriedades do DS.

Definimos **y** como um vetor de amostras. Esse vetor é composto por observações que correspondem às concentrações dos elementos  $A \in B$ , obtidos experimentalmente, em intervalos de tempo discretos. As observações somam N dados para cada elemento químico, totalizando 2N no geral. O intervalo entre observações é denominado  $\Delta t$  e é limitado pela precisão do equipamento utilizado nas medições. Desse modo,  $y_{A_i}$  representa a *i*-ésima observação da concentração do elemento A.

Temos por objetivo estimar o valor de  $\theta$ , parâmetro que descreve o comportamento da equação 3.4. Em posse do modelo matemático 3.6, que descreve a relação entre os elementos químicos e do vetor de amostras **y**, podemos recorrer aos Mínimos Quadrados, e definir os resíduos  $r(y_{A_i}, \theta)$  e  $r(y_{B_i}, \theta)$ como sendo:

$$r(y_{A_{i}},\theta) = y_{A_{i}} - \tilde{y}_{A_{i}}$$
  
=  $y_{A_{i}} - \phi(\theta, y_{A_{i-1}})$   
=  $y_{A_{i}} - (y_{A_{i-1}} + f(y_{A_{i-1}}, t_{i-1}, \theta)\Delta t_{i})$   
=  $y_{A_{i}} - (y_{A_{i-1}} - \theta y_{A_{i-1}}\Delta t_{i})$   
(3.9)

$$\begin{aligned} r(y_{B_{i},\theta}) &= y_{B_{i}} - y_{B_{i}} \\ &= y_{B_{i}} - \phi(\theta, y_{B_{i-1}}) \\ &= y_{B_{i}} - (y_{B_{i-1}} + f(y_{A_{i-1}}, t_{i-1}, \theta) \Delta t_{i}) \\ &= y_{B_{i}} - (y_{B_{i-1}} + \theta y_{A_{i-1}} \Delta t_{i}). \end{aligned}$$

Por fim, podemos definir a função objetivo  $Q_{LS}$  como as somas destes resíduos.

$$Q_{LS}(\theta) = \sum_{X \in \{A,B\}} \sum_{j=2}^{N} (r(y_{X_j}, \theta))^2 = R(\theta)^T R(\theta)$$
(3.10)

Apresentamos essa ideia graficamente em 3.3. A linha tracejada em azul corresponde ao verdadeiro comportamento da função estudada, ao passo que os pontos azuis representam as amostras. O ponto vermelho por sua vez representa o valor estimado a partir da metodologia proposta. A distância entre o ponto amostrado em azul e o ponto estimado corresponde ao resíduo, exemplificado na equação 3.9.



**Figura 3.3:** Gráfico ilustrando o método Data Shooting. Nesse método os nós de tiro correspondem ao dados coletados, particionando o intervalo  $[t_0, t_n]$  em n segmentos. Os pontos vermelhos representam o resultado de um passo do integrador numérico. A linha tracejada representa o real vetor de estados  $\mathbf{x}$ , e os pontos em azul representam os dados experimentais sujeitos a ruído. As barras em preto representam o resúduo, a diferença entre os dados e os valores obtidos pelo passo do integrador.



**Figura 3.4:** Representação gráfica detalhada do método Data Shooting. Os pontos em azul representam os dados coletados. O ponto em vermelho representa o resultado do cálculo de um passo do integrador numérico de Euler. A distância  $\Delta t_i$  é o tamanho do passo h do integrador e r corresponde ao resúduo, a diferença entre o dado experimental e o calculado pelo integrador.

A figura 3.3 pode se assemelhar à figura 3.2, que representa o método de MS. Assim como em MS, o intervalo  $[t_0, t_n]$  também é particionado. Contudo, dessa vez, não há necessidade de escolher como

será feita essa partição ou mesmo os valores dos nós de tiro de cada uma. Para o DS os nós de tiro são os próprios dados coletados pelo experimento e portanto a partição do tempo corresponde aos intervalos de coleta dos dados. Isso simplifica enormemente o problema de otimização. A figura 3.4 mostra em detalhes os elementos envolvidos no cálculo do resíduo para esse método.

Como sabemos, o problema clássico de minimização de Mínimos Quadrados Lineares é convexo [BBV04]. Como já abordado na seção 2.3 essa é uma propriedade muito interessante para problemas de otimização, pois com isso garantimos que todo mínimo local da função minimizada é também um mínimo global. O que diferencia um problema de Mínimos Quadrados Linear (MQL) de um problema de Mínimos Quadrados Não Linear (MQNL) é justamente a função utilizada como estimador dos dados experimentais. No caso clássico de MQL, também chamado de Regressão Linear, o estimador é justamente um polinômio de primeiro grau com relação aos parâmetros.

Ressaltamos no exemplo acima o que o estimador dos dados é a função  $\phi$ . Nesse exemplo  $\phi$  é linear com relação ao parâmetro  $\theta$ . Para que isso ocorra, basta que todas as derivadas do vetor de estado ( $\dot{\mathbf{x}}$ , também representadas por f na equação 3.5 do método de Euler) do sistema de EDOs sejam lineares com relação ao parâmetro  $\theta$  (ou ao vetor de parâmetros  $\Theta$ ). Isso é verdade uma vez que  $\phi$  é uma função linear e sabemos que a composição de funções lineares resulta em uma função linear. Desse modo, estamos apenas trocando uma função linear (a função original de MQL) por outra (a função  $\phi$ ) e com isso mantemos essa tão desejada propriedade de convexidade.

Gerar funções convexas só é possível por conta da enorme diferença entre resolver um PVI, como é feito no SS, e apenas calcular um passo de Euler, no DS. Contudo, essa vantagem de simplificar os cálculos ao utilizarmos Euler traz consigo as fraquezas desse método numérico. Como discutido na seção 2.1, o método de Euler é sensível ao tamanho do passo de integração h, podendo incorrer em um grande erro de truncamento, também chamado de erro de discretização. Por esse motivo é que métodos de ordem quatro ou maiores são utilizados na prática quando deseja-se solucionar PVIs. Há também o fato de que Euler é um método explícito e portanto não consegue lidar com problemas rígidos.

Tecnicamente seria possível substituir o passo de Euler dentro da função  $\phi$  por qualquer outro integrador numérico. Poderíamos utilizar desde um método de Runge-Kutta de Quarta Ordem, visando diminuir o erro de truncamento, até um método implícito, para lidar com problemas rígidos. Contudo, isso faria com que o método DS se tornasse consideravelmente mais custoso. Além disso, perderíamos a possibilidade de gerar funções objetivo convexas, no caso de linearidade nos parâmetros, ou mesmo simplificar enormemente (como veremos no capítulo 5, referente aos resultados) funções altamente não-lineares.

No caso da utilização de integradores implícitos, seria necessário realizar duas etapas de otimização para que o problema de estimação de parâmetros fosse solucionado. Teríamos de otimizar para  $\Theta$  e, ao mesmo tempo, em cada iteração desse processo de otimização, seria necessário também otimizar as variáveis de estado do próprio método implícito, como discutido na subseção 2.1.2.

Felizmente, durante a consideração dessas alternativas, notamos que seria possível adaptar um método implícito e de maior ordem para substituir o método de Euler e ainda assim preservar as vantagens já mencionadas acerca da performance, simplificação da função objetivo e possível convexidade. A equação 3.11 representa a equação do Método do Trapezoide Implícito, um integrador implícito da família dos métodos de Adams-Moulton [Sau12].

$$x_{i} = x_{i-1} + \frac{h}{2}(f(x_{i-1}, t_{i-1}, \theta) + f(x_{i}, t_{i}, \theta))$$
(3.11)

Por ser um método implícito o Método do Trapezoide consegue lidar com equações rígidas, além de ser um método de segunda ordem e portanto mais estável em comparação ao método de Euler. Por outro lado, já mencionamos as suas desvantagens. Ao simplesmente substituir diretamente o

método de Euler pelo Trapezoide estaríamos incorrendo em um custo muito maior para a avaliação da função objetivo, que ao invés de solucionar uma equação teria de solucionar mais um problema de otimização.

Contudo, recorde que no método de Euler adaptado como estimador, nós substituímos o valor de  $x_{i-1}$  na equação pelo valor do dado experimental. Se agora para o método do Trapezoide fizermos o mesmo não apenas para  $x_{i-1}$  como também para  $x_i$  do lado direito da equação, conseguimos chegar à equação 3.12. Portanto, o cálculo do valor estimado  $\tilde{y}_i$  para  $y_i$  depende não só do dado experimental  $y_{i-1}$ , mas também de  $y_i$ .

$$\tilde{y}_i = y_{i-1} + \frac{\Delta t_i}{2} (f(y_{i-1}, t_{i-1}, \theta) + f(y_i, t_i, \theta)) = \phi(\theta, y_{i-1})$$
(3.12)

Desse modo, por mais que não preservemos todas as propriedades do método implícito, obtemos um estimador notavelmente mais interessante. Primeiramente, estamos utilizando mais informações presentes nos dados para realizar a estimação, com base em relações bem estabelecidas pelo Método do Trapezoide. Além disso, todas as propriedades que ressaltamos para o estimador baseado em Euler se mantém para esse. Veja que se f for uma função linear em  $\theta$  também iremos gerar uma função objetivo convexa.

Com isso finalizamos a definição do método DS. Trata-se de um método que conta com uma função  $\phi$  que utiliza as relações do Método do Trapezoide Implícito para estimar os valores dos dados experimentais. Sua principal vantagem com relação ao SS é a possibilidade de gerar funções objetivo mais suaves, e até mesmo convexas quando f for uma função afim. Em comparação ao MS, o método DS se mostra muito mais simples em questão de implementação e menos custoso computacionamelmente. Finalmente, ressaltamos a seguir suas principais armadilhas e pontos negativos.

Uma das principais restrições deste método é a necessidade da observação de todas as variáveis de estado que fazem parte dos cálculos da EDO. Infelizmente isso nem sempre é viável. É comum que o fenômeno modelado não possa ter todas as suas variáveis de estado facilmente medidas na prática. Por outro lado, trabalhos como o de Guimarães [Gui16] podem ser utilizados em conjunto com esse método para aliviar essa restrição. Guimarães implementa um algoritmo que capaz de gerar formulações alternativas para EDOs que reduzem esse problema de medição de estados de difícil observação.

Devemos citar também a relação entre a suavização da função objetivo e a precisão do modelo. Esse é um exemplo clássico do *tradeoff* entre viés (*bias*) e variância (*variance*) [FHT01]. O estimador no caso de SS passa por solucionar um PVI, ao passo que no DS o estimador se resume a um passo de um integrador numérico. O modelo gerado por DS é muito mais simples que as alternativas SS e MS.

Essa simplicidade se reflete em um espaço de busca mais regular, reduzindo a variância dos possíveis resultados e aumentando proporcionalmente o viés do modelo. Apesar de facilitar a solução do problema de otimização, isso pode gerar resultados que não são suficientemente adequados, caso os valores estimado estejam muito distantes dos valores verdadeiros dos parâmetros  $\Theta$ .

Através dos casos de teste, constatamos que os valores gerados por DS apesar de próximos dos valores nominais, nem sempre eram próximos o suficiente para serem considerados como soluções do problema de estimação nos casos mais complexos. Por esse motivo, propomos a seguir a utilização do Data Shooting não como um substituto direto para os métodos clássicos necessariamente, mas também, quando necessário, como um dos passos da estratégia de estimação de parâmetros em duas etapas descrita por Gábor e Banga [GB15].

### 3.3 Data to Single Shooting

A estimação de parâmetros, ou Problemas Inversos (PI), são uma grande área de estudo. Quando esses PIs envolvem EDOs eles se tornam ainda mais desafiadores pois são mais suscetíveis a dar origem a equações não convexas e mau condicionadas [GB15].

Uma das formas clássicas de lidar com essas questões é através do uso de regularizadores, como vimos na subseção 2.3.3. Gábor e Banga verificam em seu estudo que a aplicação de métodos de regularização podem ser positivos, obtendo melhores resultados em comparação ao problema não regularizado [GB15].

$$\Gamma(\Theta) = (\Theta - \Theta^{ref})^T W^T W (\Theta - \Theta^{ref})$$
(3.13)

Os autores utilizam a regularização de Tikhonov, cuja fórmula retomamos na equação 3.13. Note que nesse método é preciso conhecer um vetor de parâmetros de referência  $\Theta^{ref}$  para que ele possa ser aplicado. Os autores descrevem três cenários envolvendo esses valores, e em cada um adotam uma estratégia diferente.

No melhor cenário temos uma boa ideia do resultado esperado, que iremos chamar de  $\Theta^{guess}$ . Nesse caso  $\Theta^{ref} := \Theta^{guess}$  e a matriz de pesos é inicializada de modo que  $W := diag(1./\Theta^{ref})$ , onde ./ é a divisão por cada elemento (*element-wise*). Desse modo, parâmetros de diferentes magnitudes podem ser controlados para que não influenciem demasiadamente no processo de penalização.

No cenário médio temos algum valor para  $\Theta^{guess}$ , mas de menor qualidade ou menor confiança. Nesse caso ainda utilizamos  $\Theta^{ref} := \Theta^{guess}$ , mas a matriz de peso é tomada como a matriz identidade, W := I. Isso é feito justamente para não amplificar o possível impacto negativo desse vetor de referência, dado que sabemos que ele é de menor qualidade.

No pior cenário, não temos nenhuma informação prévia acerca dos parâmetros  $\Theta$ . Nesse caso, aplicamos uma regularização em duas etapas. Primeiro, realizamos a estimação de parâmetros utilizando a regularização Ridge, resultando em uma estimativa inicial  $\Theta^{ridge}$ . Então, na segunda etapa, esse valor é utilizado como o vetor de referência,  $\Theta^{ref} := \Theta^{ridge}$ . Mais uma vez assumimos W := I.

Gábor e Banga entram em detalhe a respeito da troca entre variância e viés (*bias-variance tradeoff*) gerada pelo uso da regularização. Seus resultados corroboram, como esperado, que tanto o melhor cenário quanto o cenário médio ambos se saem muito melhor que o caso onde não se usa regularização.

Os resultados obtidos no pior cenário, por outro lado, são inferiores em comparação não apenas com o melhor cenário e o cenário médio, mas também quando comparado com o problema não regularizado. E infelizmente sabemos que o pior caso pode aparecer na prática mais vezes do que gostaríamos [GB15]. Pensando nisso e levando em conta os resultados obtidos pelo método de Data Shooting (DS) descrito na última seção, propomos a sua utilização quando pesquisadores se encontrarem nesse pior cenário.

Portanto, definimos agora essa nova estratégia a ser tomada no pior cenário. Ao não ter bons valores de referência para os parâmetros  $\Theta$ , ainda utilizaremos o esquema de regularização em duas etapas. Contudo, dessa vez, a primeira etapa consistirá em utilizar o regularizador Ridge em conjunto com o método DS, ao invés do tradicional SS. Desse modo, obtemos como resultado um valor  $\Theta^{DS}$ .

Após esse primeiro passo de otimização, avaliamos o modelo resultante contra o conjunto de teste. Caso o erro do modelo seja suficientemente pequeno, assumimos que caímos no melhor caso descrito anteriormente. Isto é, tomamos  $\Theta^{ref} := \Theta^{DS}$  e  $W := diag(1./\Theta^{ref})$ . Caso contrário, seguimos com o cenário médio e definimos  $\Theta^{ref} := \Theta^{DS}$ , com W := I. Na segunda etapa da regularização voltamos a utilizar o método SS. Adicionamos mais uma modificação a esse cenário. Além de ser utilizado como vetor de referência,  $\Theta^{DS}$  também é utilizado como valor inicial (*initial guess*)  $\Theta_0$  para o algoritmo de otimização na segunda etapa do processo.

A figura 3.5 é uma adaptação da arquitetura adotada pelos autores, modificada para demonstrar os passos descrito acima. A figura também representa a interação entre os principais componentes do algoritmo 2. Vemos a interação entre o otimizador global, a função objetivo e o integrador numérico.



**Figura 3.5:** Diagrama simplificado da arquitetura do método proposto. Adaptado de Gábor e Banga [GB15]. A caixa "Estimated parameters" substitui a caixa "Prior information" da arquitetura original apresentada pelos autores. Além disso, enfatizamos aqui o laço entre "Global optimization solver" e a "Estimated parameters". Esse laço, fechado pela seta verde tracejada, ressalta a obtenção dos valores de referência iniciais via DS (no caso de DSS) ou SS (no caso de SSR) e sua subsequente utilização no segundo passo da regularização.

As propriedades do método de DS fazem com que ele se encaixe muito bem nessa primeira etapa de regularização. DS faz uma troca agressiva entre variância e viés, por ser um modelo muito mais simples, podendo chegar a gerar uma função objetivo convexa. Isso vem ao custo de sua precisão. Apesar de ocasionalmente não gerar estimativas boas o suficiente para serem diretamente utilizadas como resultado final, elas se mostraram ótimos valores de referência.

Além disso, a simplicidade do modelo se reflete em seu custo computacional. A solução da sua equação de estimador é ordens de magnitude mais rápida em comparação à solução de um PVI. O custo do problema de estimação é completamente dominado pelo segundo passo da regularização, onde empregamos o método SS. Mais do que isso, como os parâmetros estimados pelo DS são usados também como valor inicial na solução do SS, veremos que o seu uso chega a diminuir o custo da otimização, por já fornecer um valor inicial próximo ao mínimo da função objetivo.

Com isso, finalizamos a descrição do método Data to Single Shooting (DSS). Ele visa ser uma alternativa viável para o comum cenário em que pesquisadores não tem acesso a bons valores de referência para as variáveis que desejam estimar. O DSS consiste na realização de um processo de regularização em duas etapas. Na primeira etapa, otimizamos um problema de regularização Ridge para o método DS. Na segunda, utilizamos os resultados obtidos pela primeira etapa como valores de referência, solucionando agora um problema de regularização de Tikhonov para o método SS.

No capítulo seguinte, tratamos dos detalhes de implementação desse método, discutindo como foi conduzida a avaliação dos modelos, bem como a estratégia adotada para a seleção do parâmetro de regularização. Também apresentamos os casos de teste selecionados da literatura, bem como os recursos tecnológicos utilizados.

### 36 MÉTODOS PROPOSTOS

## Capítulo 4

# Materiais e Implementação

Esse capítulo visa registrar os principais conceitos que possibilitam a correta interpretação e reprodução deste trabalho. A primeira seção, 4.1, é de grande importância pois explicitamos como são feitas as simulações dos dados, a escolha dos hiperparâmetros da regularização e como são avaliados os modelos. Na seção seguinte apresentamos os sistemas de Equações Diferenciais Ordinárias selecionados como casos de teste. A seção 4.3 fecha o capítulo informando brevemente detalhes acerca das tecnologias empregadas, bem como os recursos computacionais utilizados.

### 4.1 Detalhes de Implementação

O correto entendimento dos resultados tratados no capítulo 5, depende da compreensão de detalhes da implementação. Primeiramente trataremos sobre como os dados experimentais são simulados. Em seguida descrevemos como foi feita a escolha dos hiperparâmetros da regularização. Por fim, focamos nas métricas utilizados para comparação dos diferentes métodos.

### 4.1.1 Simulação de Experimentos

Todos os dados utilizados nesse trabalho são obtidos *in silico*, isto é, através de simulações. Após a definição do fenômeno a ser estudado, seu respectivo modelo matemático (em nosso caso EDOs) e os valores iniciais do sistema, podemos utilizar um método numérico para solucionar o PVI resultante em um determinado intervalo de tempo. Essa solução nos dá acesso ao vetor de estados  $\mathbf{x}(t)$ .

Em todos os casos de teste assumimos que todas as variáveis de estado são observáveis. Isso é necessário por conta da limitação do método DS. Essa não é uma suposição que pode ser feita sempre prática. Contudo, essa é uma suposição adotada para todos os métodos comparados, de modo que todos se beneficiaram desse contexto.

O próximo passo é tornar os dados dessa simulação mais próximos do que encontraríamos em uma coleta *in vitro*. Primeiro, definimos o número de amostras que iremos utilizar, que está relacionada à taxa de amostragem  $\Delta t$ . Se realizamos a simulação no intervalo  $[t_{ini}, t_{fim}]$  podemos particioná-lo em n intervalos iguais, obtendo o vetor  $\mathbf{t} = [t_0, t_1, \ldots, t_n]$  de tempos. Com isso, selecionamos n + 1 amostras de cada um dos estados do vetor  $\mathbf{x}(t)$ , ou seja, o conjunto  $\mathbf{x}(t_i), i = 0, 1, \ldots, n$ .

Para todos os casos de teste fixamos o número de coleta das variáveis de estado em 10, distribuídas igualmente ao longo do intervalo de tempo do problema em questão. Desse modo, para o caso FIC, por exemplo, que conta com duas variáveis de estado e intervalo de integração [0, 1], teremos uma observação de cada estado para  $t_0 = 0$ , outra em  $t_1 = 0.111$ ,  $t_2 = 0.222$  e assim sucessivamente até  $t_9 = 1$ . Ao fim do processo teremos um total de 20 observações.

Em seguida, é preciso acrescentar o ruído inerente à captura de dados experimentais. Para isso, utilizamos uma variação do método descrito por Gábor e Banga [GB15] como encontrado no docu-

mento Additional File 4 - Description of the case studies. O ruído adicionado a cada amostra segue uma distribuição Gaussiana, com média  $\mu$  igual a zero e desvio padrão calculado como

$$\sigma_{ij} = \zeta x_i(t_j) + \varepsilon, \tag{4.1}$$

onde  $\zeta \in [0,1]$  corresponde à porcentagem da influência da variável de estado  $x_i(t_j)$  no cálculo do ruído, e  $\varepsilon$  corresponde a  $\varepsilon = 0.01\bar{x}_i$ . Por sua vez,  $\bar{x}_i$  corresponde ao valor médio da variável de estado  $x_i$  ao longo do intervalo de tempo t observado.

A variável  $\varepsilon$  representa um ruído inerente à captura de dados, em que valores menores que  $\varepsilon$  não podem ser decompostos desse ruído. Os autores proponentes dessa fórmula como simulação de ruído utilizam valores específicos de  $\varepsilon$  para cada caso de teste, sem necessariamente explicitar as razões por trás dessas escolhas.

Nós optamos por calcular  $\varepsilon$  como sendo 1% do valor médio das variáveis de estado. Com isso temos uma fórmula reproduzível. De acordo com nossos testes, o valor de 1%, em conjunto com a porcentagem de  $\zeta$ , acrescenta um ruído suficiente para distorcer os valores originais sem prejudicar o processo de estimação significativamente a ponto de torná-lo inviável. Empregamos essa fórmula adaptada de Gábor e Banga para os casos de teste de Floudas [FPA+13], que não faz uso dessa técnica de simulação de experimentos e adição de ruído.

Através desses cálculos, cada observação no tempo  $t_j$  de cada variável de estado  $x_i$  possui um respectivo valor de  $\sigma$  associado, que então é utilizado para a geração do ruído Gaussiano. Isso resulta em uma razão constante entre sinal e ruído [GB15]. A figura 4.1 exemplifica esse processo de solução do PVI seguido de amostragem e adição de ruído.



**Figura 4.1:** Exemplo do processo de obtenção dos dados para treinamento do modelo. O problema em questão é o FIC (subseção 4.2.1). Primeiro o PVI é solucionado através de um integrador numérico, resultando em 4.1a. A seguir, são extraídas amostras do vetor de estados de acordo com um vetor de tempos **t**. Por fim, adiciona-se o ruído artificial, gerando 4.1b. Aqui o valor de  $\zeta$  é 0.1, de modo que  $\sigma$  seja aproximadamente 10% do valor original da variável de estado.

Entendendo como é feita a simulação dos dados, tratamos a seguir de detalhes acerca da seleção de parâmetros. Mais especificamente, sobre como obter o valor de  $\alpha$  para a regularização Tikhonov.

### 4.1.2 Seleção de Hiperparâmetros

A técnica de regularização de Tikhonov é fundamental para o método DSS. Um dos passos cruciais para o seu uso é a correta seleção do parâmetro  $\alpha$ , que controla a influência do regularizador. Quando  $\alpha = 0$  temos o problema original, ao passo que quando  $\alpha \to \infty$  o problema se torna enviesado e os parâmetros estimados passam a desconsiderar a influência dos dados durante o cálculo da otimização. Uma das formas de se encontrar o valor ideal de  $\alpha$  é através da técnica da Validação Cruzada (VC) ou *Cross-validation* (CV) [RG16]. Mais especificamente, utilizaremos a técnica denominada *K*-fold Cross-validation. Ela consiste em dividir o conjunto de dados em *K* subconjuntos ou *folds* de tamanhos iguais. Esses conjuntos são divididos em grupos de treino e validação, sendo o primeiro utilizado nos cálculos da estimação dos parâmetros e o segundo na verificação da capacidade de generalização do modelo.

O problema de estimação de parâmetros em EDOs, contudo, conta com uma peculiaridade. O vetor de estados  $\mathbf{x}$  é dependente do tempo. Isto é, os dados podem ser pensados como uma série temporal (*time series*). Existe portanto uma relação de ordem entre os dados, de tal modo que não podemos criar folds contendo subconjuntos arbitrários dos dados. Na área de estudos das séries temporais essa dificuldade é tratada através do uso da técnica do rolling forecasting [HA18], onde os folds são criados respeitando a sequência original dos dados.

Nós optamos por uma abordagem mais simples, contornando esse problema. Em nossa implementação, assumimos que  $N_e$  experimentos podem ser realizados para um dado fenômeno. Isto é, para um dado problema de estimação de parâmetros, realizamos  $N_e$  simulações, através da solução do PVI que descreve o fenômeno. Essas  $N_e$  simulações compartilham o mesmo valor inicial, mesma taxa de amostragem, mas são completamente independentes uma das outras. Com isso, elas apresentam diferentes valores para os dados coletados, uma vez que o ruído acrescentado em cada uma é aleatório e também independente.

O Leave One Out Cross-Validation (LOOCV) é um caso específico do K-fold Cross-validation onde K é igual ao número de amostras. Em nossa implementação, ao invés de amostras, aplicamos esse conceito a conjuntos de amostras, ou seja, as simulações. Portanto, realizamos  $N_e$  experimentos e tomamos  $K = N_e$ . Com isso, treinamos o modelo utilizando os dados de  $N_e - 1$  experimentos e avaliamos o modelo em um experimento, reservado para validação. Isso é feito  $N_e$  vezes e então calculamos a média da métrica de erro utilizada.

A figura 4.2 ilustra esse processo com  $K = N_e = 6$ . O dataset nesse exemplo é composto por seis simulações. Assim, são realizadas seis iterações da validação LOOCV, sendo o modelo treinado com os dados de cinco experimentos, e em seguida testado contra o experimento separado para validação. Ao fim do processo temos um valor robusto para realizar a comparação entre os modelos.



**Figura 4.2:** Exemplo do processo de LOOCV empregado. Ao invés de separar amostras em validação e treino, separamos conjuntos de amostras, que compõem um experimento.

Para avaliação dos modelos, utilizamos a métrica de erro Normalized Root Mean Square Error (NRMSE), definida como

$$NRMSE = \sqrt{\frac{1}{N_D} \sum_{k=1}^{N_e} \sum_{j=1}^{N_{y,k}} \frac{\sum_{i=1}^{N_{t,k,j}} (y_{ijk} - \tilde{y}_{ijk})^2}{(max_i \tilde{y}_{ijk} - min_i \tilde{y}_{ijk})}}.$$
(4.2)

Na equação 4.2  $N_D$  é o número total de dados observados,  $N_e$  é o número de experimentos utilizado no treino,  $N_{y,k}$  o número de estados observáveis do k-ésimo experimento,  $N_{t,k,j}$  é o número de amostras do k-ésimo experimento para o *j*-ésimo estado observável. Os valores  $y_{ijk}$  e  $\tilde{y}_{ijk}$  correspondem ao valor dos dados observados e dos valores estimados pelo modelo, respectivamente. Essa é a mesma métrica adotada por Gábor e Banga [GB15] em seu estudo, descrita no anexo Additional File 1 - Numerical methods - Details.

Após executar diversas rodadas de LOOCV, cada uma com um valor distinto para  $\alpha$ , podemos comparar seus NRMSEs e concluir o melhor valor para esse parâmetro. Em nossa implementação consideramos o conjunto de valores (1.0, 0.1, 0.001, 0.0001, 0.0) para  $\alpha$ . Além disso, realizamos a validação cruzada utilizando LOOCV com  $N_e = 5$ . Dessa forma, realizamos 5 simulações independentes para um dado modelo de interesse e elas são utilizados no esquema LOOCV conforme descrito anteriormente.

Após obter o parâmetro  $\alpha^*$  ideal, o modelo é então retreinado com esse valor, mas desta vez utilizando todos os  $N_e$  experimentos para treinamento. O próximo passo é avaliar esse modelo final contra um conjunto de experimentos não visto pelo modelo até o momento, denominado conjunto de teste. Novamente utilizamos o NRMSE nessa etapa como métrica de erro. Entraremos em detalhe a seguir acerca de outras métricas importantes para esse estudo.

### 4.1.3 Avaliação de Modelos

Há diversas formas de se avaliar a qualidade de um modelo. Iremos tratar aqui das métricas escolhidas para avaliar a eficácia e a eficiência dos modelos considerados. Queremos verificar a qualidade dos novos modelos propostos Data Shooting (DS) e Data to Single Shooting (DSS) e utilizaremos como base de comparação o método Single Shooting (SS) e o Single Shooting Regularized (SSR).

Como descrito na seção 3.3, o método DSS visa fornecer uma opção viável para o "pior cenário" descrito por Gábor e Banga [GB15], em que pesquisadores não tem acesso a boas estimativas iniciais dos parâmetros que desejam estimar. Denominamos aqui SSR o método proposto pelos autores nesse cenário, que consiste em realizar uma regularização em dois passos utilizando SS em ambos.

Recordando a estratégia, primeiro o problema é otimizado utilizando regularização Ridge. Em seguida aplica-se a regularização Tikhonov sem pesos (*non-weighted Tikhonov*). O resultado do primeiro passo é tomado como parâmetro de referência para o segundo e também como valor inicial  $\Theta_0$ para o algoritmo de otimização. Em contraste, destacamos que quando nos referimos a SS indicamos que se trata do método clássico, sem nenhum tipo de regularização.

Utilizaremos novamente aqui o NRMSE, definido na subseção anterior, conforme a fórmula 4.2. Após a escolha do hiperparâmetro da regularização, o modelo final é treinado e avaliado contra um conjunto de teste. Tomamos alguns cuidados para que possamos verificar de forma mais realista a capacidade de generalização do modelo.

Primeiramente, garantimos que os experimentos que compõem o conjunto de teste não fazem parte do conjunto de treinamento ou validação. Portanto, são dados nunca antes vistos pelo modelo. Aplicamos nas simulações do conjunto de teste os mesmos passos descritos na subseção 4.1.1, tanto com relação à amostragem quanto ao ruído sintético. Contudo, adicionamos um novo fator exclusivo aos experimentos desse conjunto. Esse fator é a simulação utilizando diferentes valores iniciais. A simulação dos experimentos se inicia com a solução de um PVI. No conjunto de treino, todos os experimentos correspondem ao mesmo PVI, sendo diferenciados apenas pelo ruído artificial adicionado posteriormente. Já para os experimentos de teste, utilizamos diferentes valores iniciais, o que nos leva a diferentes PVIs.

Esses valores iniciais são obtidos através do mesmo método de acréscimo de ruído artificial, mas agora o ruído é acrescentado aos valores iniciais, antes da simulação. Nesse caso utilizamos  $\zeta = 0.01$ , fazendo com que permaneçam em um intervalo controlado e realista. Como consequência, a dinâmica do sistema é levemente alterada. Essa técnica também é empregada por Gábor e Banga [GB15] em suas análises, com pequenas modificações. O conjunto de teste para todos os problemas avaliados é composto por 10 simulações.

O erro do conjunto de teste será o NRMSE médio das 10 simulações do conjunto. Esse será o resultado de uma rodada de estimação de parâmetros. Contudo, como discutiremos, por conta da natureza estocástica dos procedimentos adotados, precisaremos realizar diversas rodadas de estimação. Visando sumarizar esses valores, apresentamos agora a definição de Taxa de Sucesso (TS) ou Success Rate (SR), conformo proposto por Villaverde e colaboradores [VFW<sup>+</sup>19].

A SR, para esses autores, corresponde à frequência relativa de Sucessos. Um algoritmo atinge a marca de Sucesso, quando o valor de sua função objetivo consegue atingir um valor menor que um limiar (*threshold*), denominado Value To Reach (VTR), dado um limite de tempo computacional, Maximum Time Allowed (MAXT).

A imagem 4.3 ilustra essa definição. Isso faz sentido no caso dos autores, dado que estão comparando diferentes métodos numéricos de otimização, todos utilizando a mesma função objetivo. Em nosso caso, contudo, precisamos redefinir essa métrica, dado que temos diferentes funções objetivo. Esse fato nos impossibilita de realizar a comparação direta entre essas curvas.



Figura 4.3: Ilustração do critério de sucesso, adaptado de Villaverde et al.  $[VFW^+19]$ . O gráfico à esquerda mostra a evolução da função objetivo de três métodos ao longo do tempo. A linha cheia representa o a mediana, enquanto que as áreas coloridas representam o alcance dos valores obtidos nas diversas execuções do método. A linha tracejada vertical MAXT (Maximum Time) marca o tempo máximo dado para a otimização, enquanto que a linha tracejada horizontal VTR (Value To Reach) delimita o valor mínimo que deve ser alcançado para que seja contabilizada uma execução com sucesso. À direita vemos o gráfico mostrando a densidade dos valores obtidos para a função objetivo. Para cada método podemos calcular a sua Taxa de Sucesso ou Success Rate (SR) através da área sob a curva que estiver abaixo da linha VTR.



**Figura 4.4:** Ilustração do critério de sucesso adotado. O gráfico mostra a densidade da métrica NRMSE média obtida no conjunto de teste. Para cada método podemos calcular a sua respectiva Taxa de Sucesso ou Success Rate (SR) através da área sob a curva que estiver à esquerda da linha VTR.

Portanto, mantemos a definição de SR como sendo a frequência relativa de sucessos. Porém, mudamos a definição de Sucesso. Em nosso caso, um método é bem sucedido quando seu NRMSE no conjunto de teste é menor que um determinado limiar VTR, dessa vez relacionado a um valor de erro máximo desejado. Também impomos um limite de tempo computacional (MAXT) para todos os métodos. Essa definição é exemplificada na figura 4.4.

Constatamos empiricamente, através do comportamento qualitativo das variáveis de estado, que um valor razoável para VTR é 0.125. Isso porque em nossas simulações fica evidente que duas curvas de estados que possuem um NRMSE menor que esse valor são notavelmente próximas uma da outra. Já quando observamos duas curvas que geram um NRMSE maior que 0.125 entre elas, essas começam a apresentar uma divergência de comportamento mais acentuado. Nesse caso os parâmetros estimados demonstram não conseguir gerar simulações condizentes com os dados experimentais e por isso não podem ser classificados como uma estimação de sucesso. Salientamos que como se trata de uma métrica normalizada, esse *threshold* pode ser utilizado independentemente da magnitude dos valores do vetor de estados. Já o MAXT utilizado durante o treino de todos os modelos é igual a  $10^4$  segundos.

A métrica de Success Rate depende da execução de diversas rodadas do processo de treinamento, validação e teste. Esse cuidado é especialmente necessário em nosso caso pois estamos em um contexto estocástico. Isso por que acrescentamos ruído artificial aos dados, simulando dados reais. Os resultados obtidos podem então estar relacionados a uma particular configuração dos experimentos e favorecer ou penalizar arbitrariamente algum dos métodos. Por esse motivo, o processo todo é realizado um total de M = 30 vezes para cada problema, para cada método.

Além do processo estocástico envolvido na adição do ruído artificial, também existe a aleatoriedade relacionada ao valor inicial  $\Theta_0$  do processo de otimização. De modo a tornar essa etapa mais controlado e justa, para cada uma das 30 realizações um novo  $\Theta_0$  é gerado. Se trata de um valor aleatório dentro dos limites (*bounds*) conhecidos para os valores dos parâmetros a serem estimados. Ressaltamos que, para cada rodada, o mesmo valor inicial é utilizado como ponto de partida no algoritmo de otimização em todos os métodos comparados.

Um gráfico que possibilita a comparação de modelos utilizando essa métrica de SR é o apresentado



**Figura 4.5:** Gráfico para análise de Success Rate e tempo computacional, adaptado de Villaverde et al. [VFW<sup>+</sup>19]. Os pontos que se encontram na frente de Pareto são não dominados, isto é, são tecnicamente equivalentes com relação ao tradeoff entre Taxa de Sucesso e tempo. Os ponto acima e à direita dessa curva são dominados, i.e. piores que os pontos que se encontram na curva, seja em relação ao custo computacional ou em relação à Success Rate (SR). As áreas preenchidas correspondem ao tempo computacional médio  $\langle t \rangle_{succ}$  necessário para uma rodada de sucesso.

na figura 4.5. Nele vemos a relação e *tradeoff* entre custo computacional e a Success Rate. A disposição dos eixos é de tal forma que valores menores (em ambos os eixos) são melhores.

Tendo definido SR em nosso contexto, podemos utilizar o conceito de Overall Efficiency (OE), dos mesmos autores [VFW<sup>+</sup>19]. A OE de um método i, dado algum problema, é definido como

$$OE_i = \frac{\min_j\{\langle t \rangle_{succ_j}\}}{\langle t \rangle_{succ_i}},\tag{4.3}$$

onde  $\langle t \rangle_{succ_i}$  é o tempo computacional médio que o método *i* precisa para obter um Sucesso. Esse valor é calculado como  $\langle t \rangle_{succ_i} = \langle t \rangle_i / SR_i$ , onde  $\langle t \rangle_i$  e  $SR_i$  são o tempo computacional médio e a Taxa de Sucesso para *i* naquele problema, respectivamente. Já o numerador corresponde ao tempo mínimo registrado para se obter um Sucesso nesse problema, levando em conta  $\langle t \rangle_i$  de todos os métodos. Desse modo, OE está definida no intervalo fechado [0, 1] e o modelo com melhor eficiência atinge a pontuação máxima de OE = 1.

Utilizaremos o OE para fornecer uma comparação mais direta entre os métodos SS, SSR e DSS. No geral, é possível facilmente verificar a performance de um método observando o gráfico de SR versus custo, como em 4.5. OE é útil para confirmar numericamente casos ambíguos, com os métodos se posicionando perto de uma frente de Pareto por exemplo, o que dificulta a definição do melhor método graficamente.

Data Shooting é excluído da métrica OE. Primeiro, temos uma questão prática: o método se demonstra ordens de magnitude mais rápido que os outros, distorcendo os gráficos e dificultando a comparação entre os demais métodos. Em segundo lugar, e mais importante, esse método gera uma função objetivo consideravelmente diferente das demais.

Desejamos evitar o cenário no qual DS obtém uma Taxa de Sucesso muito baixa mas, ao mesmo

tempo, atinge um custo computacional excepcionalmente menor que os outros métodos. Desse modo, o custo computacional acabaria dominando o cálculo da métrica e, com isso, "compensando" sua baixa SR. Além disso, como veremos nos resultados, não haverá ambiguidades nos resultados de DS, com os gráficos sendo suficientes para análise de seu comportamento.

Destacamos também que, visando uma comparação justa, todos utilizarão o mesmo algoritmo de otimização. O algoritmo escolhido foi o otimizador global de Recozimento Simulado, ou Simulated Annealing [CSB<sup>+</sup>16, AFNKB09, BT<sup>+</sup>93]. Mais especificamente optamos por um algoritmo que consegue aplicar limites (*bounds*) nos parâmetros. Todos os hiperparâmetros desse algoritmo foram mantidos fixos para todos os modelos testados. Apesar do método DS conseguir potencialmente gerar funções convexas, os demais métodos não conseguem. Por esse motivo escolhemos um otimizador global capaz de lidar com um grande número de mínimos locais.

Banga, Moles e Alonso [BMA04] mencionam que a eficiência e robustez do método de Simulated Annealing nem sempre fazem jus à sua popularidade, sendo em alguns casos superado por alternativas mais simples. Apesar disso, os autores também explicitam que a literatura é bastante fragmentada no que diz respeito a avaliação dos melhores métodos de otimização para estimação de parâmetros em EDOs. Ainda assim, eles ressaltam que no geral não existe um algoritmo capaz de se sair melhor em absolutamente todos os casos, fazendo uma alusão ao teorema "No Free Lunch", que postula exatamente isso.

Tendo isso em mente, algumas alternativas foram experimentadas. Dentre elas, verificamos os resultados obtidos através do método híbrido Scatter Search utilizando diferentes otimizadores locais, tanto aqueles baseados em gradiente (e.g. método de Newton) como os livres de gradiente (Nelder-Mead) [AFNKB09]. Também utilizamos alguns algoritmos globais, como Particle Swarm Optimization. Ao fim dos testes, nenhuma das alternativas se destacou consideravelmente, em concordância com o teorema "No Free Lunch".

Então, por conta dessas considerações dos autores e dado o fato de que o objetivo desse trabalho é a comparação dos diferentes métodos de estimação propostos, e não a avaliação e determinação do melhor algoritmo de otimização para o nosso contexto, seguimos com o algoritmo de Simulated Annealing. Esse algoritmo é extremamente popular [BMA04], além de ser um método bastante geral, podendo ser aplicado a diferentes contextos de otimização [BT<sup>+</sup>93].

Quanto às métricas, ressaltamos que os valores de custo computacional considerados para os métodos de duas etapas dizem respeito à segunda etapa apenas. Não há ambiguidades quando consideramos o custo dos métodos Data Shooting e Single Shooting. O custo está relacionado ao tempo necessário para a convergência do algoritmo de otimização utilizando suas respectivas funções objetivo, respeitado o MAXT definido.

Com relação aos métodos Single Shooting Regularized e Data to Single Shooting, por outro lado, temos a execução de dois processos de otimização. Poderíamos estipular seu custo computacional como sendo a soma do tempo necessário para solução de ambas as etapas, mas optamos por medir apenas a segunda, de modo a facilitar a interpretação. Podemos ter uma estimativa do tempo total necessário para esses métodos simplesmente somando ao tempo obtido o custo médio de DS (no caso de DSS) ou SS (no caso de SSR).

Com isso, cobrimos os principais detalhes acerca do processo de avaliação e comparação dos métodos SS, SSR, DS e DSS. A seguir apresentamos diferentes Problemas Inversos utilizados como casos de teste para esses métodos.

### 4.2 Casos de Teste

As reações químicas selecionadas como caso de teste neste trabalho são retirados de trabalhos correlatos, que visam justamente verificar a qualidade de diferentes algoritmos ou métodos numéricos na área de Problemas Inversos. Eles estão dispostos em ordem de crescente complexidade.

### 4.2.1 First Order Irreversible Chain Reaction (FIC)

O primeiro problema corresponde a uma reação irreversível consecutiva genérica.

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \tag{4.4}$$

Do qual podemos extrair o seguinte sistema de Equações Diferenciais Ordinárias.

$$\frac{dx_1}{dt} = -k_1 x_1$$

$$\frac{dx_2}{dt} = k_1 x_1 - k_2 x_2$$

$$\mathbf{x}(0) = \begin{bmatrix} 1, & 0 \end{bmatrix}^T$$

$$t \in \begin{bmatrix} 0, & 1 \end{bmatrix}$$
(4.5)

A tabela 4.1 mostra o valor nominal dos parâmetros que desejamos estimar, bem como seus respectivos limites inferior (LB) e superior (UB). Já a imagem 4.6 demonstra a dinâmica do sistema.

Id	Nominal	LB	UB
$k_1$	5.0035	0	10
$k_2$	1.0	0	10

Tabela 4.1: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema FIC.



Figura 4.6: Solução do PVI para o problema FIC.

Os próximos problemas apresentados irão seguir essa mesma estrutura. Primeiro, quando conveniente, apresentamos a reação química do problema, em seguida a sua formulação em sistema de EDOs, seguido de uma tabela com seus valores nominais e limites, e por fim um gráfico exibindo sua dinâmica.

### 4.2.2 First Order Reversible Chain Reaction (FRC)

O segundo problema se trata de uma equação química reversível consecutiva genérica.

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \tag{4.6}$$

$$\frac{dx_1}{dt} = -k_1 x_1 + k_2 x_2$$

$$\frac{dx_2}{dt} = k_1 x_1 - (k_2 + k_3) x_2 + k_4 x_3$$

$$\frac{dx_3}{dt} = -k_4 x_3 + k_3 x_2$$
(4.7)

$$\mathbf{x}(0) = [1, 0, 0]^T$$
  
 $t \in [0, 1]$ 

Id	Nominal	LB	UB
$k_1$	4.0	0	10
$k_2$	2.0	0	10
$k_3$	40.013	10	50
$k_4$	20.007	10	50

Tabela 4.2: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema FRC.



Figura 4.7: Solução do PVI para o problema FRC.

#### Catalytic Cracking of Gas Oil (CCGO) 4.2.3

Esse problema representa a quebra catalítica do óleo diesel (A) em gasolina (B) e outro produto secundário (S). Apenas A e B foram medidos e por esse motivo a concentração de S não aparece no modelo para estimação.

$$A \xrightarrow{k_1} B$$

$$B \xrightarrow{k_2} S$$

$$A \xrightarrow{k_3} S$$

$$\frac{dx_1}{dt} = -(k_1 + k_3)x_1^2$$

$$\frac{dx_2}{dt} = k_1x_1^2 - k_2x_2$$

$$\mathbf{x}(0) = \begin{bmatrix} 1, 0 \end{bmatrix}^T$$

$$t \in \begin{bmatrix} 0, 0.95 \end{bmatrix}$$

$$\mathbf{X}(0) = \begin{bmatrix} 1, 0 \end{bmatrix}^T$$

$$t \in \begin{bmatrix} 0, 0.95 \end{bmatrix}$$

$$\mathbf{X}(0) = \begin{bmatrix} 1, 0 \end{bmatrix}^T$$

$$\mathbf{X}(0) = \begin{bmatrix} 1,$$

 Tabela 4.3: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema CCGO.

Ì



Figura 4.8: Solução do PVI para o problema CCGO.

### 4.2.4 Bellman's Problem (BEP)

Conhecido como o Problema de Bellman, esse sistema descreve a reação reversível homogênea da fase gasosa do dióxido de nitrogênio.

$$2 \operatorname{NO} + \operatorname{O}_{2} \underbrace{\stackrel{k_{1}}{\longleftarrow}}_{k_{2}} 2 \operatorname{NO}_{2}$$

$$(4.10)$$

$$\frac{dx}{dt} = k_{1} (126.2 - x) (9.19 - x)^{2} - k_{2} x^{2}$$

$$x(0) = 0$$

$$t \in [0, 39]$$

$$\overline{\frac{\operatorname{Id} \operatorname{Nominal} \operatorname{LB} \operatorname{UB}}_{k_{1}} 4.5704 \cdot 10^{-6} \ 0 \ 10^{-2}}_{k_{2}} 2.7845 \cdot 10^{-4} \ 0 \ 10^{-2}}$$

Tabela 4.4: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema BEP.



Figura 4.9: Solução do PVI para o problema BEP.

### 4.2.5 Methanol-to-Hydrocarbons Process (MHP)

O problema 5 representa a conversão de metanol em diversos hidrocarbonetos.

$$A \xrightarrow{k_1} B \qquad A + B \xrightarrow{k_2} C$$

$$A \xrightarrow{k_4} C \qquad C + B \xrightarrow{k_3} P \qquad (4.12)$$

$$A \xrightarrow{k_5} P \qquad A + B \xrightarrow{k_6} P$$

$$\frac{dx_1}{dt} = x_1 \left( -2k_1 + \frac{k_1 x_2}{(k_2 + k_5) x_1 + x_2} - k_3 - k_4 \right) 
\frac{dx_2}{dt} = \frac{k_1 x_1 (k_2 x_1 - x_2)}{(k_2 + k_5) x_1 + x_2} + k_3 x_1 
\frac{dx_3}{dt} = \frac{k_1 x_1 (x_2 + k_5 x_1)}{(k_2 + k_5) x_1 + x_2} + k_4 x_1$$
(4.13)

$$\mathbf{x}(0) = [1, 0, 0]^T$$
$$t \in [0, 1.122]$$

Id	Nominal	LB	UB
$k_1$	5.2407	0	$10^{2}$
$k_2$	1.2176	0	$10^{2}$
$k_3$	0	0	$10^{2}$
$k_4$	0	0	$10^{2}$
$k_5$	0	0	$10^{2}$

 Tabela 4.5: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema MHP.



Figura 4.10: Solução do PVI para o problema MHP.

### 4.2.6 Lotka-Volterra's Problem (LVP)

Este problema destoa dos anteriores por não ser proveniente da área da cinética química. Contudo, se trata de um problema interessante, recorrente na literatura e, portanto, que convém verificar os resultados obtidos pelo estimador proposto. Se trata do Problema de Lotka-Volterra. Ele envolve um modelo de dois parâmetros de presa-predador similar ao usado na área da ecologia.

$$\frac{dx_1}{dt} = k_1 x_1 (1 - x_2)$$
  

$$\frac{dx_2}{dt} = k_2 x_2 (x_1 - 1)$$
  

$$\mathbf{x}(0) = [1.2, 1.1]^T$$
  

$$t \in [0, 10]$$
  
(4.14)

Id	Nominal	LB	UB
$k_1$	3.2434	0	10
$k_2$	0.9209	0	10

Tabela 4.6: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema LVP.



Figura 4.11: Solução do PVI para o problema LVP.

Todos os seis problemas acima foram extraídos do livro Handbook of Test Problems in Local and Global Optimization de Floudas *et al.* [FPA<sup>+</sup>13], no qual é possível encontrar diversos problemas de otimização organizados por áreas. Em nosso caso, recorremos aos problemas de estimação de parâmetros no capítulo referente à otimização dinâmica (Capítulo 15.3).

### 4.2.7 Biomass Batch Growth (BBG)

\_

Conhecido como Biomass Batch Growth (BBG), a seguinte reação descreve a transformação de substrato em biomassa. Esse problema é utilizado por Gábor e Banga [GB15] em seus testes.

$$\frac{dC_b}{dt} = \mu_{max} \frac{C_s C_b}{K_s + C_s} - k_d C_b$$

$$\frac{dC_s}{dt} = -\frac{\mu_{max}}{yield} \frac{C_s C_b}{K_s + C_s}$$
(4.15)
$$C_b(0) = 2$$

$$C_b(0) = 30$$

$$t \in [0, 12]$$

$$\frac{Id}{\mu_{max}} \frac{Nominal}{0.4} \frac{LB}{10^{-5}} \frac{UB}{10^2}$$

$$K_s \quad 5 \qquad 10^{-5} \quad 10^2$$

$$K_d \quad 0.05 \qquad 10^{-5} \quad 10^2$$

$$yield \quad 0.5 \qquad 10^{-5} \quad 10^2$$

Tabela 4.7: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema BBG.



Figura 4.12: Solução do PVI para o problema BBG.

#### 4.2.8FitzHugh-Nagumo (FHN)

Este sistema de EDOs corresponde ao problema de FitzHugh-Nagumo (FHN). Ele descreve a relação entre a voltagem V e a corrente R sobre uma membrana de éxon [GB15].

$$\frac{dV}{dt} = \gamma (V - V^3/3 + R) 
\frac{dR}{dt} = -1/\gamma (V - \alpha + \beta R) 
V(0) = -1 
R(0) = 1 
t \in [1, 20]$$
(4.16)
  

$$\frac{Id \text{ Nominal } LB \text{ UB}}{\alpha \quad 0.2 \quad 10^{-5} \quad 10^{5}} 
\beta \quad 0.2 \quad 10^{-5} \quad 10^{5} 
\gamma \quad 0.3 \quad 10^{-5} \quad 10^{5}$$

 $10^5$ 

Tabela 4.8: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema FHN.

0.2

0.3

β

 $\gamma$ 



Figura 4.13: Solução do PVI para o problema FHN.

### 4.2.9 Mitogen Activated Protein Kinases (MAPK)

O nono problema corresponde ao modelo matemático da via de sinalização MAPK de Kholodenko [GB15]. O modelo também está disponível na base Biomodels [LDR<sup>+</sup>10] (BIOMD0000000010 - Kholodenko2000). Nesse problema nem todos os parâmetros que compõem a tabela 4.9 foram estimados, apenas os parâmetros com valores explicitados nas colunas LB e UB. Os demais tiveram seus valores nominais diretamente substituídos nas equações.

$$\begin{aligned} R_{J0} &= J0_{V1} \frac{x_1}{(1 + (x_8/J0_{Ki})^{J0_n})(J0_{K1} + x_1)} \\ R_{J1} &= J1_{V2} \frac{x_2}{J1_{KK2} + x_2} \\ R_{J2} &= J2_{k3} \frac{x_2 x_3}{J2_{KK3} + x_3} \\ R_{J3} &= J3_{k4} \frac{x_2 x_4}{J3_{KK4} + x_4} \\ R_{J4} &= J4_{V5} \frac{x_5}{J4_{KK5} + x_5} \\ R_{J5} &= J5_{V6} \frac{x_4}{J5_{KK6} + x_4} \\ R_{J6} &= J6_{k7} \frac{x_5 x_6}{J6_{KK7} + x_6} \\ R_{J7} &= J7_{k8} \frac{x_5 x_7}{J7_{KK8} + x_7} \\ R_{J8} &= J8_{V9} \frac{x_8}{J8_{KK9} + x_8} \\ R_{J9} &= J9_{V10} \frac{x_7}{J9_{KK10} + x_7} \end{aligned}$$

$$\frac{dx_1}{dt} = -R_{J0} + R_{J1}$$

$$\frac{dx_2}{dt} = R_{J0} - R_{J1}$$

$$\frac{dx_3}{dt} = -R_{J2} + R_{J5}$$

$$\frac{dx_4}{dt} = R_{J2} - R_{J3} + R_{J4} - R_{J5}$$

$$\frac{dx_5}{dt} = R_{J3} - R_{J4}$$

$$\frac{dx_6}{dt} = -R_{J6} + R_{J9}$$

$$\frac{dx_7}{dt} = R_{J6} - R_{J7} + R_{J8} - R_{J9}$$

$$\frac{dx_8}{dt} = R_{J7} - R_{J8}$$

$$\mathbf{x}(0) = [90, 10, 280, 10, 10, 280, 10, 10]^T$$
  
$$t \in [1, 1000]$$

Id	Nominal	LB	UB
$J0_{V1}$	2.5	$10^{-2}$	50
$J0_{Ki}$	9		
$J0_n$	1		
$J0_{K1}$	10		
$J1_{V2}$	0.25	$10^{-2}$	50
$J1_{KK2}$	8		
$J2_{k3}$	0.025		
$J2_{KK3}$	15		
$J3_{k4}$	0.75		
$J3_{KK4}$	15		
$J4_{V5}$	0.75	$10^{-2}$	50
$J4_{KK5}$	15		
$J5_{V6}$	0.75	$10^{-2}$	50
$J5_{KK6}$	15		
$J6_{k7}$	0.025		
$J6_{KK7}$	15		
$J7_{k8}$	0.025		
$J7_{KK8}$	15		
$J8_{V9}$	0.5	$10^{-2}$	50
$J8_{KK9}$	15		
$J9_{V10}$	0.5	$10^{-2}$	50
$J9_{KK10}$	15		

Tabela 4.9: Parâmetros e respectivos valores nominais e limites inferior e superior para o problema MAPK.



Figura 4.14: Solução do PVI para o problema MAPK.

Os resultados obtidos para os casos de teste apresentados serão tratados no seguinte capítulo, 5, e serão referenciados por suas respectivas siglas. Antes, finalizamos este capítulo apresentando as tecnologias que possibilitaram a implementação do trabalho.

### 4.3 Tecnologias

A implementação de todos os conceitos apresentados até então é um dos principais objetivos deste trabalho. Nessa seção tratamos brevemente das tecnologias envolvidas nesse processo. Primeiro apresentamos detalhes relacionados ao software e em seguida detalhes do hardware utilizado.

### 4.3.1 Software e Linguagem

A linguagem de programação escolhida para desenvolvimento neste trabalho foi a linguagem Julia, versão 1.3 (disponível em https://julialang.org/) [BEKS17]. Trata-se de uma linguagem de alto nível mas também de alta performance. Apesar de ser uma linguagem de uso geral, ela é amplamente utilizada na área da computação científica. Isso pode ser confirmado pelo vasto número de bibliotecas específicas para essa área desenvolvidas pela comunidade. Dessas, algumas foram indispensáveis para o projeto.

DifferentailEquation.jl foi a biblioteca que proporcionou os blocos essenciais para se trabalhar com Equações Diferenciais Ordinárias. Ela é desenvolvida primariamente por Rackauckas [RN17] e faz parte da organização SciML (página da organização: https://sciml.ai/). Esse pacote abarca não apenas estruturas especializadas para EDOs, mas também Equações Diferenciais Ordinárias Estocásticas, Equações Diferenciais Algébricas, Equações Diferenciais Parciais, entre outras. Além disso, ela conta com um vasto conjunto de métodos numéricos para solução de PVIs [Rac17], bem como ferramentas para análise das equações, proporcionando um ecossistema rico e integrado para esse conjunto de problemas.

Todos os Problemas de Valor Inicial foram solucionados utilizando um integrador agnóstico à natureza do problema. Isto é, se trata de um método capaz de detectar e automaticamente utilizar a correta abordagem dependendo da estrutura do problema, verificando se o problema é rígido (*stiff*) ou não. Mais especificamente, trata-se do método AutoVern7 (Rodas5()), encontrado no pacote DifferentialEquations.jl. É um método equivalente ao bastante difundido LSODA [Pet83]. Optamos por esse método por tornar o estimador proposto o mais genérico possível.

Outro pacote fundamental para este trabalho é o Optim.jl [MR18, MWR<sup>+</sup>20]. Essa biblioteca contém diferentes algoritmos de otimização, dividos entre os que não necessitam de gradiente (gradient free methods), aqueles requerem o gradiente da função objetivo (e.g. Gradiente Descendente), ou ainda métodos que dependem de informação da hessiana do problema (e.g. método de Newton). Em nosso estudo utilizamos um método gradient free global, mais especificamente o Simulated Annealing with bounds, denominado SAMIN nessa biblioteca.

Para produção dos gráficos neste trabalho foram utilizados o software Inkscape (disponível em https://inkscape.org/) e os pacotes Gadfly [JAN<sup>+</sup>20] e Plots.jl [Bre20]. Em sua maioria, os gráficos contendo exemplos conceituais foram elaborados utilizando o programa Inkscape, um editor de imagens vetoriais. Já as imagens referentes a simulações e métricas foram todos obtidos através dos pacotes Gadfly [JAN<sup>+</sup>20] ou Plots.lj [Bre20], sendo o primeiro utilizado na geração dos gráficos em duas dimensões e o segundo para gráficos em três dimensões. Os gráficos de radar foram gerados através da biblioteca ECharts (disponível em https://echarts.apache.org/).

### 4.3.2 Hardware e Sistema Operacional

O processo de estimação de parâmetros envolvendo EDOs já é naturalmente mais custoso que aqueles envolvendo equações tradicionais, por conta da solução dos PVIs durante a otimização. Além disso, ainda há a execução de diversas rodadas de otimização, para lidar com a natureza estocástica do problema.

Assim, para a obtenção dos resultados deste trabalho, foi necessário o uso de um servidor gerenciado pela rede de pesquisa Vision Group IME (página do grupo: http://www.vision.ime.usp.br/). Tratase de uma máquina com 252GB de memória RAM e um processador Intel Xeon CPU E5-2690.

Esse processador conta com 32 núcleos de 2.90GHz cada. O sistema operacional utilizado é uma distribuição GNU/Linux Ubuntu, versão 18.04.4, com arquitetura 64 bits.

Com isso, finalizamos o presente capítulo, que tratou da metodologia, materiais e experimentos utilizados no desenvolvimento deste trabalho. Entramos em detalhes de implementação, além de elencar os casos de teste e tecnologias utilizadas. Prosseguimos agora com a apresentação dos resultados e sua discussão no capítulo a seguir.

## Capítulo 5

# Resultados e Discussão

Nesse capítulo apresentamos os principais resultados deste trabalho. A seção 5.1 trata de comparar os novos métodos discutidos no capítulo anterior, Data Shooting e Data to Single Shooting, com o método clássico de Single Shooting e o Single Shooting Regularized de Gábor e Banga [GB15]. Finalmente, indicamos na seção 5.2 informações sobre a implementação e onde o código fonte para reprodução dos resultados pode ser obtido.

### 5.1 Comparação dos Métodos

Começaremos contrastando as superfícies das funções objetivo dos métodos de Single Shooting e Data Shooting, na subseção 5.1.1. Esperamos com isso destacar visualmente a principal diferença entre esses dois métodos. Em seguida, na subseção 5.1.2, fazemos a avaliação dos diferentes modelos de Shooting utilizando as estratégias e métricas discutidas na seção 4.1. Com isso, mostramos como os métodos propostos se comportam na prática e se comparam com a alternativa clássica.

### 5.1.1 Convexidade

Apresentamos na seção 3.2 o método Data Shooting, um dos métodos propostos neste trabalho. Mostramos que sua simplicidade conceitual se reflete em sua fórmula matemática. Exploramos aqui como essa estrutura simples se manifesta com relação à superfície de otimização gerada no cálculo da função objetivo.

Sabemos que o método dos Mínimos Quadrados Linear (MQL) é convexo por sua formulação. O que difere um problema do tipo MQL de um do tipo Mínimos Quadrados Não Linear (MQNL) é a função estimadora utilizada no cálculo do resíduo. Quando essa função é linear nos parâmetros  $\theta$ , temos um MQL, do contrário, caímos no caso de um MQNL.

No caso do método Data Shooting, a função objetivo gerada em seus cálculos depende da derivada do vetor de estados, i.e.  $\dot{\mathbf{x}}$ . Quando todas essas funções que compõem o sistema de Equações Diferenciais Ordinárias (EDOs) são lineares, sabemos que teremos um problema convexo em mãos, como discutido na seção 3.2. Do contrário, teremos um problema de otimização não convexo.

Começamos exemplificando essa propriedade através do simples Problema de Valor Inicial (PVI)

$$\dot{x} = \theta x$$

$$x(0) = 1.$$
(5.1)

Esse é um PVI com uma das mais simples EDOs conhecidas, possuindo inclusive uma solução analítica. Quando  $\theta = 1$  temos a EDO vista em 3.1, que dá origem à função exponencial

$$x(t) = e^t. (5.2)$$

Nesse exemplo, a função objetivo do método DS seria calculada como

$$Q_{LS}^{DS} = R(\theta)^{T} R(\theta)$$
  
=  $\sum_{i=1}^{n} (y_{i} - \phi(\theta, y_{i-1}))^{2}$   
=  $\sum_{i=1}^{n} (y_{i} - \frac{\Delta t}{2} (f(y_{i-1}, t_{i-1}, \theta) + f(y_{i}, t_{i}, \theta)))^{2}$   
=  $\sum_{i=1}^{n} (y_{i} - \frac{\Delta t}{2} (\theta y_{i-1} + \theta y_{i}))^{2}$  (5.3)

Com esse exemplo fica claro que temos uma função objetivo convexa. Vamos utilizar essa função para verificar o perfil de sua curva de otimização. Primeiro solucionamos o PVI com  $\theta = 1$  e coletamos um conjunto de dados para que possamos realizar as contas com  $y_i$ . Então, deixando livre  $\theta$  na função  $Q_{LS}^{DS}$ , podemos variar esse parâmetro e verificar o comportamento da curva para o método Data Shooting e contrastar com a função  $Q_{LS}^{SS}$  gerada pelo Single Shooting.



(a) Curva da função objetivo para o método DS. (b) Curva da função objetivo para o método SS.

**Figura 5.1:** Comparação entre curva de otimização dos métodos Data Shooting e Single Shooting para o PVI 5.1. Resultado para 10 dados experimentais. A reta vertical magenta indica o valor nominal do parâmetro  $\theta$ , nesse caso igual a 1.

Temos na figura 5.1 o comportamento esperado. É visível que a função gerada pelo método DS é (estritamente) convexa, conforme aponta sua fórmula. O método SS também gerou uma função aparentemente convexa nesse exemplo. Veremos adiante que essa é uma exceção.

Para isso, mostraremos agora diferentes superfícies de otimização geradas a partir de alguns dos casos de teste apresentados na seção 4.2. Começando pelo problema First Order Irreversible Chain (FIC) tratado na subseção 4.2.1.


(a) Gráfico tridimensional para DS, amplo.



(c) Gráfico de contorno para DS, amplo. Single Shooting



(e) Gráfico tridimensional para SS, amplo.



(g) Gráfico de contorno para SS, amplo.

**Figura 5.2:** Comparação entre curva de otimização dos métodos Data Shooting e Single Shooting para o problema FIC. Gráficos com indicação de "amplo" mostram o perfil da função objetivo percorrendo um grande intervalo de valores para os parâmetros. Já os classificados como "reduzido", mostram o comportamento da função em um intervalo menor, centralizado no valor mínimo da função. Cálculos realizados com coleta de 10 amostras igualmente espaçadas para cada variável de estado.



(b) Gráfico tridimensional para DS, reduzido.



(d) Gráfico de contorno para DS, reduzido. Single Shooting



(f) Gráfico tridimensional para SS, reduzido.



<sup>(</sup>h) Gráfico de contorno para SS, reduzido.

Os gráficos da figura 5.2 mostram o comportamento da função objetivo através de visualizações em três dimensões e também em gráficos de contorno. Para os gráficos tridimensionais, os eixos x e y contém a variação dos parâmetros  $\theta_1 e \theta_2$ , respectivamente. O eixo z corresponde aos valores da função objetivo para cada uma das combinações desses parâmetros.

Já nos gráficos de contorno temos a mesma lógica, mas com eixo z sendo substituído apenas pela variação de cor relacionada aos valores da função. Adicionamos a esses gráficos duas retas indicando os valores nominais dos parâmetros  $\theta_i$ , i = 1, 2. Com isso podemos ver o quão longe dos parâmetros nominais se encontra o valor mínimo da função observando a versão "reduzida" desse gráfico.

Tanto o gráfico tridimensional quanto o de contorno são exibidos em duas versões. Uma versão "ampla" e outra "reduzida". No primeiro caso, percorremos um intervalo maior de valores para os parâmetros  $\theta_i$ . Utilizamos o mesmo intervalo tanto para a função gerada por DS quanto pela gerada por SS. No exemplo da figura 5.2,  $\theta_i \in [0, 10], i = 1, 2$  para ambos os métodos.

No segundo tipo de gráfico, "reduzido", o intervalo é menor e podem ser diferentes para cada método. Isso porque os intervalos aqui são calculados de modo a manter o valor mínimo atingido pela função no centro do gráfico. Esses gráficos podem ser pensados como um *zoom-in* na região onde se encontra o mínimo da função. Na figura 5.2,  $\theta_1 \in [4.4, 5.3], \theta_2 \in [0.5, 1.5]$  para DS, enquanto  $\theta_1 \in [4.5, 5.5], \theta_2 \in [0.5, 1.5]$  para SS.

Confirmamos que o problema FIC possui a estrutura particular que permite a geração de funções convexas pelo método DS. Isso se reflete em seus gráficos. Recordando, isso acontece pois todas as derivadas de seu vetor de estados são funções lineares nos parâmetros.

Quando analisamos os gráficos gerados para o problema LVP, apresentado na subseção 4.2.6, fica ainda mais evidente a discrepância entre a função objetivo gerada pelo método Single Shooting e o método Data Shooting. A figura 5.3e mostra uma superfície muito mais irregular, em comparação com 5.3a, convexa e bem comportada. Por outro lado, a reta vertical com os valores nominais para  $\theta_1$  sequer pode ser desenhada no gráfico 5.3d pois o mínimo da função se encontra no intervalo [4.5, 5.0], enquanto o valor nominal para esse parâmetro é 3.2434.

O gráfico 5.3d evidencia portanto o já discutido tradeoff entre viés e variância (bias-variance tradeoff) para o método DS. Apesar de suavizar a superfície da função e, nesse caso, gerar uma função convexa, os valores de  $\theta$  que minimizam a função não correspondem exatamente aos valores nominais, demarcados pelas retas magenta e ciano.

O gráfico 5.3h, por outro lado, se encontra centralizado exatamente nos valores nominais. Isso indica que o método SS consegue, em teoria, convergir para esses valores. Felizmente, como podemos verificar na figura 5.4, mais especificamente no gráfico 5.4d, o método DS consegue se aproxima do valor nominal quando aumentamos a quantidade de dados de treinamento. Note essa propriedade comparando os gráficos 5.4d e 5.3d.



(a) Gráfico tridimensional para DS, amplo.



(c) Gráfico de contorno para DS, amplo. Single Shooting



(e) Gráfico tridimensional para SS, amplo.



(g) Gráfico de contorno para SS, amplo.

(h) Gráfico de contorno para SS, reduzido.

**Figura 5.3:** Comparação entre curva de otimização dos métodos Data Shooting e Single Shooting para o problema LVP. Gráficos com indicação de "amplo" mostram o perfil da função objetivo percorrendo um grande intervalo de valores para os parâmetros. Já os classificados como "reduzido", mostram o comportamento da função em um intervalo menor, centralizado no valor mínimo da função. Cálculos realizados com coleta de 10 amostras igualmente espaçadas para cada variável de estado.



(b) Gráfico tridimensional para DS, reduzido.



(d) Gráfico de contorno para DS, reduzido. Single Shooting



(f) Gráfico tridimensional para SS, reduzido.





(a) Gráfico tridimensional para DS, amplo.







(e) Gráfico tridimensional para SS, amplo.

Single Shooting

5.0 4.5

4.0

3.5 3.0

2.5

2.0

1.5

1.0

0.5 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

 $\theta_2$ 



(b) Gráfico tridimensional para DS, reduzido.



(d) Gráfico de contorno para DS, reduzido. Single Shooting



(f) Gráfico tridimensional para SS, reduzido.



(g) Gráfico de contorno para SS, amplo.

 $\theta_1$ 

(h) Gráfico de contorno para SS, reduzido.

**Figura 5.4:** Comparação entre curva de otimização dos métodos Data Shooting e Single Shooting para o problema LVP. Gráficos com indicação de "amplo" mostram o perfil da função objetivo percorrendo um grande intervalo de valores para os parâmetros. Já os classificados como "reduzido", mostram o comportamento da função em um intervalo menor, centralizado no valor mínimo da função. Cálculos realizados com coleta de 30 amostras igualmente espaçadas para cada variável de estado.

Objective Function



(a) Gráfico tridimensional para DS, amplo.



(c) Gráfico de contorno para DS, amplo. Single Shooting



(e) Gráfico tridimensional para SS, amplo.



(g) Gráfico de contorno para SS, amplo.

(h) Gráfico de contorno para SS, reduzido.

**Figura 5.5:** Comparação entre curva de otimização dos métodos Data Shooting e Single Shooting para o problema FHN. Gráficos com indicação de "amplo" mostram o perfil da função objetivo percorrendo um grande intervalo de valores para os parâmetros. Já os classificados como "reduzido", mostram o comportamento da função em um intervalo menor, centralizado no valor mínimo da função. Cálculos realizados com coleta de 10 amostras igualmente espaçadas para cada variável de estado.



(b) Gráfico tridimensional para DS, reduzido.



(d) Gráfico de contorno para DS, reduzido. Single Shooting



(f) Gráfico tridimensional para SS, reduzido.



Destacamos por fim o problema FHN, como visto em 4.2.8, amplamente conhecido na literatura por sua estrutura não-linear. Mais uma vez, temos uma diferença notável entre os métodos comparados. Ressaltamos novamente o *trade-off*: o método DS apesar de suavizar a função objetivo acrescenta um viés considerável.

Desse modo, focamos nessa seção em apresentar resultados ilustrativos acerca da capacidade de suavização da função objetivo por parte do método DS. Na próxima seção apresentaremos a avaliação dos quatro modelos comparados nesse estudo, apontando as principais vantagens e desvantagens das novas abordagens propostas nesse trabalho.

#### 5.1.2 Estimação

Na subseção anterior contrastamos o método de Data Shooting com o Single Shooting focando em suas funções objetivo. Exploramos agora os resultados para o problema de estimação de parâmetros focando na eficácia e eficiência dos quatro métodos de interesse: Single Shooting (SS), Data Shooting (DS), Single Shooting Regularized (SSR) e Data to Single Shooting (DSS).

Todas as métricas para os diferentes casos de teste foram obtidos através da mesma configuração dos experimentos. Isto é, os resultados correspondem a dados agregados de 30 execuções do procedimento de estimação. Em cada rodada foram geradas 5 simulações destinadas para uso em seleção de hiperparâmetros e treinamento através do processo Leave One Out Cross Validation. Para o conjunto de teste, 10 simulações foram geradas. Cada simulação gerada é composta por 10 amostras igualmente espaçadas no tempo de cada variável de estado do problema em questão.

A variância do ruído artificial adicionado é proporcional a 10% do sinal original, controlada através da variável  $\zeta$  (referir à discussão em 4.1.1) para todos os problemas. A única exceção é o problema LVP, onde utilizamos o valor de 1%. Isso foi necessário devido ao seu caráter oscilatório de alta frequência. Um nível de ruído na casa das dezenas se mostrou muito destrutivo, impossibilitando a estimação das variáveis.

Iremos discutir os resultados obtidos utilizando as métricas apresentadas na seção 4.1.3: Normalized Root Mean Squared Error (NRMSE), Success Rate (SR), e Overall Efficiency (OE). O foco permanecerá nas métricas OE e SR. Contudo, os resultados de OE dependem dos valores de SR, que por sua vez dependem de NRMSE. Por esse motivo, começaremos com uma breve exploração dos erros obtidos por cada um dos métodos em análise.

A métrica Normalized Root Mean Squared Error será sempre utilizada ao longo desta seção para avaliar o erro dos modelos estimados com relação ao conjunto de teste. O conjunto de teste contém dados que simulam experimentos reais e que foram isolados da etapa de treinamento. Primeiro destacamos o problema CCGO, detalhado na subseção 4.2.3. A figura 5.6 mostra alguns resultados obtidos pelos diferentes métodos para esse problema.

Nesse exemplo, apenas os resultados dos métodos DSS, SS e SSR apontam para uma estimação bem sucedida. Isso porque, conforme a definição da Success Rate, todos atingiram um NRMSE menor que o Value To Reach (VTR) designado, de 0.125. Outro exemplo de resultados obtidos é o da figura 5.7. Aqui temos o problema MHP (subseção 4.2.5), e vemos que todos são bem sucedidos nesse caso. Em ambos os exemplos, é interessante notar que DS, apesar de gerar parâmetros piores do ponto de vista do NRMSE, eles parecem ser bastante efetivos como parâmetros de referência para o método DSS.

Não vamos, porém, prosseguir nossa discussão observando tais resultados pontuais. Eles são provenientes de uma rodada de estimação apenas. Para que nossas conclusões sejam válidas, devemos considerar os dados agregados. Introduzidos esses exemplos, iremos focar agora nas demais métricas que melhor sumarizam os resultados das 30 rodadas de execução.



Figura 5.6: Comparação entre curvas estimadas e dados experimentais para o problema CCGO.



Figura 5.7: Comparação entre curvas estimadas e dados experimentais para o problema MHP.

Observando os resultados agregados para os casos de teste mais simples, vemos um padrão condizente com o obtido por Gábor e Banga [GB15]. A utilização de regularização em alguns casos pode gerar resultados piores, dado o viés introduzido em troca da diminuição da variância do modelo. Tanto nas imagens da figura 5.8 quanto nas de 5.9 e 5.10, vemos que os métodos que passaram por regularização (DSS e SSR) se saem pior que os métodos sem regularização (DS e SS).

Por outro lado, já podemos notar nesses gráficos a enorme eficiência de um dos métodos propostos. Data Shooting atinge uma Taxa de Sucesso (Success Rate) menor que a do método Single Shooting para o problema FIC a menos de um sétimo de seu custo. Enquanto a média do tempo gasto por SS chegou a aproximadamente 63 segundos, DS obteve resultados melhores e em pouco mais de 8 segundos em média. Para os problema FRC (subseção 4.2.2) e CCGO (subseção 4.2.3) os resultados são similares, com a diferença de custos entre DS e SS chegando a 3 ordens de magnitude.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



(b) Boxplot do erro de teste. (c) Gráfico de eficiência geral.

Figura 5.8: Avaliação de modelos para o problema FIC.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



(b) Boxplot do erro de teste. (c) Gráfico de eficiência geral.

Figura 5.9: Avaliação de modelos para o problema FRC.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



Figura 5.10: Avaliação de modelos para o problema CCGO.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



(b) Boxplot do erro de teste. (c) Gráfico de eficiência geral.

Figura 5.11: Avaliação de modelos para o problema BEP.







Figura 5.12: Avaliação de modelos para o problema MHP.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



Figura 5.13: Avaliação de modelos para o problema LVP.

Ambos os métodos sem regularização obtém resultados melhores apenas nos três primeiros casos de teste. Os casos estão ordenados de maneira crescente com relação a complexidade de suas EDOs. Notamos que a regularização se torna mais eficaz a medida que os sistemas de EDOs passam a apresentar uma estrutura mais desafiadora. A complexidade do modelo se torna uma potencial fonte de mal condicionamento e não convexidade, cenários estes amenizados pela regularização.

No problema BEP (subseção 4.2.4), mais especificamente no gráfico 5.11b, notamos que os modelos SS e SSR conseguem convergir para o mínimo global apenas em algumas das rodadas de experimentos, como evidenciado pela distribuição dos erros. Enquanto isso, tanto DS quanto DSS alcançam esses valores ótimos sem problemas. Como já discutido, o método DS produz superfícies mais suaves, facilitando a convergência do algoritmo de otimização. O DSS por sua vez, utiliza esses valores obtidos por DS como ponto de partida para o algoritmo de otimização e também como parte da estratégia de regularização por Tikhonov.

Os resultados do problema MHP (subseção 4.2.5), na figura 5.12, introduzem um novo cenário. Aqui, enfim, os métodos não regularizados se saem pior que os métodos regularizados. A reduzida Taxa de Sucesso do método SS pode ser explicada pelas dificuldades relacionadas ao problema de otimização não-linear somado ao mal condicionamento. Já o método DS é prejudicado pelo *trade-off* entre viés e variância, se saindo muito pior. Ressaltamos que nesse caso os gráficos 5.12a e 5.12b podem não indicar facilmente o melhor método entre SS, SSR e DSS. Por outro lado, o gráfico 5.12c, considerando tanto SR quanto custo computacional, aponta claramente DSS como o mais eficiente.

No problema LVP (subseção 4.2.6) temos de forma ainda mais acentuada a atuação do viés do método DS, afetando significativamente a convergência do método. Mas note que aqui também fica evidente a vantagem de sua utilização no processo de regularização em dois passos. Todos os métodos chegam à praticamente mesmo Taxa de Sucesso. Porém, SSR é mais de duas vezes mais custoso que DSS.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



(b) Boxplot do erro de teste. (c) Gráfico de eficiência geral.

Figura 5.14: Avaliação de modelos para o problema BBG.







(b) Boxplot do erro de teste. (c) Gráfico de eficiência geral.

Figura 5.15: Avaliação de modelos para o problema FHN.



(a) Gráfico do inverso da taxa de sucesso versus custo computacional.



Figura 5.16: Avaliação de modelos para o problema MAPK.

A figura 5.14, referente ao problema BBG (subseção 4.2.7), apresenta resultados similares aos já discutidos para o problema BEP. Já os resultados do problema FHN (subseção 4.2.8), na figura 5.15, mostram uma grande vantagem do método DSS. Os métodos SS e DS não atingem o VTR em nenhuma rodada. Por outro lado, DSS e SSR, por conta da regularização, atingem uma Taxa de Sucesso de 90% e 70%, respectivamente. Mais do que isso, DSS consegue não apenas superar SSR com relação a Success Rate mas também é em média mais de 4 vezes mais rápido.

O problema MAPK (subseção 4.2.9), por fim, é o que possui o sistema de EDOs mais complexo de todos os casos de teste estudados. Os resultados obtidos são similares aos vistos em MHP. E aqui, mais uma vez, observamos a vantagem dos método regularizados, com DSS gerando resultados superiores que as alternativas. Aproveitamos para destacar que esse é um exemplo claro em que o método DS não gera uma função objetivo convexa. Isso ocorre pois as derivadas dos vetores de estados de MAPK não são funções lineares nos parâmetros. Essa propriedade se evidencia na distribuição dos erros de DS, onde temos uma grande dispersão.

Apesar disso, destacamos aqui uma propriedade interessante de DS, já prenunciado nos exemplos das figuras 5.6 e 5.7. O método obtém valores inferiores na métrica NRMSE em comparação com SS nos problemas LVP, MHP, FHN e MAPK, como vemos nos gráficos 5.13b, 5.12b, 5.15b e 5.16b, respectivamente. Ainda assim, DS proporciona parâmetros de referência melhores que SS para o segundo passo do processo de regularização. Afirmamos isso já que DSS claramente atinge uma Taxa de Sucesso equivalente ou então maior em comparação com SSR nesses casos. Além disso, a discrepância de custo computacional entre DSS e SSR é notável, com o segundo método chegando a ser mais de duas vezes mais custoso que o primeiro.

Esse é um fenômeno interessante e merece uma consideração mais detalhada. Ele pode ser entendido quando consideramos as superfícies sendo otimizadas e a inevitável atração do otimizador a mínimos locais. Todos os problemas de otimização gerados pelos casos de teste foram solucionados utilizando

o algoritmo Simulated Annealing (SA), um otimizador global.

Porém, como tratado na seção 2.3.2, esse algoritmo emprega uma metaheurística no processo de otimização, fornecendo garantias de convergência para o mínimo global apenas assintoticamente. Para os casos mais simples, vemos que isso não é um problema. Contudo, MAPK e FHN apresentam superfícies de otimização muito mais desafiadoras devido à não linearidade em suas EDOs.

Mesmo empregando um algoritmo de otimização global, a função objetivo tem um papel fundamental no sucesso ou não do processo de estimação. Isso porque o algoritmo de otimização terá de explorar a superfície gerada pela função objetivo. Constatamos, através dos resultados, que se SS cai em um mínimo local na primeira etapa da regularização, isso impacta diretamente a segunda, prejudicando portanto o método SSR.

O gráfico 5.15b, referente ao problema FHN, ilustra especialmente bem isso. SS fica claramente preso em um mínimo local em todas as rodadas dos experimentos. Seus resultados se concentram na região de 0.4 na métrica de erro NRMSE. O método SSR também possui valores nessa região, como vemos em seu boxplot, e até mesmo DS e DSS encontram valores próximos a esse mínimo. Mas isso ocorre em poucos casos para esses dois últimos métodos, sendo apenas *outliers* da distribuição.

Os resultados de FHN podem parecer contraditórios inicialmente, mas não o são. Poderíamos esperar que SSR devesse se sair melhor em comparação com DSS, já que SS atinge erros menores em comparação com DS. Mas o que ocorre é que enquanto SS converge para um mínimo local e prejudica SSR, DS consegue, na maioria das vezes, convergir para o seu mínimo global, em torno de 0.55. Os parâmetros resultantes desse mínimo, por sua vez, se encontram mais próximos da base de atração do mínimo global da superfície do Single Shooting do que da base de atração de um mínimo local.

A figura 5.17 ilustra precisamente esse comportamento. Os parâmetros estimados por DS, apesar de gerarem curvas com NRMSE de 0.5136, são qualitativamente próximas do comportamento oscilatório presente nos dados. SS, por sua vez, converge para parâmetros que geram uma curva de menor erro, 0.4403, mas cujo comportamento é longe do esperado. Vemos que se trata de um mínimo global de forte atração, pois quando utilizado como referência e parâmetro inicial do processo de estimação no método DSS, o método converge para praticamente os mesmos valores.



Figura 5.17: Comparação entre curvas estimadas e dados experimentais para o problema FHN.

Com isso concluímos que o método de regularização em dois passos proposto por Gábor e Banga [GB15] apresenta essa armadilha. A superfície de otimização em ambas as etapas é praticamente a mesma. Assim, se o algoritmo de otimização fica preso em um mínimo local na primeira etapa, isso faz com que o valor de referência gerado para a segunda acabe a prejudicando, ao invés de auxiliá-la. O mesmo não é verdade para DSS. A superfície gerada por DS e SS são fundamentalmente diferentes, dado os cálculos que originam suas funções objetivo. Contudo, ressaltamos que não fornecemos garantias de que esse mesmo cenário não possa eventualmente ocorrer com DSS, apesar dos resultados observados não indicarem isso.

Feitas essas considerações, finalizamos essa seção concluindo que nenhum dos métodos atinge o melhor resultado consistentemente em todos os casos de teste. Para os problemas mais simples, FIC, FRC e CCGO, os métodos não regularizados Single Shooting e Data Shooting obtiveram os melhores resultados. Vemos que nesses casos a regularização em duas etapas prejudica tanto o custo computacional quanto a acurácia dos métodos Single Shooting Regularized e Data to Single Shooting.

Já os problemas MHP e FHN penalizam ambos os métodos não regularizados DS e SS. BEP e BBG, por outro lado, possuem uma estrutura não linear que prejudica os métodos SS e SSR, mas que não apresenta dificuldades para DS ou DSS. Em LVP, SS e DSS atingem a mesma Taxa de Sucesso. O mais complexo dos problemas, MAPK, é solucionado de maneira eficiente apenas por DSS, atingindo a melhor Taxa de Sucesso dentre os métodos avaliados.

Mesmo com todas essas nuances, fica evidente que os dois métodos propostos, DS e DSS, se destacam positivamente em comparação com o método clássico SS e sua versão regularizada, SSR. O método DS se mostra extremamente eficiente, com tempos de execução atingindo a faixa dos décimos de segundos, com ótimos resultados para os problemas mais simples. Em CCGO, por exemplo, DS atinge uma Taxa de Sucesso de 100% levando em média 0.15 segundos para obter seus resultados. SS também atinge 100% de Taxa de Sucesso, mas a um custo de 90 segundos. Já em BBG, DS obtém uma SR de 86% a um custo de 0.6 segundos. SS leva 1176 segundos e atinge uma Taxa de Sucesso de 3%.

Nos problemas onde DS não consegue bons resultados no processo de estimação, DSS se sai melhor que as alternativas com relação a Taxa de Sucesso e/ou ao custo computacional. Nos casos de teste LVP e FHN, por exemplo, enquanto a SR de Data Shooting é de 0% para ambos, Data to Single Shooting atinge Taxas de 96% e 90%, respectivamente. No caso de teste FHN, em que tanto DS quanto SS atingem 0% de SR, DSS atinge uma Taxa de 90%. SSR atinge uma Taxa de 70%. Além de superar todos os demais métodos com relação a Taxa de Sucesso, DSS gera esses resultados em menos de um quarto do tempo de SSR.

Os resultados das principais métricas discutidas nesse capítulo são apresentados nos gráficos de radar da figura 5.18. No eixo horizontal temos as métricas de Success Rate e Overall Efficiency. Portanto, quanto maiores os valores atingidos nesse eixo, melhor. Já no eixo vertical as métricas são de NRMSE Médio e Custo (Computacional) Médio. Aqui, quanto menor os valores, melhor. Os dados brutos podem ser encontrados no apêndice A, tabela A.1.



(c) Métricas para o caso de teste CCGO.



(f) Métricas para o caso de teste LVP.



(i) Métricas para o caso de teste MAPK.

Figura 5.18: Gráficos de radar para visualização das métricas de avalização dos modelos. No eixo vertical, valores mais próximos do centro são melhores. No horizontal, valores mais próximos do centro são piores.

### 5.2 Código Fonte

Faz parte dos objetivos deste trabalho a implementação dos métodos propostos. Por esse motivo, destacamos aqui que seu código pode ser encontrado no repositório <a href="https://github.com/atgmello/DEParamEst">https://github.com/atgmello/DEParamEst</a>. Sua utilização é detalhada no README, encontrado na raiz do repositório. É possível ter acesso não apenas ao código como também a todos os dados utilizados nos gráficos apresentados na seção anterior. Esses dados encontram-se serializados e podem ser facilmente recuperados.

O código disponibilizado cobre não apenas a implementação dos métodos, mas todo o processo de estimação de parâmetros e análise dos dados. Portanto, ele pode ser utilizado para se obter todos os resultados aqui apresentados. Desde a realização dos experimentos e simulações, até o tratamento dos dados gerados e confecção dos gráficos discutidos. Com isso, esperamos que o trabalho seja facilmente reproduzível e replicável. O acesso ao código fonte também permite uma análise detalhada da implementação, podendo impulsionar trabalhos correlatos.

O seu desenvolvimento foi feito seguindo as principais recomendações para otimização de código na linguagem Julia. Apesar de ser uma linguagem de alto nível, para que Julia possa atingir velocidades próximas às linguagens C ou Fortran é necessário o uso não apenas idiomático da linguagem, mas também a aplicação de técnicas específicas. Dado o custo elevado do problema atacado, tanto por conta dos métodos numéricos utilizados quanto do processo estocástico envolvido, o entendimento e utilização de técnicas mais avançadas foi necessário, como garantia de estabilidade de tipos, préalocação de memória, entre outras.

Ainda visando produzir um código robusto e eficiente, foram utilizadas técnicas de computação paralela. A necessidade de execução de diversas rodadas do procedimento de estimação, por conta do processo estocástico, impôs essa necessidade. Do contrário o tempo computacional do programa desenvolvido se tornaria proibitivo. Felizmente, a realização de cada experimento pode ser paralelizada de forma relativamente simples uma vez que se tratam de processos independentes e dado o suporte oferecido pela linguagem a esse paradigma. A forma de paralelismo utilizada, portanto, é a execução de trechos *thread-safe* do código, distribuídos através dos núcleos do computador.

Quanto aos aspectos legais, a licença escolhida para disponibilização do código é a Licença MIT. Trata-se de uma licença Open Source permissiva, dando liberdade para utilização, modificação e redistribuição do código. Com isso reforçamos o compromisso da pesquisa com a comunidade Open Source, da qual esse trabalho se beneficiou integralmente, sem a qual não teria sido possível atingir os mesmos resultados em termos de qualidade de código. Essa é a licença mais utilizada pelos desenvolvedores de pacotes para Julia, além da própria linguagem ser disponibilizada sob essa licença, sendo portanto a mais coerente a ser adotada nesse contexto.

Desse modo finalizamos o capítulo referente aos resultados. Começamos com a comparação dos métodos tradicionais e os propostos nesse trabalho, analisando primeiramente as diferenças entre as superfícies de otimização dos métodos DS e SS. Em seguida, apresentamos e discutimos os resultados dos problemas de otimização originados dos casos de teste selecionados para os métodos SS, SSR, DS e DSS. Por fim, ressaltamos aqui, como um dos resultados deste trabalho, o software desenvolvido. No próximo capítulo apresentamos a síntese desta pesquisa e realizamos as considerações finais.

#### 78 RESULTADOS E DISCUSSÃO

### Capítulo 6

## Conclusões

Após desenvolver os fundamentos, definir novos conceitos e discutir os resultados obtidos, oferecemos nesse capítulo as considerações finais. Começamos com a síntese do trabalho realizado, na seção 6.1. Nesse seção recapitulamos os principais conceitos apresentados nos capítulos anteriores, destacando as propriedades dos métodos propostos e sua comparação com as alternativas clássicas. A seção 6.2 destaca as contribuições deste trabalho. Por fim, tratamos de possíveis trabalhos futuros, na seção 6.3.

#### 6.1 Síntese do Trabalho

Problemas Inversos (PI) são comuns em diversas áreas da ciência. Esse tipo de problema se materializa quando pesquisadores desejam modelar determinado fenômeno mas não conseguem descrevê-lo apenas a partir de princípios teóricos. Nesse caso, o modelo proposto conta com alguns parâmetros que precisam ser ajustados através de dados experimentais, para que possa então ser utilizado de fato.

No âmbito da bioquímica, PIs se apresentam no estudo de reações metabólicas. Utilizando como base conceitos da cinética química, é possível extrair sistemas de Equações Diferenciais Ordinárias (EDOs) a partir de reações. Os dados, por sua vez, podem ser coletados a partir de experimentos in vitro. Assim, em posse de modelo e dados, o próximo passo para solução do PI é recorrer às técnicas de estimação de parâmetros.

Problemas Inversos vindos dessa área são particularmente suscetíveis a dois fatores bem conhecidos dos estudos de otimização. São eles o mau condicionamento e a não convexidade. São duas propriedades que dificultam a solução do problema de estimação. O mau condicionamento leva à instabilidade de métodos numéricos, impossibilitando sua a convergência para um mínimo. Já a não convexidade pode levá-los a resultados subótimos. O trabalho visa minimizar tais problemas.

Apresentamos esse contexto e os objetivos do trabalho no capítulo 1. Como possível solução às dificuldades mencionadas, propomos dois novos métodos a serem utilizados na estimação de parâmetros, intitulados Data Shooting (DS) e Data to Single Shooting (DSS). Antes de defini-los, tratamos, no capítulo 2, dos fundamentos para que esses métodos possam ser introduzidos no capítulo seguinte.

O capítulo de Fundamentos Teóricos (2) se divide em três. Primeiro, apresentamos o conceito de Equações Diferenciais Ordinárias (EDOs). Trata-se de um modelo matemático determinístico utilizado para modelagem de reações químicas. Além de detalhar suas propriedades ao longo do capítulo, tratamos especificamente, na subseção 2.1.2, de técnicas de integração numérica.

Ressaltamos que técnicas de integração numérica são de grande importância para o trabalho. Elas possibilitam a solução de Problemas de Valor Inicial (PVI). PVIs são compostos por EDOs e um

valor inicial para seu vetor de estados. Solucionar PVIs é uma etapa fundamental no processo de estimação de parâmetros.

Em seguida, a seção 2.2 faz a conexão entre cinética química e Equações Diferenciais Ordinárias e como sistemas de EDOs podem ser obtidos a partir de reações. Esse capítulo tem como propósito melhor estabelecer o contexto inicial do trabalho, ressaltando sua aplicação em casos práticos dessa área. Esse capítulo também aborda a estimação de parâmetros, na seção 2.3. Cobrimos a teoria dos Mínimos Quadrados (2.3.1), noções de otimização (2.3.2) e também regularização (2.3.3).

Os novos métodos propostos para a família dos Shootings são apresentados no capítulo 3. Começamos introduzindo os métodos clássicos, Single Shooting (SS) e Multiple Shooting (MS) (3.1). São métodos que utilizam a técnica dos Mínimos Quadrados para a estimação de parâmetros.

Por estarmos lidando com funções estimadoras que envolvem EDOs, e não equações comuns, as funções objetivo geradas são muito mais complexas. É preciso solucionar um PVI em cada iteração do processo de otimização. Essa complexidade dá origem a superfícies de otimização não convexas, originadas de funções altamente não-lineares.

Visando lidar com esse problema, introduzimos na seção seguinte (3.2) o Data Shooting. A simplicidade de sua fórmula se reflete em uma superfície de otimização mais bem comportada, podendo ser até mesmo convexa em casos particulares. Por outro lado, a função objetivo gerada por esse método não necessariamente leva aos mesmos mínimos encontrados pelos métodos tradicionais de Shooting. Essa troca pode ser entendida como um *tradeoff* entre viés e variância.

O método Data to Single Shooting, por sua vez, combina os métodos Data Shooting e Single Shooting e é introduzido na seção 3.3. DSS emprega o esquema de regularização em duas etapas proposto por Gábor e Banga [GB15]. Através desse método, desejamos fazer uso dos pontos fortes dos métodos SS e DS, ao mesmo tempo que minimizamos seus pontos fracos.

Para que seja efetiva, a técnica de regularização por Tikhonov proposta depende de bons valores de referência para os parâmetros sendo estimados. Gábor e Banga admitem que não é comum que pesquisadores possuam essa informação a priori. A regularização em duas etapas é sugerida por eles nesse cenário. Chamamos de Single Shooting Regularized esse método de duas etapas descrito pelos autores, que utiliza SS em ambas etapas.

Em contraste, no método DSS, o Data Shooting é utilizado na primeira etapa do processo de regularização, por se mostrar simples e eficiente. Mesmo não convergindo para o mínimo esperado por conta do viés introduzido, ele fornece bons valores de referência para a segunda etapa, solucionada com Single Shooting.

Uma vez definidos os novos métodos, seguimos, no capítulo 4, tratando de detalhes conceituais que possibilitam a compreensão e reprodução dos resultados. A seção 4.1, muito importante para esse capítulo, trata de detalhes de implementação. Primeiro, discutimos como os experimentos são simulados, uma vez que todos os testes envolvendo os diferentes métodos são feitos com dados in silico. Explicamos como esses dados são gerados e também o processo estocástico envolvido na introdução de um ruído artificial às amostras.

A seleção de hiperparâmetros (4.1.2) também merece destaque, devido ao papel central do processo de regularização. Abordamos também a métrica Normalized Root Mean Square Error (NRMSE). Por fim, detalhamos o processo de avaliação dos modelos (4.1.3), possibilitando a correta interpretação dos resultados apresentados. Tratamos dos conceitos de Success Rate (SR) e Overall Efficiency (OE), adaptados de Villaverde e colaboradores [VFW<sup>+</sup>19].

Na seção 4.2 elencamos os nove PVIs utilizados para comparar os diferentes métodos de estimação. Esses problemas são organizados em ordem de complexidade de suas equações, sendo os seis primeiros provenientes de Floudas [FPA<sup>+</sup>13] e os três subsequentes de Gábor e Banga [GB15]. Finalizamos o capítulo com a seção 4.3 tratando de aspectos práticos envolvidos na pesquisa. No capítulo 5 tratamos dos resultados obtidos. Primeiro, na seção 5.1 focamos em comparar os diferentes métodos, começando pela estrutura de suas superfícies de otimização, na subseção 5.1.1. Mostramos como as superfícies geradas pelo DS são bem comportadas, contrastando-as com as obtidas pelo método SS. Utilizamos alguns dos casos de teste para isso.

Em seguida, comparamos os resultados do processo de estimação de parâmetros, na subseção 5.1.2. Nenhum dos quatro métodos comparados se saiu bem em absolutamente todos os casos de teste. Contudo, no geral, os métodos propostos DS e DSS obtiveram melhores resultados quando comparados com SS e SSR tanto em relação à qualidade da estimação quanto em relação à performance.

Destacamos que DS apresentou um custo computacional ordens de magnitude menor que os demais. No caso de teste BBG, por exemplo, o custo computacional médio do DS fica em torno de 0.6 segundos, enquanto que SS chega a levar quase 1177 segundos para completar o processo de estimação. Além disso, ele atinge uma Taxa de Sucesso superior a 80% para cinco dos nove casos de teste. No entanto, esse método sofre em casos de teste complexos, chegando a taxas de sucesso de 0% para o caso de LVP, por exemplo.

Por outro lado, o método DSS cobre bem essa fraqueza, se saindo particularmente bem nesses casos complexos em que DS falha. Isso é, DSS atinge uma Taxa de Sucesso igual ou superior a 80% para os casos de teste MHP, LVP, FHN. Em MAPK, o modelo mais desafiador dentre os selecionados, DSS atingiu uma Taxa de Sucesso de aproximadamente 36%, menor que os resultados obtidos nos casos de teste anteriores, mas a maior Taxa dentre os métodos comparados.

Por fim, na subseção 5.2 ressaltamos como resultado deste trabalho a implementação dos métodos propostos e de todo o arcabouço necessário para execução dos experimentos e processamento dos dados. Fornecemos o endereço para o código fonte e os dados utilizados, de modo que pesquisadores interessados possam replicar o estudo.

### 6.2 Contribuições

Elencamos aqui as principais contribuições do trabalho à área de estimação de parâmetros em Equações Diferenciais Ordinárias.

- Introdução de dois novo métodos para a estimação de parâmetros de EDOs, com o objetivo de melhor lidar com as dificuldades impostas por Problemas Inversos.
  - O método Data Shooting possui como principal característica a simplificação da superfície de estimação de problemas envolvendo Equações Diferenciais Ordinárias, podendo gerar funções convexas em casos particulares.
  - O método Data to Single Shooting cobre as fraquezas de DS e soluciona um problema relevante mencionado por Gábor e Banga [GB15]. A abordagem de regularização em duas etapas depende que cientistas conheçam a priori bons valores de referência para os parâmetros a serem estimados, o que não é comum na prática. DS atua então como um método capaz de gerar bons valores de referência para a regularização, tornando DSS mais eficiente que a alternativa proposta pelos autores.
  - Apesar do contexto inicial da bioquímica, ambos os métodos desenvolvidos são de aplicação geral, podendo ser utilizados em diferentes domínios da ciência que façam uso de Equações Diferenciais Ordinárias.
- Implementação do método proposto, análise de seus resultados práticos, disponibilização do código e dos dados experimentais.
  - A análise realizada corrobora o principal argumento de Gábor e Banga [GB15], que afirmam a importância da regularização no processo de estimação de parâmetros de sistemas dinâmicos, especialmente no contexto da bioquímica.

 A disponibilização do código e dos dados gerados possibilita a replicação do trabalho, além de encorajar sua continuidade ou o surgimento de trabalhos correlatos, estimulando e facilitando o avanço da discussão nessa área.

### 6.3 Trabalhos Futuros

Após explicitar as contribuições do trabalho na seção anterior, prosseguimos sugerindo agora caminhos para trabalhos futuros. Começamos ressaltando que o trabalho procurou cobrir com suficiente largura e profundidade diferentes áreas do conhecimento, desde conceitos de sistemas dinâmicos, à cinética química, otimização numérica, problemas inversos, regularização, além de passar por conceitos tangenciais à área de aprendizagem de máquina.

Apesar do embasamento teórico dos fundamentos que justificam os métodos propostos, houve grande foco na parte prática e em sua implementação, verificando empiricamente sua aplicabilidade. Desse modo, como trabalho futuro, apontamos que seria valioso um estudo mais aprofundado acerca dos aspectos teóricos relacionados aos métodos propostos em si, fornecendo provas mais rigorosas de todas as suas propriedades matemáticas.

A comparação entre os métodos propostos e os métodos clássicos é feita através da solução de casos de teste selecionados da literatura. Apesar de complexos e não lineares, esses casos de teste possuem um número de parâmetros reduzido em comparação aos modelos mais elaborados da bioquímica. Sugerimos, portanto, a realização de mais testes com modelos maiores. Seria interessante também verificar a eficácia dos métodos utilizando dados coletados in vitro.

O processo de estimação de parâmetros foi conduzido utilizando um algoritmo de otimização global estocástico, denominado Simulated Annealing. Existem diversos outros métodos de otimização globais, locais ou mesmo híbridos que poderiam ser aplicados. Deixamos como sugestão, portanto, a verificação de algoritmos de otimização, possivelmente menos genéricos e melhor equipados para lidar com a estrutura dos métodos propostos.

A seleção de parâmetros para a etapa de regularização neste trabalho é realizada através do esquema Leave One Out Cross Validation (LOOCV). É uma técnica bem estabelecida, mas custosa. Além disso, a implementação do LOOCV utilizada é adaptada para o problema em particular envolvendo séries temporais. Gábor e Banga [GB15], por sua vez, aplicam outras abordagens, e.g. Robust Generalized Cross Validation (RGCV). Sugerimos o estudo e implementação dessas ou de outras abordagens alternativas e a verificação de seu impacto no processo de seleção de parâmetros.

Uma desvantagem do método DS é o fato de que todos os vetores de estado precisam ser observáveis para que possa executar os cálculos necessários. Isso nem sempre é possível, especialmente em experimentos químicos, em que alguns elementos são de difícil medição. Essa é, portanto, uma clara direção em que o método poderia ser melhorado. Alternativamente, o trabalho de Guimarães [Gui16] apresenta formas de reescrever EDOs que minimiza o número de variáveis de estado de difícil observação. Sua abordagem poderia ser utilizada como um passo de pré-processamento para DS.

Por fim, apesar da motivação inicial e contexto deste trabalho serem o de Equações Diferenciais Ordinárias provenientes da cinética química, os métodos propostos são completamente gerais. Isto é, Data Shooting e Data to Single Shooting poderiam ser aplicados em diferentes domínios, e, como sabemos, EDOs são prevalentes em diversas disciplinas. Assim, deixamos como última sugestão o estudo da aplicabilidade dos métodos aqui desenvolvidos em outras áreas do conhecimento.

Esperamos que as contribuições citadas na seção anterior tenham deixado claro a relevância deste trabalho e encorajem a continuação da pesquisa nesse tópico. Essa seção final sugere diversas linhas de continuidade, tanto práticas quanto teóricas. Com isso, esperamos ter avançado a discussão acerca da estimação de parâmetros em Equações Diferenciais Ordinárias bem como a importância da regularização, além de ter oferecido algumas alternativas robustas através dos métodos propostos.

## Apêndice A

# Métricas

A seguinte tabela agrega as principais métricas para cada um dos métodos estudados, em cada um dos casos de teste. A coluna Overall Efficiency conta com os valores obtidos para os métodos Single Shooting (SS), Single Shooting Regularized (SSR) e Data to Single Shooting (DSS). Data Shooting (DS) é excluído dos cálculos, como justificado na subseção 4.1.3. Os bons resultados do método DS são facilmente verificados pelas colunas de Success Rate, em porcentagem, e Custo Computacional, em segundos.

Problema	Método	NRMSE Médio	Success Rate (%)	Custo Médio (s)	Overall Efficiency
	DS	0.0862	100	8.3957	-
FIC	DSS	0.1005	90	54.4286	1.0
	$\mathbf{SS}$	0.0878	96.67	62.7307	0.93
	SSR	0.0939	90	108.6360	0.50
	DS	0.1066	86.67	0.2127	-
FRC	DSS	0.1320	70	101.502	0.80
	$\mathbf{SS}$	0.0977	100	116.6657	1.00
	SSR	0.1066	70	229.6422	0.35
	DS	0.0890	100	0.1535	-
CCGO	DSS	0.1127	80	82.2496	0.87
	$\mathbf{SS}$	0.0540	100	90.0230	1.0
	SSR	0.1172	76.67	169.7176	0.40
	DS	0.0810	100	0.1536	-
BEP	DSS	0.0794	100	77.8717	1.0
	$\mathbf{SS}$	0.1899	6.67	80.0199	0.0649
	SSR	0.1625	30	117.5460	0.1987
	DS	0.1543	10	0.2241	-
MHP	DSS	0.1153	80	224.0144	1.0
	$\mathbf{SS}$	0.1523	33.33	215.1932	0.4337
	SSR	0.1148	83.33	368.8969	0.6326
	DS	0.4724	0	10.3842	-
LVP	DSS	0.0927	96.67	65.8623	1.0
	$\mathbf{SS}$	0.0946	96.67	68.7752	0.9576
	SSR	0.1150	93.33	161.2559	0.3943
	DS	0.1095	86.67	0.6016	-
BBG	DSS	0.1083	90	286.9617	1.0
	$\mathbf{SS}$	0.3910	3.33	1176.6913	0.009
	SSR	0.3054	23.33	384.2863	0.1936
	DS	0.5338	0	0.2194	-
FHN	DSS	0.0971	90	186.4527	1.0
	SS	0.4000	0	655.0961	0.0
	SSR	0.1711	70	837.2326	0.1732
	DS	0.1627	20	9.8800	-
MAPK	DSS	0.1378	36.67	1043.1977	1.0
	$\mathbf{SS}$	0.1430	26.67	1317.5833	0.5758
	SSR	0.1381	33.33	2387.6698	0.3972

 Tabela A.1:
 Tabela com as principais métricas avaliadas para todos os casos de teste.

### Bibliografia

- [AFB06] Luís G Arnaut, Sebastiao Jose Formosinho e Hugh Burrows. Chemical kinetics: from molecular structure to chemical reactivity. Elsevier, 2006. 14
- [AFNKB09] Maksat Ashyraliyev, Yves Fomekong-Nanfack, Jaap A Kaandorp e Joke G Blom. Systems biology: parameter estimation for biochemical models. The FEBS journal, 276(4):886–902, 2009. 44
  - [AL07] Andreas Antoniou e Wu-Sheng Lu. Practical Optimization: Algorithms and Engineering Applications. Springer, 2007. 12, 19, 20
  - [BBV04] Stephen Boyd, Stephen P Boyd e Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 32
- [BCPB<sup>+</sup>08] Eva Balsa-Canto, Martin Peifer, Julio R Banga, Jens Timmer e Christian Fleck. Hybrid optimization method with general switching strategy for parameter estimation. BMC systems biology, 2(1):1–9, 2008. 20
  - [BEKS17] Jeff Bezanson, Alan Edelman, Stefan Karpinski e Viral B Shah. Julia: A fresh approach to numerical computing. SIAM Review, 59(1):65–98, 2017. 55
  - [BMA04] Julio R Banga, Carmen G Moles e Antonio A Alonso. Global optimization of bioprocesses using stochastic and hybrid methods. Em Frontiers in global optimization, páginas 45–70. Springer, 2004. 44
    - [Bre20] Tom Breloff. Juliaplots/plots.jl: v1.5.3, Junho 2020. 55
  - [BT<sup>+</sup>93] Dimitris Bertsimas, John Tsitsiklis et al. Simulated annealing. Statistical science, 8(1):10–15, 1993. 21, 44
  - [Car07] Robert W Carr. Modeling of chemical reactions, volume 42. Elsevier, 2007. 13, 14
  - [CC15] Steven C Chapra e Raymond P Canale. Numerical methods for engineers. McGraw-Hill New York, 2015. 8, 10, 20, 22
  - [CSB<sup>+</sup>16] Gunnar Cedersund, Oscar Samuelsson, Gordon Ball, Jesper Tegnér e David Gomez-Cabrero. Optimization in biology parameter estimation and the associated optimization problem. Em Uncertainty in Biology, páginas 177–197. Springer, 2016. 44
  - [CWLA] L Chen, R Wang, C Li e K Aihara. Modelling biomolecular networks in cells: Structures and dynamics. 2010. NY: Springer-Verlag London Limited. 13
  - [Dav13] Mark E Davis. Numerical methods and modeling for chemical engineers. Courier Corporation, 2013. 5, 10
- [DBDW06] Moritz Diehl, Hans Georg Bock, Holger Diedam e P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. Em Fast motions in biomechanics and robotics, páginas 65–93. Springer, 2006. 28

- [FHT01] Jerome Friedman, Trevor Hastie e Robert Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001. 22, 33
- [FPA<sup>+</sup>13] Christodoulos A Floudas, Panos M Pardalos, Claire Adjiman, William R Esposito, Zeynep H Gümüs, Stephen T Harding, John L Klepeis, Clifford A Meyer e Carl A Schweiger. Handbook of test problems in local and global optimization, volume 33. Springer Science & Business Media, 2013. 38, 50, 80
  - [GB15] Attila Gábor e Julio R Banga. Robust and efficient parameter estimation in dynamic models of biological systems. *BMC systems biology*, 9(1):74, 2015. 2, 22, 23, 33, 34, 35, 37, 38, 40, 41, 51, 52, 53, 57, 66, 73, 80, 81, 82
- [GFR94] William L Goffe, Gary D Ferrier e John Rogers. Global optimization of statistical functions with simulated annealing. *Journal of econometrics*, 60(1-2):65–99, 1994. 20
  - [Gob] Matthias Gobbert. Robertson's example for stiff differential equations. 11
- [Gui16] Amanda Sayuri Guimarães. Um algoritmo para simplificar sistemas de equações diferenciais que descrevem a cinética de reações químicas. Tese de Doutorado, Universidade de São Paulo, 2016. 33, 82
- [HA18] Rob J Hyndman e George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018. 39
- [Ham11] Franz Hamilton. Parameter estimation in differential equations: A numerical study of shooting methods. SIAM Undergraduate Research Online, 4:16–31, 2011. 1, 27
- [HLDVI12] Boris Houska, Filip Logist, Moritz Diehl e Jan Van Impe. A tutorial on numerical methods for state and parameter estimation in nonlinear dynamic systems. Em Identification for Automotive Systems, páginas 67–88. Springer, 2012. 1
  - [HW96] Ernst Hairer e Gerhard Wanner. Examples of Stiff Equations, páginas 2–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996. 11
  - [JAN<sup>+</sup>20] Daniel C. Jones, Ben Arthur, Tamas Nagy, Mattriks, Shashi Gowda, Godisemo, Tim Holy, Andreas Noack, Avik Sengupta, Darwin Darakananda, Adam B, Iain Dunning, Simon Leblanc, Keno Fischer, David Chudzicki, Morten Piibeleht, Alex Mellnik, Dave Kleinschmidt, Tom Breloff, Yichao Yu, Joey Huchette, Mike J Innes, inkyu, john verzani, Artem Pelenitsyn, Calder Coalson, Ciarán O'Mara, Elliot Saba e Gerhard Aigner. Giovineitalia/gadfly.jl: v1.3.0, Junho 2020. 55
- [LDR<sup>+</sup>10] Chen Li, Marco Donizelli, Nicolas Rodriguez, Harish Dharuri, Lukas Endler, Vijayalakshmi Chelliah, Lu Li, Enuo He, Arnaud Henry, Melanie I. Stefan, Jacky L. Snoep, Michael Hucka, Nicolas Le Novère e Camille Laibe. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology, 4:92, Jun 2010. 53
  - [LH95] Charles L Lawson e Richard J Hanson. Solving least squares problems, volume 15. Siam, 1995. 17
- [MHM09] Claas Michalik, Ralf Hannemann e Wolfgang Marquardt. Incremental single shooting—a robust method for the estimation of parameters in dynamical systems. Computers & Chemical Engineering, 33(7):1298–1305, 2009. 28
  - [MR18] Patrick Kofod Mogensen e Asbjørn Nilsen Riseth. Optim: A mathematical optimization package for Julia. Journal of Open Source Software, 3(24):615, 2018. 55
- [MWR<sup>+</sup>20] Patrick Kofod Mogensen, John Myles White, Asbjørn Nilsen Riseth, Tim Holy, Miles Lubin, Christof Stocker, Andreas Noack, Antoine Levitt, Christoph Ortner, Blake Johnson, Dahua Lin, Kristoffer Carlsson, Yichao Yu, Christopher Rackauckas, Alex

Williams, Ben Kuhn, Jeffrey Regier, cossio, Ron Rock, Thomas R. Covert, Takafumi Arakaki, Alexey Stukalov, Andrew Clausen, Benjamin Deonovic, Benoît Legat, Don MacMillen, Iain Dunning, Jarrett Revels e Josua Grawitter. Julianlsolvers/optim.jl: v0.22.0, Junho 2020. 55

- [Pet83] Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. SIAM journal on scientific and statistical computing, 4(1):136–148, 1983. 55
- [Pre07] William H Press. Numerical recipes 3rd edition: The art of scientific computing. Cambridge university press, 2007. 7, 9, 10, 13, 17
- [Rac17] Christopher Rackauckas. A comparison between differential equation solver suites in matlab. R, Julia, Python, C, Mathematica, Maple, and Fortran, The Winnower, 5:e153459, 2017. 55
- [RG16] Simon Rogers e Mark Girolami. A first course in machine learning. CRC Press, 2016. 39
- [RN17] Christopher Rackauckas e Qing Nie. Differential equations. jl–a performant and featurerich ecosystem for solving differential equations in julia. Journal of Open Research Software, 5(1), 2017. 55
- [Sau12] T Sauer. Numerical analysis, 2nd edn. george mason university, 2012. 32
- [SB13] Josef Stoer e Roland Bulirsch. Introduction to numerical analysis, volume 12. Springer Science & Business Media, 2013. 1, 7
- [SSP06] Zoltan Szallasi, Jörg Stelling e Vipul Periwal. System modeling in cellular biology. From Concepts to, 2006. 13
  - [TD] José JC Teixeira-Dias. Molecular physical chemistry. 13, 14
- [VFW<sup>+</sup>19] Alejandro F Villaverde, Fabian Fröhlich, Daniel Weindl, Jan Hasenauer e Julio R Banga. Benchmarking optimization methods for parameter estimation in large kinetic models. *Bioinformatics*, 35(5):830–838, 2019. 20, 41, 43, 80
  - [Vog86] CR Vogel. A constrained least squares method for nonlinear ill-posed problems. preprint, 1986. 22
  - [VTK04] Henning U Voss, Jens Timmer e Jürgen Kurths. Nonlinear dynamical system identification from uncertain and indirect measurements. International Journal of Bifurcation and Chaos, 14(06):1905–1933, 2004. 26
  - [VVP13] Donald Voet, Judith G Voet e Charlotte W Pratt. Fundamentals of biochemistry: life at the molecular level. Number 577.1 VOE. 2013. 16
  - [Wol06] John Wolberg. Data analysis using the method of least squares: extracting the most information from experiments. Springer Science & Business Media, 2006. 18