UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

JOSÉ OSVALDO DE SOUZA GUIMARÃES

**Computação evolutiva na resolução de equações
diferenciais ordinárias não lineares no espaço de Hilbert**

São Paulo

2009

JOSÉ OSVALDO DE SOUZA GUIMARÃES

**Computação evolutiva na resolução de equações
diferenciais ordinárias não lineares no espaço de Hilbert**

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

Área de Concentração: Engenharia de
Sistemas Eletrônicos
Orientador: Prof. Dr. Marcio Lobo Netto

São Paulo

2009

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

São Paulo, 25 de Março de 2009.

José Osvaldo de Souza Guimarães

Marcio Lobo Netto

Guimarães, José Osvaldo de Souza

Computação evolutiva na resolução de equações diferenciais ordinárias não lineares no espaço de Hilbert / José Osvaldo de Souza Guimarães. –ed. rev. -- orientador Marcio Lobo Netto. – São Paulo, 2009.
209 p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1. Equações diferenciais 2. Computação evolutiva 3. Polinômios de Legendre 4. Matrizes 5. Problemas do valor inicial
I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Minha Cris

Talvez um dia, o suor das virtudes e do intelecto molhe de paz o chão do planeta.

Nesse dia, as flores nascerão sorrindo para os olhos que choram poesia e teoremas.

Ao amigo ausente dessa nossa realidade efêmera, mas eternamente presente no mundo das idéias, dos ideais, da Ciência, Prof. Dr. Henrique Schützer Del Nero, cujo espírito permeia as indagações e inquietudes do Grupo de Ciência Cognitiva e é o responsável pelo meu regresso à Escola Politécnica.

Ao meu orientador Prof. Dr. Márcio Lobo Netto que, a despeito da diversidade de minha área de pesquisa, me recebeu como orientado e me guiou pela Academia.

Ao Prof. Dr. José Roberto Castilho Piqueira, pela maestria do seu curso, pela colaboração nos resultados e pesquisa desta tese, pelas agudas observações e, sobretudo, pela postura ética e companheira.

À minha esposa Cristine: amor e confiança, viço das idéias.

A todos os companheiros do grupo de Ciência Cognitiva, a meus filhos Hermes e Ariane e afilhados André e Bruno, ao grande parceiro, Wilson Carron e ao amigo Abílio pelos seus comentários e sugestões.

“Depois de tudo que escarnece e exalta,
depois de tudo, quando nada falta,
depois de tudo, falta muito mais!”

Giuseppe Ghiaroni (1919-2008)

RESUMO

GUIMARÃES, J. O. S. **Computação evolutiva na resolução de equações diferenciais ordinárias não lineares no espaço de Hilbert**. 2009. 209 p. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2009.

A tese apresenta um método para a solução dos problemas do valor inicial (*PVIs*) com margens de erro comparáveis às de métodos numéricos consagrados (*MN*), tanto para a função quanto para suas derivadas. O método é aplicável a equações diferenciais (*EDs*) lineares ou não, sendo o ferramental desenvolvido até a quarta ordem, que pode ser expandido para ordens superiores. A solução é uma expressão polinomial de alto grau com coeficientes expressos pela razão entre dois inteiros. O método se mostra eficaz mesmo em alguns casos em que os *MN* não conseguiram “dar a partida”. As resoluções são obtidas considerando que o espaço de soluções é um espaço de Hilbert, equipado com a base completa dos polinômios de Legendre. Em decorrência do método aqui desenvolvido, os majorantes de erros para a função e derivadas são determinados analiticamente por um cálculo matricial também deduzido nesta tese. Paralelamente a toda fundamentação analítica, foi desenvolvido o software *SAM*, que automatiza todas as tarefas na busca de soluções dos *PVIs*. A tese propõe e verifica a validade de um novo critério de erro no qual pesam tanto os erros locais quanto os erros globais, simultaneamente. Como subprodutos dos resultados já descritos, igualmente integrados ao *SAM*, obtiveram-se também: (1) Um critério objetivo para analisar a qualidade de um *MN*, sem necessidade do conhecimento de seu algoritmo; (2) Uma ferramenta para aproximações polinomiais de alta precisão para funções de quadrado integrável em determinado intervalo limitado, com um majorante de erro; (3) Um ferramental analítico para transposição genérica (linear ou não) dos *PVIs* até 4ª ordem, nas mudanças de domínio; (4) As matrizes de integração e diferenciação genéricas para todas as bases polinomiais do espaço de Hilbert.

Palavras-chave: Equações Diferenciais, Computação Evolutiva, Polinômios de Legendre, Matrizes Operacionais de Integração e Diferenciação, Problema do Valor Inicial.

ABSTRACT

GUIMARÃES, J. O. S. **Computação evolutiva na resolução de equações diferenciais ordinárias não lineares no espaço de Hilbert**. 2009. 209 p. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2009.

This thesis shows a new method to get polynomial solutions to the initial value problems (IVP), with an error margin comparable to the consecrate numerical methods (NM), for both the function and its derivatives. The method works with differential equations (DEs) linear or not, beeing the developed tolls available until 4th order, whose can be expanded to higher orders. The solution is a polynomial high degree expression with coefficients expressed by the ratio between two integers. The method behaves efficiently even in some cases that NM cannot get started. The resolutions are gotten considering that, the solution space is a Hilbert space, equipped with a complete set basis of Legendre Polynomials. Due the method here developed, the error's majoratives for the function and its derivatives are found analytically by a matrix calculus, also derived in this thesis. Beside all analytical foundation, a software (SAM) was developed to automate the whole process, joining all the tasks involved in the search for solutions to the IVP. This thesis proposes, verifies and validates a new error criterion, which takes in account simultaneously the local and global errors. As sub-products of the results described before, also integrated to the SAM, the following achievements should be highlighted: (1) An objective criterion to analyze the quality of any NM, despite of the knowledge of its algorithm; (2) A tool for a polynomial approximation, of high precision, for functions whose square is integrable in a given limited domain, with an error's majorative; (3) A tool-kit for a generically transpose (linear or not) of the IVPs' domain and form, taking into account its derivatives, until the 4th order; (4) The generic matrices for integration and differentiation for all the polynomial basis of the Hilbert space.

Keywords: Differential Equation, Evolutive Computation, Legendre's Polynomials, Operational Matrices of Differentiation and Integration, Initial Value Problem.

LISTA DE GRÁFICOS

Gráfico 2. 1. Resíduo (RS) ao substituir o polinômio de aproximação na ED.	33
Gráfico 2. 2. Erros (resíduo) de diferentes hipotéticos candidatos à melhor solução.	35
Gráfico 2. 3. Comparativo de critérios de erro.	37
Gráfico 2. 4. Solução numérica do PVI de 2ª ordem.	49
Gráfico 3. 1. O polinômio é uma aproximação de, no intervalo, com erro menor do que ϵ	51
Gráfico 3. 2. Polinômios de Legendre da ordem 0 até 5. Sendo n a ordem do polinômio, ele tem n raízes reais.	58
Gráfico 3. 3. Polinômios de Legendre de ordem 6 a 10.	58
Gráfico 3. 4. Polinômio de Legendre de ordem 21. À medida que aumenta a ordem do polinômio, os valores que ele assume entre os extremos do intervalo vão ficando cada vez menores, “preenchendo” blocos cada vez menores da função aproximada.	59
Gráfico 3. 5. Transposição não-linear que confina os valores de u entre -1 e 1	91
Gráfico 4. 1. Histograma dos coeficientes gerados. A linha pontilhada é a função de distribuição calculada.	97
Gráfico 4. 2. Histograma de números gerados que determinam o locus do crossing-over. A linha pontilhada corresponde aos valores assumidos pela função de distribuição $g(x)$	98
Gráfico 4. 3. Distribuição normalizada dos números obtidos a partir da função geratriz deduzida.	100
Gráfico 4. 4. Resultados de uma roleta viciada na qual quanto menor for o número, maior será a probabilidade de ele ser sorteado.	101
Gráfico 4. 5. Distribuição das probabilidades dos <i>loci</i> de <i>cross-over</i>	103
Gráfico 5. 1. Valores da derivada de maior ordem que o polinômio candidato produziu no intervalo de trabalho. A partir deles será obtida uma série de Legendre que, depois de integrada duas vezes, será comparada com o próprio candidato.	113
Gráfico 5. 2. Dispersão relativa entre os coeficientes obtidos pela matriz de integração e os coeficientes do polinômio candidato.	114
Gráfico 5. 3. Com a evolução, após 300 gerações, diminuiu a dispersão nos coeficientes.	115
Gráfico 5. 4. Resíduo decorrente da substituição da série aproximadora na ED.	115
Gráfico 5. 5. O SAM encontrou um polinômio de grau 9 que mimetiza a função e sua derivada ao longo de todo intervalo. No limite da resolução do gráfico, os pontos estão todos superpostos. A margem de erro é menor do que um milionésimo de pixel.	116
Gráfico 5. 6. Erro ao longo do intervalo na função e nas derivadas na comparação com a solução analítica. Note-se que os valores majorantes calculados são de fato bem maiores que os erros máximos.	117
Gráfico 5. 7. Dispersão dos coeficientes, em relação à derivada de maior ordem, usando o método de integração ode23.m.	124
Gráfico 5. 8. Coerência entre os coeficientes da função e os das derivadas obtidas pelo ode23.m com baixa precisão	125
Gráfico 5. 9. O aumento da precisão do método numérico diminuiu a dispersão externa.	127
Gráfico 5. 10. O aumento da precisão do método reduziu muito também a dispersão interna.	127
Gráfico 5. 11. Dispersão externa acusada pelo método ao analisar os pontos da solução exata.	129
Gráfico 5. 12. Dispersão interna dos coeficientes obtidos com o método ao analisar a solução exata.	129

Gráfico 5. 13. Diferença em relação à unidade da soma dos coeficientes. A limitação da quantidade de algarismos provoca pequenos erros nos coeficientes de um polinômio de Legendre escrito na base canônica, que aumentam conforme o aumenta grau do polinômio.	131
Gráfico 5. 14. Dispersão externa dos coeficientes na 1ª geração em três tentativas diferentes.	135
Gráfico 5. 15. Erro (resíduo RS) ao substituir ao polinômio na EDO. A imposição das CI (poly by SAM) aumentou um pouco o erro quadrático total, mas ele será reduzido no processo evolutivo.	136
Gráfico 5. 16. Dispersão externa dos coeficientes do polinômio de aproximação após 1000 gerações de evolução. A dispersão interna é nula.	138
Gráfico 5. 17. Resíduo (RS) obtido ao se substituir o polinômio na EDO.	139
Gráfico 5. 18. Erro do polinômio de aproximação em relação aos valores da função e derivadas.	140
Gráfico 5. 19. Dispersão externa dos coeficientes da série que aproxima a função.	142
Gráfico 5. 20. Gráficos da função e derivadas obtidas pelo SAM, comparados com o obtido pelo ode113, com a opção de precisão máxima. Neste limite de resolução, os gráficos são indistinguíveis, pois o erro é menor que milésimos de pixel.	143
Gráfico 6. 1. Resíduo que se obtém ao substituir o polinômio de aproximação na ED.	147
Gráfico 6. 2. Comparativo entre a função e suas derivadas com as do polinômio de aproximação. Para esta resolução gráfica, eles são indistinguíveis.	148
Gráfico 6. 3. Dispersão relativa (externa) dos coeficientes.	148
Gráfico 6. 4. Erros na função e nas derivadas ao longo do domínio após 200 gerações.	149
Gráfico 6. 5. Como essa série passou por um processo evolucionário mais longo, o critério LG fez com que o erro ficasse mais distribuído ao longo do intervalo.	150
Gráfico 6. 6. Solução para o PVI obtida pelo SAM e a solução analítica após as considerações sobre o valor de k (vide a discussão sobre a multiplicidade de soluções ao final do exemplo).	151
Gráfico 6. 7. Dispersão relativa dos coeficientes para o PVI proposto.	151
Gráfico 6. 8. Margens de erro, em relação à solução exata, do polinômio de aproximação e sua derivada.	152
Gráfico 6. 9. Resíduo resultante da substituição do polinômio de aproximação na ED.	153
Gráfico 6. 10. Concordância entre a função e derivadas exatas e as obtidas pelo SAM. Neste caso ele determinou a solução exata.	154
Gráfico 6. 11. Dispersão relativa dos coeficientes. Nesse caso, essa dispersão é na verdade fruto de um “ruído” da QGL, que terá reflexos nos majorantes de erro previstos.	154
Gráfico 6. 12. Erro entre o polinômio de aproximação e derivadas comparados com a solução exata.	155
Gráfico 6. 13. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.	156
Gráfico 6. 14. Comparativo entre a função e suas derivadas com as do polinômio de aproximação.	156
Gráfico 6. 15. Dispersão relativa (externa) dos coeficientes.	157
Gráfico 6. 16. Erros em relação à função e às derivadas.	157
Gráfico 6. 17. Comparativo dos resíduos entre o melhor polinômio — obtido por polinômios ortogonais — com o polinômio evoluído pelo SAM.	158
Gráfico 6. 18. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.	159
Gráfico 6. 19. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação. .	159
Gráfico 6. 20. Dispersão relativa (externa) dos coeficientes.	160

Gráfico 6. 21. Erros na função e derivadas. Neste caso os erros obtidos são um pouco maiores que os majorantes devido à limitação de dígitos do processador.....	160
Gráfico 6. 22. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i>	162
Gráfico 6. 23. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação. .	162
Gráfico 6. 24. Dispersão relativa (externa) dos coeficientes.	163
Gráfico 6. 25. Erros em relação à função e às derivadas.	163
Gráfico 6. 26. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i>	164
Gráfico 6. 27. Comparativo entre a solução de Adams (máxima precisão) e suas derivadas com as do polinômio de aproximação.	165
Gráfico 6. 28. Dispersão externa relativa dos coeficientes.	165
Gráfico 6. 29. Erros em relação à função e às derivadas.	166
Gráfico 6. 30. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i>	167
Gráfico 6. 31. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação. .	168
Gráfico 6. 32. Dispersão relativa (externa) dos coeficientes.	168
Gráfico 6. 33. Erros em relação à função e às derivadas.	169
Gráfico 6. 34. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i> . O fato de o erro estar regularmente distribuído no domínio mostra a proximidade da estagnação evolutiva com o critério <i>LG</i>	170
Gráfico 6. 35. Comparativo entre a solução do <i>MN</i> com as do polinômio de aproximação.....	170
Gráfico 6. 36. Dispersão relativa (externa) dos coeficientes.	171
Gráfico 6. 37. Erros em relação à função e às derivadas, em relação ao método de Adams.	171
Gráfico 6. 38. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i>	172
Gráfico 6. 39. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação. .	173
Gráfico 6. 40. Dispersão externa relativa dos coeficientes.	173
Gráfico 6. 41. Erros em relação à função e às derivadas.	174
Gráfico 6. 42. Comparativo de erros em relação ao resíduo no <i>PVI</i> entre a série de Legendre e o polinômio evoluído.	175
Gráfico 6. 43. Comparativo de erros em relação ao resíduo no <i>PVI</i> entre o polinômio evoluído e a série de Taylor.	175
Gráfico 6. 44. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na <i>ED</i>	176
Gráfico 6. 45. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação. .	177
Gráfico 6. 46. Dispersão externa relativa dos coeficientes.	177
Gráfico 6. 47. Erros em relação à função e às derivadas.	178
Gráfico I. 1. Gráfico da função e derivadas no intervalo próprio.	190
Gráfico I. 2. Gráfico da função e derivadas no intervalo de trabalho.	191
Gráfico I. 3. Dispersão relativa dos coeficientes nesse estágio da evolução.	193
Gráfico I. 4. Erro ao longo do domínio e a raiz do erro quadrático médio (<i>RMS</i>).....	194
Gráfico I. 5. O polinômio e sua derivada se superpõem aos pontos obtidos pelo método numérico de precisão.....	194
Gráfico I. 6. Além de o polinômio evoluído ter <i>RMS</i> menor, ele é bem comportado nos extremos do domínio.	195

Gráfico I. 7. Erro absoluto ao longo do intervalo para as sucessivas gerações.....	197
Gráfico I. 8. Erros no intervalo em que a função foi aproximada.....	200

LISTA DE FIGURAS

Figura 3. 1. Código em Matlab para obtenção da Matriz de Integração na base canônica.	70
Figura 3. 2. Código em Matlab para obter a matriz de diferenciação na base de Legendre.	76
Figura 3. 3. Código em Matlab para obter a Matriz de Diferenciação na base de Fourier.	79
Figura 3. 4. Código em Matlab para efetuar a transposição de domínios a partir de uma transformação genérica (tx2u) e sua inversa (zi).	86
Figura 3. 5. A semelhança de triângulos permite obter: $\frac{u - c}{d - c} = \frac{x - a}{b - a}$	89
Figura 4. 1. O diabo de Maxwell controlando o fluxo de partículas e produzindo um ambiente mais “organizado”.	104

LISTA DE TABELAS

Tabela 2.1. Classificação das <i>EDs</i> de acordo com a quantidade de variáveis independentes.....	31
Tabela 2.2. Linearidade nas <i>EDs</i>	32
Tabela 2.3. Comparação dos coeficientes dos polinômios. No processo evolutivo-adaptativo, alguns aumentaram, outros diminuíram de forma a satisfazer o critério de erro <i>LG</i>	37
Tabela 2.4. Comparativo dos <i>MNs</i> : o primeiro erro se refere à função e o segundo a sua derivada.....	47
Tabela 3. 1. Coeficientes do Polinômio de Legendre de ordem n na base canônica.	57
Tabela 3. 2. Coeficientes da série de Legendre da função e das derivadas integradas conforme as <i>CI</i> e os respectivos desvios padrão.	84
Tabela 3. 3. Precisão da <i>QGL</i> para diferentes funções	93
Tabela 5. 1. Coeficientes da série de Legendre após 60 gerações, partindo do polinômio nulo.....	112
Tabela 5. 2. Coeficientes da série de Legendre depois de 400 gerações.	117

LISTA DE SIGLAS

<i>AG</i>	Algoritmo Genético
<i>ED</i>	Equação Diferencial
<i>CI</i>	Condições Iniciais da Equação Diferencial
<i>ED_t</i>	Equação Diferencial após a transposição de domínio.
<i>MQ</i>	Mínimos Quadrados
<i>PVI</i>	Problema do Valor Inicial
<i>MN</i>	Método numérico de integração para o <i>PVI</i>
<i>RS</i>	Resíduo decorrente da substituição do polinômio de aproximação na <i>ED</i> ao longo do domínio.
<i>SAM</i>	Programa baseado no algoritmo genético diferencial, polinômios de Legendre e nas matrizes de integração e diferenciação que automatiza as tarefas relacionadas ao <i>PVI</i> , avalia qualidade de métodos numéricos e aproxima funções.
<i>QGL</i>	Quadratura de Gauss-Legendre
<i>RS</i>	Resíduo remanescente após a substituição da aproximação na <i>ED</i> .
<i>M_{LI}</i>	Matriz de Integração de Legendre
<i>M_{LD}</i>	Matriz de Diferenciação de Legendre
<i>M_{CI}</i>	Matriz de Integração – Base Canônica
<i>M_{CD}</i>	Matriz de Diferenciação – Base Canônica
<i>M_{FD}</i>	Matriz de Diferenciação – Base de Fourier
<i>LG</i>	Critério de erro Local × Global
<i>RMS</i>	$RMS = \sqrt{\int_a^b (\text{erro})^2 \cdot dx / (b - a)}$. Raiz do erro quadrático médio.
<i>erro_{LG}</i>	$\text{erro}_{LG} = (\text{erro})_{\text{máx}} \cdot \sqrt{\int_a^b (\text{erro})^2 \cdot dx / (b - a)}$

LISTA DE SÍMBOLOS

f^*	Série que aproxima a função f .
$f^{(n)}$	Derivada de ordem n da função f .
D_n	Derivada de ordem n da função f . (Notação do programa).
f_t	Função f após a transposição de domínio
D_{tn}	Derivada de ordem n da função f_t .
δ_{mn}	Delta de Kronecker
P_n	Polinômio de Legendre de Ordem n
C	Vetor dos coeficientes da expansão em série da função $f(x)$.
q	Ordem da Equação Diferencial
h	Passo de um processo numérico de integração
u	Variável no intervalo de trabalho
x	Variável no domínio original da função
$\vec{y}_0$	Vetor das condições iniciais do PVI
$\vec{y}_{0t}$	Vetor das condições iniciais do PVI com domínio transposto
Ω	Matriz Sanduíche
Z^T	Transposta da matriz Z

Sumário

CAPÍTULO 1. INTRODUÇÃO	20
1.1 OBJETIVOS.....	20
1.2 MOTIVAÇÃO	22
1.3 FUNDAMENTAÇÃO ANALÍTICA DESENVOLVIDA	23
1.4 ESCOLHA DO PROCESSO EVOLUTIVO.....	25
1.5 PROCESSO ADAPTATIVO PARA NA EVOLUÇÃO DOS COEFICIENTES: APRENDIZADO.....	26
1.6 RESULTADOS E ANÁLISES.....	28
1.7 APÊNDICE.....	28
CAPÍTULO 2. EQUAÇÕES DIFERENCIAIS: OTIMIZAÇÃO DE SOLUÇÕES	29
2.1 EQUAÇÕES DIFERENCIAIS.....	30
2.1.1 <i>Classificação das Equações Diferenciais</i>	31
2.2 VISÃO GERAL DAS QUESTÕES E PROPOSTAS DE SOLUÇÃO	32
2.2.1 <i>Critério de erro</i>	32
2.2.2 <i>Melhoria de coeficientes</i>	38
2.2.3 <i>O SAM e os PVIs</i>	38
2.2.4 <i>Preservação das condições iniciais</i>	39
2.2.5 <i>Qualidade de uma solução numérica</i>	40
2.2.6 <i>Avaliação dos coeficientes: adaptação</i>	40
2.2.7 <i>Abrangência e custo computacional</i>	41
2.2.8 <i>Margens de erro</i>	42
2.3 MÉTODOS NUMÉRICOS DE INTEGRAÇÃO PARA OS PVIS DE 1ª ORDEM	43
2.3.1 <i>Método de Runge-Kutta</i>	44
2.3.2 <i>Método de Adams</i>	45
2.3.3 <i>Comparativo dos métodos de Adams e Runge-Kutta do Matlab</i>	46
2.4 MÉTODOS NUMÉRICOS PARA PVIS DE ORDEM 2 OU SUPERIORES	48
CAPÍTULO 3. FUNDAMENTAÇÃO ANALÍTICA.....	50
3.1 TEOREMA DE WEIERSTRASS.....	50
3.2 ORTOGONALIDADE E COMPLETUDE.....	51
3.3 POLINÔMIOS DE LEGENDRE	55
3.3.1 <i>Propriedades</i>	56

3.3.2	<i>Integral de uma função expressa na base de Legendre</i>	59
3.3.3	<i>Derivada de uma função expressa na base de Legendre</i>	61
3.3.4	<i>Erro nos coeficientes</i>	63
3.3.5	<i>Mudança de base: Legendre $\times$ Canônica</i>	64
3.4	BASE ORTONORMAL DE FOURIER	66
3.4.1	<i>Derivada de uma função expressa na base de Fourier</i>	67
3.5	MATRIZES DE INTEGRAÇÃO E DIFERENCIAÇÃO (M_I E M_D)	67
3.5.1	<i>Base canônica</i>	68
3.5.2	<i>Base de Legendre</i>	71
3.5.3	<i>Base de Fourier</i>	77
3.6	RELAÇÃO ENTRE OS COEFICIENTES E AS CONDIÇÕES INICIAIS	80
3.6.1	<i>Condição inicial – base canônica</i>	80
3.6.2	<i>Condição inicial – base de Legendre</i>	81
3.7	TRANSPOSIÇÃO DE DOMÍNIOS NAS EDS	84
3.7.1	<i>Transposição Genérica</i>	84
3.7.2	<i>Transposição Linear</i>	89
3.7.3	<i>Transposição trigonométrica</i>	90
3.7.4	<i>Transposição Irracional</i>	91
3.8	INTEGRAÇÃO DE GAUSS-LEGENDRE	92
CAPÍTULO 4. ALGORITMO GENÉTICO		94
4.1	CARACTERÍSTICAS DOS ALGORITMOS GENÉTICOS	95
4.1.1	<i>Hereditariedade simulada</i>	95
4.1.2	<i>Crossing-over</i>	95
4.1.3	<i>Elitismo</i>	96
4.1.4	<i>Seleção por roleta viciada</i>	96
4.1.5	<i>Espaço de soluções e Inteligência Artificial</i>	104
4.1.6	<i>Aplicações dos AGs</i>	105
4.2	VARIEDADES DE ALGORITMOS GENÉTICOS	105
4.2.1	<i>Algoritmo Genético Diferencial (dAG)</i>	106
4.2.2	<i>Algoritmo Genético Compacto e Particle Swarm</i>	107
CAPÍTULO 5. OS PVIS E O SAM		108
5.1	PROCESSO EVOLUTIVO-ADAPTATIVO NA RESOLUÇÃO DAS EDS	109
5.1.1	<i>Dispersão dos coeficientes de Legendre</i>	109
5.2	MAJORANTES DAS MARGENS DE ERRO	118
5.2.1	<i>Majorante do erro da série em relação à função</i>	119

5.2.2	<i>Majorante do erro nas derivadas da série</i>	120
5.2.3	<i>Transposição dos majorantes de erros</i>	121
5.3	QUALIDADE DE UM MÉTODO NUMÉRICO DE INTEGRAÇÃO	122
5.3.1	<i>Dispersão nos coeficientes de uma solução numérica</i>	122
5.3.2	<i>Limites da análise</i>	128
5.4	EVOLUÇÃO DE UMA FUNÇÃO EXPANDIDA EM SÉRIE	131
5.4.1	<i>Primeira geração</i>	132
5.4.2	<i>Seleção dos melhores</i>	137
5.4.3	<i>Mutações, Crossing-over e Elitismo</i>	137
5.5	RECORRÊNCIA NA RESOLUÇÃO DE UM PVI	140
5.5.1	<i>Realimentação com a saída</i>	141
5.5.2	<i>Realimentação com a média</i>	143
5.6	APROXIMAÇÃO DE UMA FUNÇÃO	144
5.7	SOLUÇÕES PERIÓDICAS	145
CAPÍTULO 6.	RESULTADOS E ANÁLISES	146
6.1	EXEMPLOS	146
	EXEMPLO 1	146
	EXEMPLO 2	149
	EXEMPLO 3	153
	EXEMPLO 4	155
	EXEMPLO 5	158
	EXEMPLO 6	161
	EXEMPLO 7	164
	EXEMPLO 8	166
	EXEMPLO 9	169
	EXEMPLO 10	172
	EXEMPLO 11	176
6.2	CONCLUSÃO	178
6.3	TRABALHOS FUTUROS	180
6.3.1	<i>Matriz Sanduíche (Ω)</i>	181
6.4	TEOREMA DA MATRIZ SANDUÍCHE	181
6.4.1	<i>Exemplos</i>	183
APÊNDICE –	DESCRIÇÃO DO ALGORITMO E COMENTÁRIOS	187
1.1	BASE DE DADOS	187
1.1.1	<i>Método numérico</i>	189

1.1.2	<i>Método aleatório.....</i>	<i>191</i>
1.2	criar um polinômio.....	192
1.3	Retomada de um processamento.....	196
1.4	Gráficos de processamentos	197
1.5	Qualidade de um processo de integração numérica	198
1.6	Recorrência	198
1.7	Aproximação de uma função	199
1.8	Comentários.....	200
1.8.1	<i>Artifícios do código.....</i>	<i>200</i>
1.8.2	<i>Resultados simbólicos</i>	<i>201</i>
EXEMPLO 1.....		201
EXEMPLO 2.....		201
EXEMPLO 3.....		201
EXEMPLO 4.....		201
EXEMPLO 5.....		202
EXEMPLO 6.....		202
EXEMPLO 7.....		202
EXEMPLO 8.....		202
EXEMPLO 9.....		203
EXEMPLO 10.....		204
EXEMPLO 11.....		204
REFERÊNCIAS BIBLIOGRÁFICAS.....		205

Capítulo 1. Introdução

Além do âmbito da matemática pura, as equações diferenciais (*EDs*) estão presentes na descrição dos fenômenos naturais em geral — físicos, químicos, biológicos —, modelos econômicos, modelos populacionais e outras variedades [Boas, 1983, pp. 337-338].

Ao conseguir descrever um fenômeno por uma equação diferencial, tem-se em mãos um objeto matemático cuja manipulação pode permitir previsões, dimensionamentos, avaliação de comportamentos etc. [Arfken & Weber, 1995, pp. 456-462].

No entanto, a resolução das equações diferenciais, muitas vezes, esbarra em sérias dificuldades, principalmente as não lineares. Muitas não têm solução analítica descritível por uma combinação finita de funções elementares, ou nem mesmo uma série infinita com relação de recorrência determinável.

Na maioria das aplicações, o fenômeno (problema) descrito pela *ED* está sujeito a determinadas condições iniciais (*CI*). Nestes casos, o tema é conhecido como o Problema do Valor Inicial (*PVI*), o principal foco desta tese. Ou seja, encontrar uma solução para a equação diferencial que atenda às condições iniciais estabelecidas.

1.1 Objetivos

Este trabalho propõe e desenvolve um método geral de resolução aproximada por polinômios de alto grau, com margens de erro muito pequenas, para o *PVI*, incluindo as *EDs* não lineares, com as seguintes contribuições:

- Uma nova fundamentação analítica matricial para a resolução dos *PVIs*;
- Um *software* que se utiliza dessa nova fundamentação analítica matricial e do algoritmo genético diferencial, descrito na seção 4.2, para aprimorar as soluções e confinar mais ainda as margens de erro. Por brevidade, esse software será chamado simplesmente de *SAM*¹. No *SAM*, desenvolvido em Matlab 7, toda fundamentação analítica deduzida nesta tese juntamente com outras consagradas já são implementadas de forma que o usuário não precise delas ter conhecimento se quiser aplicá-lo a um *PVI* qualquer. Automaticamente, o software vai gerar os gráficos comparativos, as

¹ Um semi-acrônimo para “software de automação na aplicação do algoritmo genético diferencial modificado para os problemas do valor inicial” necessário, diante da quantidade de vezes que o termo irá comparecer nesta tese.

expressões analíticas do polinômio de aproximação e suas derivadas, bem como os majorantes do erro em cada um desses elementos.

Como benefícios indiretos decorrentes das contribuições anteriores foram obtidos também:

- Um método genérico e analítico para transposição de domínios no *PVI*, seja por transformação **linear ou não**, determinando o impacto da transposição na função, derivadas, nas *CI* e na *ED*.
- Um critério objetivo para se analisar a qualidade de soluções numéricas de *PVIs*, independentemente do conhecimento do processo utilizado na obtenção do conjunto de pontos que compõe a solução numérica.
- O cálculo dos majorantes dos erros nas funções obtidas e suas derivadas, ao longo de todo o domínio.
- Aproximações polinomiais de funções com coeficientes descritos pela razão entre dois inteiros (precisão ilimitada nas casas decimais), com polinômios de graus relativamente altos (até grau 26, ou mais, dependendo da limitação do processador), com majorante de erro determinado e extremamente baixo.
- Um método analítico direto para obter diretamente (sem o processo evolutivo) soluções polinomiais de alto grau como aproximação da função e suas derivadas, com baixa margem de erro, para os *PVIs lineares*.
- Um critério de erro que minimiza, nas aproximações por séries — polinomiais, trigonométricas ou outra qualquer —, tendo por base o critério dos mínimos quadrados, as aberrações que ocorrem nos extremos do domínio, permitindo aumentar a confiança na aproximação ao longo de todo o intervalo em que a função é definida, utilizando computação evolutiva.
- Uma ferramenta para expansão polinomial de funções com polinômios de alto grau com margens de erro inferiores a 10^{-12} .

Todas essas tarefas adicionais também são feitas automaticamente pelo *SAM*, conforme a escolha do usuário.

1.2 Motivação

Soluções polinomiais para os *PVIs* já foram propostas e implementadas [Sezer & Kesan, 2000], [Kesan, 2003], [Babolian & Fattahzadeh, 2007], mas com margens de erro bem maiores do que as obtidas neste trabalho por razões que serão estudadas e discutidas na secção 5.2.2.

Babolian & Fattahzadeh [2007] discutem, deduzem e aplicam uma matriz de diferenciação usando como base os polinômios de Chebyshev — base ortogonal na qual os pontos do intervalo não têm todos o mesmo peso — na resolução de *PVIs*. Nesta tese, a escola de Babolian & Fattahzadeh foi preterida com a intenção de dar o mesmo peso a todos os pontos do domínio, razão pela qual se escolheu trabalhar com os polinômios de Legendre. O exemplo 1 da secção 6.1 compara os resultados obtidos por esses pesquisadores com os desta tese.

O capítulo 2 apresenta uma visão geral sobre o *PVI*, sobre classificação das equações diferenciais e um pouco do histórico do problema.

Em vista da dificuldade, ou até mesmo impossibilidade, da obtenção de soluções analíticas em muitos *PVIs*, vários métodos numéricos para resolução foram propostos. Atualmente, a maioria desses métodos já é função residente em softwares de computação numérica-simbólica, como o Matlab, por exemplo. Em geral, eles são extremamente precisos e rápidos, mas cada um, dependendo da equação a que será aplicado, tem suas limitações. A secção 2.3 apresenta alguns de maior relevância em relação a este trabalho.

Em várias classes de *PVIs*, mesmo **lineares**, os *MN* não conseguem sair do ponto inicial e prosseguir no cálculo das funções e derivadas para as abscissas seguintes do intervalo, como por exemplo, na equação de Bessel de ordem zero: $y'' \cdot x + y' + xy = 0$, $x \in [0, 1]$, tal que $y(0) = 1$ e $y'(0) = 0$.

Nos casos analisados, o *SAM* foi capaz determinar uma solução polinomial com excelente aproximação, ou, mesmo obter a solução exata, nos casos em que ela era polinomial.

Ainda no capítulo 2, ao longo da secção 2.2, apresentam-se as 8 questões fulcrais que nortearam esta tese, delineando onde e como serão respondidas. Os temas dessas questões são:

- critério de erro que leve em conta tanto a otimização global como as locais;
- evolução da série que aproxima a função;
- preservação das condições iniciais durante a evolução;
- análise da qualidade de uma solução numérica por um critério externo a ela, ou seja, que não depende do conhecimento do algoritmo interno que gera o conjunto de pontos da solução;
- abrangência e custo computacional;
- majorantes para as margens de erro da função e suas derivadas.

Todos esses temas se relacionam com o tema maior — a otimização de soluções aproximadas dos *PVIs*, que, em alguns casos, convergiu para a solução exata.

1.3 Fundamentação analítica desenvolvida

O capítulo 3 expõe a fundamentação analítica que alicerça todo o processo computacional arquitetado neste trabalho, começando com a apresentação do Teorema de Weierstrass (secção 3.1) — que garante a aplicabilidade de séries de Legendre na aproximação de funções —, e deduzindo as matrizes de integração e diferenciação na base de Legendre que serão os instrumentos principais na computação das soluções.

A secção 3.2 discute a importância da ortogonalidade da base escolhida para expansão em série da função e também da completude, assegurando que não faltam “azulejos”, por menor que sejam, para preencher integralmente o gráfico da função, desde que ela seja de quadrado integrável. A razão dessa última exigência é explicitada detalhadamente no tópico 3.3.1.

A secção 3.3 se ocupa da discussão das séries de Legendre. Nessa secção, duas propriedades importantes — pelo menos para o uso nesta pesquisa — foram deduzidas por esta tese: a integral e a diferencial de um polinômio genérico de Legendre, expressas na própria base de Legendre.

A obtenção dessas propriedades permitiu a dedução das matrizes de integração (M_{LI}) e de diferenciação (M_{LD}), pelas quais as derivadas puderam ser comparadas com a

função e, inclusive, determinar qual é a margem de erro que cada derivada herda da função de aproximação, em cada coeficiente.

O êxito na obtenção dessas matrizes motivou a busca de um mesmo processo para outras bases, eventualmente aplicáveis em outras pesquisas, como a base canônica e a de Fourier, matrizes essas discutidas nas secções 3.5.

Inicialmente, trabalhava-se diretamente a partir da melhor série de Legendre para determinado conjunto de pontos. Pontos obtidos, por exemplo, por meio de um método numérico consagrado, como o de Runge-Kutta, aprimorado por Dormand & Prince [1980] ou o de Adams [Hull et al., 1972]. Em geral, esse melhor polinômio não satisfazia rigorosamente as condições iniciais.

Ora, sabe-se que a mudança de condições iniciais acarreta uma mudança na solução obtida. Este fato levou esta pesquisa a deduzir (determinar) — analiticamente — que condições deveriam ser impostas aos coeficientes de forma que as condições iniciais ficassem sempre rigorosamente mantidas. Esses resultados são apresentados pela secção 3.6.

Na base de Legendre, a manutenção das CI é garantida pela solução de um sistema linear possível e indeterminado. Sendo q a ordem da equação diferencial e n o número de coeficientes, obtém-se um sistema com q equações e n incógnitas. Pode-se então escolher-se, arbitrariamente, q coeficientes para serem expressos em função dos $n - q$ restantes. Essa relação vai impor, rigorosamente, que as condições iniciais sejam mantidas. Os q coeficientes escolhidos podem ser re-escolhidos ao longo do processo de evolução com a intenção de permitir que todos sejam mutáveis, e não apenas um reflexo dos outros $n - q$ elementos.

Novamente, o êxito nessa obtenção induziu esta pesquisa a determinar como seria essa manutenção das CI em outras bases, como a Canônica, por exemplo, cujos resultados são apresentados na secção 3.6.

O processo de integração indefinida de uma função nos leva a outra função a menos de uma constante arbitrária. Porém, se houver uma condição a ser obedecida pela integral da função, essa constante deixa de ser arbitrária e pode ser determinada. A secção 3.6.2 mostra como isso poder ser feito quando se usa a matriz de integração de Legendre.

Outro aspecto que carece de fundamentação analítica é a questão dos domínios das funções.

As bases ortogonais completas são de domínios bem definidos. Caso a função procurada tenha um domínio diferente, se faz necessária então uma mudança de variável de

forma a levar a função do domínio original para um domínio de trabalho, que varia conforme a base ortogonal que se escolhe.

A mudança de variável acarreta mudanças na equação e também nas condições iniciais. Essas transposições de domínios e o impacto que elas têm nas derivadas da função tornaram necessária a determinação analítica de como isso deveria ser feito e como a transposição afetaria a própria equação, o que é feito na secção 3.7.1.

Em vista da possibilidade de várias propostas para transposição de domínios, considerou-se, nesta tese, uma transposição genérica, por uma função bijetora, para, a seguir, analisarem-se três transformações propostas: a linear, a trigonométrica e a irracional. Todos esses resultados e deduções são apresentados na secção 3.7, considerando inclusive o impacto da transposição de domínios nas condições iniciais. Enfatizando, o *SAM* realiza todas essas tarefas automaticamente para *PVIs* de até quarta ordem.

Por fim, o capítulo 3 descreve o processo de quadratura de Gauss-Legendre [Campos, 2007, p. 229] — de longe, o mais preciso dos processos de integração numérica [Keiner & Potts, 2008] — que foi usado ao longo de todo este trabalho. Tal é a acurácia e a relativa rapidez desse processo de integração numérica, comparada a outros processos [Campos, *ib.*, p.245], que tem sido ele o escolhido em vários cálculos de precisão [Duan, 1997].

1.4 Escolha do processo evolutivo

Em 1975, J. H. Holland apresentou uma proposta para resolver problemas de otimização baseada em hereditariedade por recombinação genética, seleção dos melhores e mutações [Holland, 1975], aplicados a determinada população. Basicamente, é um processo heurístico-estocástico inspirado no Darwinismo e na genética, denominado por Holland Algoritmo Genético (*AG*). As mutações são uma parte estocástica do algoritmo.

Desde então, variedades desse processo têm sido propostas e aprimoradas, como, por exemplo, o Algoritmo Genético compacto, (*cGA*) [Rastegar & Hariri, 2006], o *AG* diferencial (*dGA*) [Storn, 1996] e outras, cujas comparações de desempenho foram analisadas em alguns casos [Hrstka et al., 2003] e serão discutidas na 2ª secção do capítulo 4, que discorre sobre alguns tipos de *AGs*.

Esta tese tem suas origens no Núcleo de Ciências Cognitivas da USP, que se encontra hospedado no Departamento de Engenharia de Sistemas Eletrônicos da Escola

Politécnica, sendo também um grupo de pesquisa deste departamento e vinculado ao Laboratório de Sistemas Integráveis. Um grupo que pelo carácter multidisciplinar convidou o trabalho a inserir elementos da Biologia na abordagem de problemas formais, aproveitando a idéia seminal de Holland [1975].

Com o acervo até aqui descrito, chega-se ao capítulo 5, mas não ainda com todas as armas para “enfrentar” os *PVIs* em geral.

1.5 Processo adaptativo para na evolução dos coeficientes: aprendizado

Que capacidade tem um candidato à solução de um *PVI* de aprender?

Bem, se fosse conhecida a solução exata, o aprendizado seria como a velha brincadeira do “está quente, está frio”, pondo o candidato a tatear cegamente o espaço de soluções. Mas, a aplicabilidade do *SAM* deve ser para problemas cuja solução não é conhecida, sendo aqueles de solução conhecida usados apenas na fase de teste. Além disso, como os coeficientes de uma série são variáveis contínuas, o espaço de soluções tem infinitos elementos.

Há, contudo, outro processo de aprendizado que pode nos nortear na busca da melhor proposta, qual seja, analisar o resultado da ação de determinado candidato.

A secção 5.1 mostra, acompanhada de um exemplo, como, ao testar um candidato para solução, podemos obter o perfil da última derivada da função que seria a solução. A diferença entre esse perfil e o perfil do candidato — resultado da ação — é um processo reflexivo, numa óptica em que, se ele é o melhor candidato, o perfil obtido após a passagem pelo filtro da equação, integrado pela matriz de integração, nos fornece de volta ele mesmo. Se não for o próprio, haverá uma “dispersão”, um erro. Descobre-se assim o quão ruim é o candidato, mesmo sem saber quem é o candidato ideal.

Na secção 5.1.1 mostra-se como a dispersão nos coeficientes pode confinar a faixa de variação dos valores dos coeficientes (denominada aqui “envelope”) que querem evoluir, de forma que o espaço de soluções se torne cada vez mais restrito.

Um produto inesperado, por assim dizer, dessa idéia foi a possibilidade de com ela analisar soluções numéricas obtidas por processos quaisquer, analisando-se a dispersão dos coeficientes que os pontos dessa solução acarretam. Com essa análise foi possível determinar os majorantes de erro da solução numérica, e, a partir da dispersão dos

coeficientes, determinar a sua qualidade, tanto no aspecto da coerência entre a função e as derivadas que o método apresenta (dispersão interna), quanto pelo aspecto da diferença entre o melhor candidato que ela apresenta e o fruto de sua substituição na ED (dispersão externa).

Além disso, a dispersão permitiu obter os majorantes para o erro do processo, no intervalo analisado. Tudo isso é descrito na secção 5.2.

Uma vez que a integração de Gauss-Legendre [Mughal, Ye & Iqbal, 2006] precisa dos valores das funções em abscissas muito bem determinadas e o método numérico a ser analisado possa ter utilizado outras abscissas, usam-se os *splines* cúbicos para fazer uma adequação das abscissas. Como descrito nessa secção, se o conjunto de pontos for suficientemente denso, essa adequação pouco interfere nas margens de erro a serem calculadas.

Após a evolução, chega a hora de se utilizar o polinômio que aproxima a função em situações práticas. Qual é a margem de erro que essa aproximação acarreta?

A margem de erro que se obtém ao substituir a aproximação na equação diferencial (resíduo da substituição - *RS*) **não** é a margem de erro da função. A margem de erro da função será obtida e discutida, na secção 5.2, pelos valores da dispersão externa dos coeficientes, concebendo uma situação extremamente conservadora — o pior dos cenários — na qual todas as dispersões individuais errariam no mesmo sentido, a maior, por exemplo, em todos os pontos da função. Mesmo assim, como ao final da evolução as dispersões são extremamente pequenas, esse majorante do erro resultou muito pequeno, em geral, menor do que 10^{-6} .

Em seguida, ainda na secção 5.2, com a aplicação da matriz de diferenciação, foram obtidos os majorantes do erro nas derivadas.

Diante de margens calculadas tão pequenas, era inevitável a confrontação desses cálculos com os erros propriamente ditos. Isso foi feito nos casos em que se conhecia a solução analítica e, de fato, o cálculo se mostrou extremamente conservador, isto é, todos os erros propriamente ditos ficaram confinados à margem calculada e, em geral, eram bem menores (cerca de 1/10), tanto para a função como para as derivadas.

O processo pelo qual se calcula a dispersão externa, deduzido nesta tese, (secção 5.1) nos fornece, para um dado vetor de coeficientes de entrada $\vec{t}_1$ um vetor de saída $\vec{t}_2$. Se eles forem iguais, $\vec{t}$ é solução exata do *PVI*. Mas se não forem, realimenta-se o processo com $\vec{t}_2$, buscando um $\vec{t}_3$, e assim sucessivamente até que a entrada seja praticamente igual à saída. Isso seria um processo de busca de solução, semelhante ao usado no mapa logístico [Arfken

& Weber, 1995, pp. 993-996], e se mostrou funcional em grande parte dos *PVIs*, convergindo rapidamente, com cerca de 30 iterações (em torno de 5 s), mas não em todos os casos.

Eventualmente, $\vec{t}_1$ e $\vec{t}_2$ se alternavam na entrada e saída, deixando o processo de busca em um círculo vicioso. Para evitar essa alternância, a seção 5.4.2 propõe a realimentação com a média entre os vetores. A proposta teve êxito nas situações em que havia alternância, mas não se mostrou funcional nos casos em que a recorrência divergia.

Como proceder nesses casos? Evoluir o polinômio com o *SAM*. Esse método foi o que se mostrou geral.

Curiosamente, mesmo após a recorrência ter obtido êxito, foi possível evoluir ainda mais o polinômio com o *SAM*. Como?

A recorrência minimiza a dispersão externa, levando as margens de erro da função e derivadas quase para os limites da precisão da integral numérica de Gauss-Legendre dentro do critério dos *MQ*, enquanto o *SAM* utiliza o critério de erro *LG*; ou seja, de forma aparentemente paradoxal, é possível encontrar um polinômio de aproximação melhor, segundo o critério *LG*, com pequeno sacrifício na dispersão.

1.6 Resultados e Análises

O capítulo 6 – Resultados e Análises — mostra os resultados de alguns processamentos feitos e apresenta os comentários finais, comparando ou com a solução analítica (quando expressável por combinação finita de funções elementares), ou com os *MNs*.

1.7 Apêndice

O apêndice é praticamente um manual para a utilização do *SAM*, descrevendo suas telas de entrada e saída, além de suas funcionalidades.

Alguns dos resultados das séries obtidas também fazem parte do apêndice, uma vez que, pelo fato de os coeficientes serem expressos pela razão entre dois inteiros grandes, eles ocupariam páginas inteiras no corpo da tese.

Capítulo 2. Equações Diferenciais: Otimização de Soluções

Henry Poincaré (1854-1912), considerado o último dos renascentistas pelo seu brilho em Filosofia, Matemática, Física e Retórica, após suas expressivas contribuições na Teoria da Relatividade [Granek, 2000], [Cuvaj, 1968], questionava na conferência de Solvay, em novembro de 1911, se os recém-descobertos fenômenos quânticos poderiam ser descritos por equações diferenciais.

“As novas investigações aqui discutidas não somente parecem questões dos princípios fundamentais da Mecânica, mas também sacodem violentamente uma concepção que até agora estava completamente atrelada ao conceito de lei natural. Podemos ainda expressar essas leis na forma de equações diferenciais?” [Mehra & Reichenberg, 1982]

Havia um profundo significado nessa pergunta, qual seja, se a matemática seria capaz de descrever o imprevisível. O contexto histórico aqui é de extrema importância e pode ser aquilatado pela descrição biográfica que C. Watson faz de William Thomson – Lord Kelvin (1824-1907), um dos mais destacados físicos daquela época.

“Durante a primeira metade de sua carreira ele parecia um ser incapaz de estar errado, enquanto na segunda metade parecia incapaz de estar certo.” [apud Harré, 1969]

Essa posição extremada de Watson se prende à recusa de Kelvin em aceitar a natureza granular da matéria (átomos), seus cálculos incorretos da idade da Terra e do Sol, que conflitavam a Teoria da Evolução de Darwin e sua oposição às idéias de Rutherford sobre radioatividade. Em suma, a oposição de Kelvin estava centrada contra os fenômenos quânticos — a imprevisibilidade.

Kelvin estava imerso no determinismo de Laplace (1749-1827), que preconizava:

“Nós podemos tomar o estado presente do universo como o efeito do seu passado e a causa do seu futuro. Um intelecto que, em dado momento, conhecesse todas as forças que dirigem a natureza e todas as posições de todos os itens dos quais a natureza é composta, se este intelecto também fosse vasto o suficiente para analisar essas informações, compreenderia numa única fórmula os movimentos dos maiores corpos do universo e os do menor átomo; para tal intelecto nada seria incerto e o futuro, assim como o passado, seria presente perante seus olhos.” [Hahn, 2005]

Um mês após a conferência de Solvay, seis meses antes de seu falecimento, Poincaré apresentou na Academia de Ciências de Paris a demonstração da resposta à sua pergunta: sim, os fenômenos quânticos também podiam ser descritos por equações diferenciais.

Ainda no terreno da previsibilidade $\times$ imprevisibilidade, Poincaré havia sido premiado, em 1889, por mostrar que a solução do problema de três corpos continha, conforme a relação entre as massas, elementos caóticos [Mehra & Rechenberg, 1982], isto é, mesmo na Física Clássica havia problemas na previsibilidade.

Dessa forma, seja no previsível ou não, Poincaré demonstrou que as equações diferenciais poderiam também descrever os fenômenos naturais da emergente Física Moderna.

Da demonstração de Poincaré, quinze anos se passaram até que Erwin Schrödinger, em 1926, publicasse “Quantisierung als Eigenwertproblem” no *Annalen der Physika*, a célebre equação diferencial da mecânica ondulatória [Mehra & Rechenberg, *ib.*].

Esse ligeiro apanhado histórico visa situar a importância do tema: as equações diferenciais.

Quando para determinada equação diferencial são dadas as condições iniciais (*CI*), a situação é denominada Problema do Valor Inicial (*PVI*). O nome especial se justifica, pois, em vários casos, na impossibilidade de se determinar a solução analítica, as condições iniciais “dão a partida” em métodos numéricos de resolução.

Este trabalho sobre otimização de soluções do *PVI* vai utilizar como suporte o algoritmo genético diferencial (*dAG*) [Krink, Thiemo & Paterlini, 2006], [Hrstka et al., 2003], [Hrstka & Kucerovala, 2004], com algumas modificações, cuja breve descrição pode ser encontrada na secção 2.2.3 deste capítulo; e também os polinômios de Legendre (secção 3), com algumas propriedades, identidades e ferramentas matriciais aqui deduzidas, juntamente com as já consagradas, tendo como plataforma para a programação o Matlab 7 ®.

As descrições mais detalhadas desses dois últimos suportes são apresentadas no capítulo 3.

2.1 Equações diferenciais

Uma equação diferencial (*ED*) é uma equação matemática que relaciona uma função desconhecida, de uma ou mais variáveis, com as variáveis e as derivadas dessa função.

Remotamente, até o começo do século XX, as aplicações das *EDs* eram quase que exclusivas da Física e da Química — uma parte das ciências naturais.

Sucintamente, as equações diferenciais relacionam grandezas com taxas de variação dessa grandeza e outras funções. Dessa forma, construir a *ED* que governa um fenômeno, seja biológico, econômico ou de outra área qualquer, é muitas vezes relativamente simples. Obtém-se então um modo de descrever o fenômeno de forma compacta e elegante.

Como decorrências têm-se hoje as *EDs* presentes em vários ramos das ciências e tecnologia, como a Economia, Biologia, Física, Engenharia etc.

A relativa facilidade em construir a *ED* para determinado fenômeno, freqüentemente, rivaliza com a dificuldade em resolvê-la. De fato, algumas equações atravessaram séculos à espera de uma solução analítica, algumas permanecem aguardando. Além disso, há também vários casos em que a equação não tem solução analítica descritível por uma combinação de funções elementares. Em alguns desses casos, a solução exata é uma série infinita de potências com uma relação de recursividade entre os termos, mas em vários outros casos dessa categoria, nem mesmo esse recurso é aplicável.

Diante dessas dificuldades apresentadas, vários métodos numéricos para integração das *EDs* foram propostos, sujeitas a determinadas condições iniciais (*PVI*), entre os quais, os de maior notoriedade e aplicação talvez sejam o elaborado pelos matemáticos alemães Carl David Tolmé Runge (1856-1927) e Martin Wilhelm Kutta (1867 –1944) [O'Connor, 2008], no começo do século XX, aprimorado por Dormand & Prince [1980], e o preditor-corretor de Adams [Hull et al., 1972].

A última secção deste capítulo apresenta uma descrição de métodos numéricos para resolução das *EDs* sujeitas a determinadas condições iniciais que serão usados neste trabalho, as condições em que podem ser aplicados, vantagens e desvantagens, assim como os limites de precisão que se obtém em cada um.

2.1.1 Classificação das Equações Diferenciais

Diante da extensão de tipos de *EDs*, elas são classificadas conforme vários critérios. A seguir, apresentam-se alguns critérios acompanhados de exemplos. O tema é está largamente consolidado na literatura [Arfken & Weber, 1995, p. 457-478]. Neste trabalho consideram-se somente as *EDs* de variáveis reais.

Um dos critérios para a classificação das *EDs* é a quantidade de variáveis independentes, conforme a tabela seguinte. De acordo com esse critério, as *EDs* são:

Tabela 2.1. Classificação das *EDs* de acordo com a quantidade de variáveis independentes.

Ordinárias Uma variável independente	Parciais Duas, ou mais, variáveis independentes
$y''' = x^2 + \frac{x}{\sqrt{y}}$	$\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2}$

Outro critério é a linearidade. Se a equação for uma combinação linear da função e suas derivadas, ela é linear. Além disso, para as equações lineares existe o critério da homogeneidade. Se a combinação linear da função e suas derivadas for nula, a *ED* linear é homogênea, conforme ilustra a tabela seguinte.

Tabela 2.2. Linearidade nas *EDs*.

Linear	Não linear
$a_0 y + a_1 y' + a_2 y'' = b$ $b = 0 \Rightarrow$ homogênea $b \neq 0 \Rightarrow$ não-homogênea Os coeficientes a_0 , a_1 e a_2 podem depender de x (mas não de y).	$y''' + y \cdot y'' = 0$

Ordem da equação diferencial

Esta análise supõe funções contínuas com derivadas contínuas.

Por definição, a ordem de uma equação diferencial é a ordem da derivada de maior ordem presente na equação.

Nas descrições dos fenômenos físicos em geral, raramente têm-se equações diferenciais de ordem maior do que dois. Contudo, com o intuito de ampliar a abrangência, o *SAM* foi desenvolvido de forma a trabalhar com equações diferenciais até 4ª ordem.

2.2 Visão geral das questões e propostas de solução

Esta secção apresenta algumas questões que motivaram este trabalho e uma breve visão das resoluções propostas, cujos detalhamentos permearão os capítulos seguintes. Procura-se aqui apresentar essas questões em ordem crescente de importância, conforme opinião do autor, mas, sem dúvida, essa ordem admite certa subjetividade.

2.2.1 Critério de erro

Comumente, utiliza-se como norma para minimizar o erro em uma otimização o critério dos mínimos quadrados (*MQ*).

Se o objetivo for encontrar o melhor polinômio de grau n que representa um conjunto de pontos em determinado intervalo, ou ainda determinada função ($f(x) = \exp(-x^2)$),

por exemplo, a aplicação dos polinômios de Legendre nos leva diretamente à resposta, pois eles satisfazem esse critério — conforme demonstrado na secção 3 do capítulo 3 e incorporado à literatura [Boas, 1983, p. 504].

A título de exemplo, considera-se a função que satisfaça a equação diferencial: $y' + 2xe^{-x^2} = 0$, no intervalo $[-1,1]$, cuja solução analítica é $y = e^{-x^2}$ quando se impõe $y(-1) = e^{-1}$.

Aproximando-se a função por um polinômio de grau 8, utilizando os polinômios de Legendre, obtém-se, com quatro decimais:

$$y = 0,0263x^8 - 0,1551x^6 + 0,4963x^4 - 0,9996x^2 + 1,0000$$

A substituição da solução exata no *PVI* vai resultar na função identicamente nula ao longo de todo o intervalo. Entretanto, substituindo-se o polinômio de aproximação na equação diferencial, ter-se-á um resíduo não nulo (*RS*), decorrente da substituição da função aproximada, cujo valor varia ao longo do intervalo.

Gráfico 2. 1. Resíduo (*RS*) ao substituir o polinômio de aproximação na *ED*.

A análise do gráfico anterior — resíduo ao longo do intervalo — nos mostra que o critério dos *MQ* determina a função que apresenta a melhor concordância ao longo de todo intervalo, mas penaliza os extremos. Mas, por que nos extremos?

Embora nos extremos o erro seja muito alto, como eles ocorrem em intervalos muito pequenos, a integração resulta em valores pequenos, obedecendo ao critério. Grosso modo, o método dos mínimos quadrados procura “empurrar” o erro para fora do intervalo.

Seria possível obter um polinômio “melhor” por algum método computacional evolutivo? Um polinômio que não divergisse nos extremos e tivesse o erro bem distribuído ao longo de todo intervalo, respeitando um critério de erro mínimo? Seria possível obter um critério que conciliasse a procura de mínimos para erros locais com a procura de mínimos para erros globais?

Resposta

O critério *MQ* faz uma otimização global no domínio em que é aplicado, mas sacrifica os extremos, ou seja, admite erros locais expressivos. D. D. Shepard, em 1968, introduziu o conceito dos mínimos quadrados dinâmicos² (*MLS*), considerando as derivadas de primeira ordem nos problemas envolvendo equações diferenciais parciais [Zuppa, 2003], conceito mais tarde generalizado para ordens superiores³ por Nayroles, Belytschko et al. [Zuppa, *ib.*], considerando o espaço de Sobolev.

Armentano [2001] apresentou um apurado estudo dessa questão, obtendo majorantes de erro máximo, até segunda ordem, na função e derivadas; e Lim, Im & Cho mostraram a aplicação do *MLS* [2007], nos métodos de elementos finitos.

No entanto, neste trabalho, o objeto de minimização de erro é o resíduo do *PVI* no qual se tem: $E(x, y, y^{(1)}, \dots, y^{(q)}) = 0$, onde E é a equação diferencial, e não a função $y(x)$ em si. Diante disto, esta tese considerou três possíveis abordagens, brevemente descritas a seguir, optando pela última delas.

Uma primeira abordagem seria considerar a função E e a sua derivada, que é a função identicamente nula, e isso, naturalmente, não é aplicável.

Uma segunda abordagem possível seria considerar o resíduo obtido ao substituir a série aproximadora no *PVI*, obter o resíduo após a substituição e, finalmente, considerando a derivada desse resíduo, aplicar os mínimos quadrados dinâmicos. Esse processo iria desaguar

² Uma tradução livre de *moving least square*. Shepard, D.D. (1968). A two dimensional interpolation function for irregularly spaced data, in: **Proc. 2rd Nat. Conf. ACM**, ACM Press, NY. *Apud* Zuppa [2003].

³ Touzot, G., Villon, P. & Nayroles, B. (1991). La méthode des éléments diffus, C.R.Acad. Sci. Paris, Sér.II -313, p.133–138, 1991. *Apud* Zuppa [2003].

em um sistema de equações que poderia ser mal-condicionado, ou mesmo de difícil solução, quando não impossível, nos casos não lineares. Além disso, o resíduo genérico é obtido inicialmente pelas ferramentas simbólicas do Matlab e, na maioria dos casos, é uma expressão longa, de computação prolongada, que ficaria quase proibitiva com a inclusão da derivada, principalmente nas aproximações por polinômios de graus muito altos. Mas, de qualquer forma, isso foi deixado para análises futuras, uma vez que o critério mais simples proposto nos parágrafos seguintes apresentou resultados satisfatórios, usando computação evolutiva.

Além disso, mesmo pensando somente no *MLS*, um método adaptativo para minimização de erros foi proposto por Lee et al. [2007], mostrando que o processo é passível de evolução.

Em geral, nos *AGs*, cada geração passa por uma fase denominada **torneio** [Gibbs, Dandy & Maier, 2008] na qual se estabelece um *ranking* para a população, de acordo com determinado critério. Essa fase é de extrema importância, pois o *ranking* do candidato vai determinar a sua probabilidade de acasalamento, sua preservação — se houver elitismo — ou mesmo a sua exclusão, o que é feito com determinada faixa dos piores candidatos.

Para justificar a escolha da norma desta tese, considerem-se os gráficos que apresentam o resíduo decorrente da substituição de 4 candidatos na *ED* do *PVI*.

Gráfico 2. 2. Erros locais (resíduo) de diferentes hipotéticos candidatos à melhor solução.

As normas de erro L^2 e L^∞ têm o sentido usual [Gibbs, Dandy & Maier, 2008], respectivamente, dos mínimos quadrados e do módulo do erro máximo local ao longo de todo domínio.

Se o critério de erro fosse L^2 , o melhor candidato seria o (I), com o (IV) em segundo, seguido por (III) e (II). Entretanto, nos *PVIs*, o extremo inferior do domínio é a abscissa das condições iniciais — o ponto de partida dos métodos numéricos ou analíticos para a resolução. Sendo assim, erro local significativo nessa abscissa compromete a solução. Além disso, o critério não torna possível confinar a margem de erro do objeto de análise, já que o erro local (L^∞) pode ser extremamente alto nos extremos do domínio.

Por outro lado, se o critério for L^∞ , o candidato (II) estaria empatado, em primeiro lugar, com (III), mas vê-se claramente que ele tem L^2 maior que todos os outros, de modo que usando o critério L^2 para desempate, o indivíduo III seria o melhor. Mas e se o erro máximo do candidato (II) fosse ligeiramente maior que o do (III), mantendo o baixo resíduo ao longo do restante do domínio?

Uma forma que se propõe para levar em conta simultaneamente os dois critérios de erro é minimizar o produto $L^\infty \cdot \sqrt{L^2}$ — local $\times$ global (*LG*) —, de forma que os dois fatores pesem conjuntamente na montagem do *ranking*.

Com esse critério, o melhor candidato é o (IV). O candidato (III) é melhor do que o (II) e, dependendo do valor do resíduo que o candidato (I) apresenta nos extremos, ele poderia ser o pior.

Com a norma *LG* monta-se o *ranking* dos polinômios candidatos a resolver as *EDs* no processo evolutivo. Dessa forma, candidatos com erro global diminuto, mas erros locais expressivos perderão posições no *ranking* para outros mais regulares, eventualmente, com erro global um pouco maior. Os resultados apresentados no capítulo 6 mostram a efetividade da aplicação dessa norma.

A figura seguinte apresenta o polinômio solução “evoluído” (4000 gerações) da equação apresentada na secção anterior, $y' + 2xe^{-x^2} = 0$, no intervalo $[-1,1]$, comparado com o candidato inicial, obtido pela aplicação dos polinômios de Legendre.

Gráfico 2.3. Comparativo de critérios de erro.

A tabela seguinte compara os coeficientes das séries de cada candidato.

Tabela 2.3. Comparação dos coeficientes dos polinômios. No processo evolutivo-adaptativo, alguns aumentaram, outros diminuiram de forma a satisfazer o critério de erro LG.

	x^8	x^6	x^4	x^2	x^0
Inicial	0,0263	-0,1551	0,4963	-0,9996	1,0000
Evoluído	0,0242	-0,1511	0,4939	-0,9991	1,0000

O erro quadrático ao longo do intervalo do polinômio inicial, em relação à ED, é $e_{\text{ini}} = 8,4 \cdot 10^{-8}$.

Após a evolução, a integral do erro quadrático é $e_{\text{evo}} = 3,5 \cdot 10^{-8}$.

Cabe aqui um esclarecimento: se o desenvolvimento da função em polinômios de Legendre obedece ao critério dos mínimos quadrados, como a evolução pôde chegar a um polinômio com erro quadrático menor?

A resposta é que o polinômio procurado pela equação diferencial deve ser adequado à função e também à sua derivada, conforme a relação que esses elementos guardam entre si na equação diferencial, ou seja, além da mudança no critério de erro, de MQ para LG , houve também uma mudança do objeto avaliado.

Esta pesquisa constatou a redução na integral do erro quadrático em todos os casos analisados.

2.2.2 Melhoria de coeficientes

Outra pergunta ainda a respeito do melhor polinômio se refere aos coeficientes. Com um novo critério, não seria possível incorporar pequenas diferenças aos coeficientes menores de modo a compensar o truncamento da série?

Resposta

A secção 3.3 mostra que, tendo como critério os MQ , essas diferenças serão nulas, ou seja, em se tratando de polinômios, os de Legendre já são a melhor escolha⁴. Entretanto, em um processo computacional, há truncamento nos resultados das operações aritméticas. Nesse cenário, um método computacional evolutivo pode buscar essa pequena compensação, como o processamento demonstrou. O exemplo apresentado na primeira pergunta deu sinais dessa possibilidade. Como a função procurada é par, os coeficientes de Legendre ímpares são nulos. Entretanto, o processo evolutivo encontrou uma solução com valores extremamente pequenos para os coeficientes ímpares que “melhoraram” a solução. De fato, o *ranking* nos processos evolutivos depende da limitação, do critério seletivo, e, em um processo computacional, o truncamento das operações aritméticas é também uma limitação presente.

2.2.3 O SAM e os PVI

O título deste tópico é também o título do capítulo 4, que vai discutir com mais detalhes as aplicações e características desse tipo particular de computação evolutiva — o algoritmo genético diferencial constrito.

Ao perguntar pelo “melhor” polinômio na forma de um *PVI*, em vez de perguntar pelo melhor polinômio que preenche determinado conjunto de pontos, há de fato uma

⁴ Considerando-se ainda que função peso seja unitária.

mudança de critério. Agora o melhor polinômio é o mais adequado ao *PVI* e não à função simplesmente. Para uma série infinita isso não faria diferença, mas para uma série limitada, com a imposição das condições iniciais, há uma mudança de cenário. Os algoritmos genéticos podem enfrentar esse problema? Em princípio, os *AGs* poderiam enfrentar qualquer problema, o que não significa resolver, mas uma pergunta essencial que este trabalho procurou responder foi:

— Há resultados que corroboram a hipótese?

Resposta

O capítulo 6 vai apresentar vários resultados que mostraram ser aplicável o algoritmo genético diferencial (*dAG*), com pequenas modificações, no aprimoramento de polinômios solução do *PVI*, forma adotada pelo *SAM*. Grosso modo, a população no algoritmo genético diferencial é construída de forma que haja pequenas diferenças percentuais, a maior e a menor, em torno de um elemento central de coeficientes que seja o melhor da geração, para cada gene.

Em 2006, Charles Karr, Igor Yakushin e Keith Nicolosi apresentaram uma proposta de relativo sucesso para três equações em particular [Karr et al., 2000], abordando problemas de transmissão do calor, equação da onda e o laplaciano de uma distribuição de cargas, que o capítulo 6 discute com mais detalhes.

2.2.4 Preservação das condições iniciais

Em um processo evolutivo para um conjunto de coeficientes que visam resolver um *PVI*, seja para uma base de polinômios, série trigonométrica, ou outra base completa qualquer, os melhores ajustes no intervalo podem entrar em conflito com a preservação das condições iniciais. Em verdade, em alguns dos *PVIs* experimentados neste trabalho, o processo evolutivo fazia a solução convergir para a solução trivial, quando esta era uma solução, mas, às vezes, a não desejável.

Como preservar as condições iniciais em um *PVI* em um processo evolutivo utilizando bases ortogonais?

Resposta

A secção 3.6 do capítulo 3 mostra como impor uma relação entre os coeficientes de forma a preservar as *CI* durante a evolução, considerando a base canônica e a de Legendre.

2.2.5 Qualidade de uma solução numérica

Os métodos numéricos nos fornecem como resposta a uma equação diferencial um conjunto seccional de polinômios de baixa ordem (2 a 5 conforme a precisão do método). Em todos eles, o valor da derivada de maior ordem, ao longo do intervalo, é o que resta ao substituirmos a função e as outras derivadas na *ED*. Assim, a avaliação do erro no processo de integração numérica fica dificultada, uma vez que, em cada ponto do intervalo, o valor da derivada de maior ordem **é o que lhe resta ser**.

Considerando o conjunto de pontos que obtemos ao aplicar um método numérico na resolução de um *PVI*, haveria uma forma de avaliar a qualidade do processo de integração somente pelos resultados produzidos, independentemente do processo utilizado?

Resposta

A secção 5.2 apresenta um critério objetivo para se avaliar o quanto a solução numérica de fato atende ao *PVI*, usando funções ortogonais e completas, obtendo a margem de erro em cada coeficiente. Além de permitir a avaliação da qualidade, o processo permite construir a margem de erro da solução numérica ao longo do intervalo.

2.2.6 Avaliação dos coeficientes: adaptação

O critério objetivo citado no tópico anterior poderia balizar uma evolução das soluções de tal sorte que elas pudessem, cada vez mais, em um processo adaptativo, convergir para o melhor polinômio?

Resposta

Considerando-se que um polinômio de grau n seja o melhor candidato até então encontrado para resolver o *PVI* (critério *LG*), é natural perguntar-se quão longe ele está da solução.

À primeira vista, essa pergunta parece ingênua, já que se fosse conhecido exatamente quão longe ele está da solução, saber-se-ia a solução, mas não é esse o caso.

A secção 5.1.1 descreve como, dado um candidato à solução, obter a margem de erro em cada um dos seus coeficientes, que irá restringir a sua faixa de variação no processo evolutivo, a partir do resíduo que ele deixa ao ser substituído no *PVI*. Essa restrição estabelece um foco na busca de candidatos melhores.

Encontrado um candidato melhor, a faixa de busca — aqui denominada **envelope** — se estreita ainda mais, focalizando ainda mais a busca.

Com esse recurso, as novas gerações vão aprendendo com as antigas, a seleção vai descartando os piores, os melhores vão produzindo bons descendentes, e os *crossing-over* abrem a oportunidade de se pegar o melhor de cada um. Além disso, pequenas mutações varrem os vãos entre os candidatos. O processo pode continuar até uma precisão pré-estabelecida pelo usuário, ou até que se observe a estagnação evolutiva.

Inicialmente, esta pesquisa tomava com ponto de partida os pontos gerados por métodos numéricos consagrados, que são funções residentes do Matlab. Em termos de evolução a idéia foi chegar ao *homo sapiens* já partindo de um primata. Mas a pergunta nº 7, se o processo evolutivo funcionaria partindo da estaca zero, ou seja, da sopa primordial, parecia não calar.

A precisão e a rapidez dos métodos numéricos já existentes permitiram um balizamento do processo evolutivo. Foram muitos os ajustes nas taxas de mutação, *crossing-over*, número de cromossomos etc.

Quando as curvas obtidas começaram praticamente a se superpor, com polinômios de grau relativamente baixo, parecia a hora de tentar.

Partindo de um polinômio identicamente nulo, em cerca de 2000 gerações o processo chegou à mesma solução que chegava quando partia dos pontos do método de Adams [Hull et al., 1972].

2.2.7 Abrangência e custo computacional

Há, pelo menos, três perguntas que emergem imediatamente quando da apresentação de um novo processo, teoria ou tecnologia. Primeiramente, se o novo faz o que os antigos faziam. Depois, se ele faz alguma coisa que os antigos não fazem e, finalmente, o quanto custa. São essas perguntas que este tópico procura responder.

Respostas

Tão admirável é a precisão dos métodos numéricos consagrados para resolução das *EDs* que, inicialmente, esta pesquisa partia deles na construção dos coeficientes a serem evoluídos, conforme já descrito no tópico anterior. Mas o emperramento de vários deles em

alguns casos patológicos⁵, com equações não lineares, levou este trabalho à tentativa de aplicar o método evolutivo nesses casos. Em todos eles os coeficientes evoluíram para ajustes melhores e foi possível obter-se uma solução aproximada. Esses casos são apresentados no capítulo 6 — Resultados e Análise.

Nesse aspecto, o *SAM* aplicado a alguns *PVIs* fez algo que outros não fizeram.

Nos casos em que a solução exata era polinomial, o *SAM* a encontrou em relativamente poucas gerações, algo fora da abordagem dos métodos numéricos tradicionais.

Entretanto, é preciso salientar que o polinômio obtido pelo *SAM* **não** tem, necessariamente, precisão maior do que os métodos numéricos consagrados, mas pode chegar bem perto, dependendo da extensão do intervalo e do grau do polinômio utilizado. Isso pode ser explicado pelo fato de que os métodos tradicionais usam polinômios de baixo grau (4 ou 5), mas fatiam o intervalo em vários subintervalos, digamos 40, mudando então de polinômio, o que corresponde, praticamente, à utilização de um polinômio de grau 200!

Quanto ao custo computacional do *SAM*, se é maior ou menor do que os métodos tradicionais aplicados ao *PVI*, a resposta depende. Muitas vezes, na impossibilidade de encontrar uma solução analítica para um *PVI* não linear, a função procurada é substituída pelo método, que é então executado a cada vez que a função é chamada. Nesses casos, não há dúvida de que o polinômio seria muito mais econômico. Entretanto, se as condições iniciais mudam a cada chamada da função e, sempre, um novo polinômio precisa ser encontrado, os métodos tradicionais são incomparavelmente mais rápidos. Mesmo assim, alguns ganhos obtidos nesta pesquisa permanecem: são as respostas às seis perguntas anteriores e a continuidade da função obtida, em lugar de um discreto conjunto polinômios de baixo grau.

2.2.8 Margens de erro

Um dos problemas das soluções numéricas nas *EDs* é a determinação da margem de erro. Na maioria deles, é possível se estipular a máxima margem de erro tolerável a cada passo (*h*), mas como a integração ao longo de todo intervalo pode compreender milhares de passos, não se sabe exatamente qual foi o erro acumulado ao final do processo de integração. Por exemplo, a respeito do método de Runge-Kutta, Arfken & Weber [1995, p. 531] argumentam:

⁵ Matriz de massa nula ou equações não lineares que levam os algoritmos para o campo dos números complexos.

O método de Runge-Kutta é estável, isto é, pequenos erros não são amplificados. É auto-inicializável, ou seja, precisamos apenas ter x_0 e y_0 e seguir em frente. Mas tem desvantagens. Quatro cálculos separados de $f(x, y)$ são requeridos a cada passo. Os erros, embora da ordem de h^5 em cada passo, não são conhecidos.

Seria possível determinar objetivamente quais são os valores extremos do erro absoluto em um processo de integração numérica e qualidade do método, conhecendo apenas os pontos obtidos e a equação, seja o processo qual for?

Resposta

A secção 5.2 mostra que é possível se determinar um majorante para o erro de um processo, além de mostrar também como detectar se o processo tem coesão interna para os resultados obtidos, e qual é a coerência entre os resultados obtidos e a proposta do *PVI*.

2.3 Métodos numéricos de integração para os *PVIs* de 1ª ordem

Campos [2007, p. 324] destaca que: “os métodos analíticos são restritos a algumas formas especiais de função, visto que nem toda *EDO* tem solução analítica.”

Ainda conforme Campos, o problema do valor inicial de **primeira ordem** apresenta a forma:

$$\text{Eq. } \{1\} \begin{cases} y' = G(x, y) \\ y(a) = \eta \\ a \leq x \leq b \text{ e } -\infty \leq y \leq \infty \end{cases}$$

A solução do *PVI* é uma função $y = y(x)$ contínua e diferenciável que satisfaça a Eq. {1}.

Para os *PVIs* de primeira ordem, vale o Teorema de Lipschitz:

Seja $G(x, y)$ uma função definida e contínua para todo (x, y) na região D definida por $a \leq x \leq b$ e $-\infty \leq y \leq \infty$, sendo a e b finitos, e seja uma constante L tal que:

$$\|G(x, y) - G(x, y^*)\| \leq L \|y - y^*\|, \text{ seja válida para todo } (x, y), (x, y^*) \in D, \text{ então}$$

existe uma única solução $y(x)$ do *PVI*, onde $y(x)$ é contínua e diferenciável para todo $(x, y) \in D$. [Campos, *ib.* p. 324]

A inequação apresentada no parágrafo anterior é conhecida como condição de Lipschitz, e L é uma constante que também leva o seu nome.

Já em 1768, Euler propôs um método para resolução numérica do *PVI*, usando aproximações sucessivas de primeira ordem, na forma:

$$\text{Eq. } \{ 2 \} \quad y_1 = y_0 + hy'_0,$$

em que h é o passo do processo sucessivo de integração, de forma que:

$$(x_1, x_2, \dots, x_n) = (x_0 + h, x_0 + 2h, \dots, x_0 + nh)$$

Considerando a expansão de Taylor para $y(x_0 + h)$, considerando que a série seja convergente, tem-se:

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{h^2}{2!} y''(x_0) + \dots$$

Se a série for truncada para iguais ou superiores a h^2 , então:

$y_1 = y_0 + h \cdot G(x_0, y_0)$. Uma vez que $x_0 = a$ e $y(a) = \eta$, definindo-se o passo da integração obtém-se então y_1 .

Utilizando essa mesma idéia sucessivamente, tem-se:

$$\text{Eq. } \{ 3 \} \quad y_{j+1} = y_j + h \cdot G(x_j, y_j), \text{ conforme Arfken \& Weber [1995, p. 530].}$$

Esses autores apontam que “Claramente, ele [o método] é sujeito a sérios erros pelo negligenciamento de termos de ordem h^2 e superiores.” [*Id.*, *ib.*] No entanto, o método de Euler proveu uma frutífera idéia para que fossem desenvolvidos métodos de ordens superiores, baseados na expansão de Taylor, ou de Lagrange, como mostram as duas secções seguintes.

2.3.1 Método de Runge-Kutta⁶

O método de Runge-Kutta de ordem 4 (erro de ordem h^5) tem como base as seguintes expressões:

$$\text{Eq. } \{ 4 \} \quad y_{n+1} = y_n + (k_0 + 2k_1 + 2k_2 + k_3)/6, \text{ onde}$$

⁶ Conforme Campos [2007, p. 328], C. D. T. Runge desenvolveu o primeiro método em 1895, e M.W. Kutta elaborou a formulação geral em 1901.

$$\text{Eq. \{ 5 \}} \left\{ \begin{array}{l} k_0 = h \cdot G(x_n, y_n) \\ k_1 = h \cdot G(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_0) \\ k_2 = h \cdot G(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ k_3 = h \cdot G(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \end{array} \right.$$

Essas equações são construídas de forma a obedecerem à expansão de Taylor até ordem 4 [Arfken & Weber, 1995, p. 530].

Para o caso especial em que $G(x, y)$ seja somente função de x , a Eq. {4} se reduz à regra de Simpson para integração entre x_n e x_{n+1} .

Richard H. Battin [1976; 1977] ampliou o algoritmo para ordem 8, em um caso particular, com relativo sucesso, mas o processo se revelou instável sob determinadas condições.

Em 1980, J.R. Dormand e P.J. Prince publicaram um minucioso estudo do método de Runge-Kutta, ampliando-o para ordem 5, possibilitando-lhe um passo variável e inserindo termos para reforçar a sua estabilidade, considerando um sistema com 25 equações e 25 incógnitas [Dormand & Prince, 1980].

Atualmente, tal é a eficiência e relativa rapidez desse método, que se torna quase desnecessário discorrer sobre sua aplicabilidade. Mesmo assim, algumas melhorias continuam sendo propostas [Calvo et al., 2008] e [Zhu et al., 2008], em situações específicas.

O método de Runge-Kutta, com os aprimoramentos de Dormand & Prince, é função residente do Matlab 7® (função ode45.m).

2.3.2 Método de Adams

Basicamente, o método de Adams, cuja formulação original remonta ao séc. XIX, utiliza-se do polinômio interpolador de Lagrange.

Atualmente, pelas contribuições que recebeu ao longo do tempo, o método é denominado Adams-Bashforth-Moulton e é também função residente do Matlab 7 (função ode113.m).

Dadas $n+1$ ordenadas correspondentes às abscissas $x_0, x_1, \dots, x_n$, existe um polinômio de grau n ($L_n(x)$) que contém todas essas ordenadas, cuja forma é:

$$L_n(x) = \sum_{i=0}^n y_i' \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}, \text{ onde } y_i' = G(x_i, y_i).$$

Para uma aplicação de ordem 4, além da condição inicial, o método precisaria de mais quatro ordenadas, que podem advir do método de RK-Dormand & Prince, ou seja, o método de Adams não é auto-inicializável.

Obtido o polinômio, ele pode ser integrado analiticamente entre x_j e x_{j+4} , e assim criar uma expressão para a extrapolação de y_{j+5} .

Arfken & Weber [1995, p. 532] apontam que a formulação original era instável, mas se aprimorou com a invenção do mecanismo preditor-corretor inventado por Hamming⁷.

2.3.3 Comparativo dos métodos de Adams e Runge-Kutta do Matlab

A seguir, analisa-se o comportamento dos dois métodos títulos desta secção para três *PVIs* [Campos, 2007, p. 342], com soluções analíticas conhecidas, utilizando-se as funções já residentes no Matlab 7. Comparações de desempenho e precisão para vários outros *PVIs* podem ser encontradas na referência [Shampine & Reichelt, 1997].

Margens de erro

Em vista da expressiva rapidez dos processadores atuais, os intervalos foram divididos em 1000 partes e foram usadas as opções de precisão máxima no Matlab 7, para os dois métodos, de forma que o parâmetro “options” foi:

```
options = odeset('RelTol', 2.22045e-014, 'AbsTol', 1e-21);
```

A raiz do erro médio quadrático (*RMS*) e o erro máximo ($e_{\max}$), máximo do módulo da diferença entre a solução analítica e a obtida, são comparados na tabela 1, para cada um dos *PVIs* propostos, em cada um dos métodos.

Explicitamente, a raiz do erro quadrático médio em um intervalo $[a, b]$, é:

$$\text{Eq. } \{ 6 \} \quad RMS = \frac{\sqrt{\int_a^b (\text{erro})^2 dx}}{b - a}.$$

Os *PVIs*, domínios e soluções analíticas são apresentados na tabela seguinte, com os respectivos erros.

Tabela 2.4. Comparativo dos MNs: o primeiro erro se refere à função e o segundo a sua derivada.

<i>PVI</i>	Solução Analítica	<i>Erros</i>			
		<i>RK</i>		<i>Adams</i>	
		<i>RMS</i>	<i>e_{máx}</i>	<i>RMS</i>	<i>e_{máx}</i>
$y_1' = -2x^2 y_1^2, y_1(0) = 2$ $x \in [0, 2]$	$y_1 = \frac{6}{4x^3 + 3}$	4e-14 3e-14	9e-13 7e-13	1e-14 1e-14	3e-14 5e-14
$y_2' = \cos(x)y_2, y_2(0) = 1$ $x \in [0, 10]$	$y_2 = e^{\sin(x)}$	8e-15 1e-14	7e-13 6e-13	6e-15 7e-15	1e-13 5e-14
$y_3' = \sin(x) - y_3, y_3(0) = 0$ $x \in [0, \pi]$	$y_3 = \frac{e^{-x} + \sin(x) - \cos(x)}{2}$	2e-15 3e-15	1e-13 1e-13	2e-15 2e-15	2e-14 2e-14

Nos exemplos analisados, nota-se que o método de Adams se mostrou ligeiramente superior, mas as margens de erro e o tempo de processamento⁸ são tão pequenos em ambos, que é praticamente indiferente o uso de um ou de outro, na maioria dos casos.

Matriz de massa singular

Considere-se a EDO de primeira ordem:

$$y'm - y - x^2 \cos(x) = 0, x \in [0, 2] \text{ e } y(0) = 0 \text{ e } m = m(x, y).$$

Uma vez que, nos MNs, a equação será avaliada em várias abscissas $(x_0, x_1, \dots, x_n)$, ela pode ser considerada uma equação matricial, na forma:

$$M_m Y' = Y + X^2 \cos(X), \text{ onde}$$

$$X = \text{diag}(x_0, x_1, \dots, x_n), Y = \text{diag}(y_0, y_1, \dots, y_n) \text{ e assim por diante.}$$

Nessa formulação, a matriz M_m é matriz de massa [Shampine & Reichelt, 1997].

Se em $x = x_0$ a matriz de massa for singular, nem o método de Runge-Kutta, nem o método de Adams (que depende do RK), conseguem inicializar a seqüência de cálculos, pois o sistema de equações fica indeterminado.

Supondo, por exemplo, $m = x$, o cálculo de $y'(0)$, levaria à indeterminação:

$$y'(0) = \frac{0}{0}. \text{ Nessa condição, não é válido o Teorema de Lipschitz, apresentado na secção 2.3,}$$

⁷ Como referência, Arfken apresenta: Hamming, R. W. (1994), *Numerical Methods for Physics*, Englewood Cliffs, NJ: Prentice-Hall, mas essa obra não foi consultada nesta pesquisa.

⁸ Em um processador Core-Duo 2,4 GHz com 2 GB de RAM, o tempo foi menor do 1/10 de s.

pois $G(0,0)$ é indeterminado e não existe L que satisfaça a imposição do Teorema, ou seja, neste caso, não há garantia de que PVI tenha solução única.

De fato, o exemplo 2 do capítulo 6, secção 6.1, mostra como, a partir do *SAM* e considerações analíticas, foi possível determinar-se uma família de soluções analíticas para o PVI :

$$y'x - y - x^2 \cos(x) = 0, \quad x \in [0, 2] \text{ e } y(0) = 0.$$

2.4 Métodos numéricos para $PVIs$ de ordem 2 ou superiores

As secções anteriores descrevem os MNs para os $PVIs$ de 1ª ordem.

As equações de ordem 2, ou superiores, podem ser escritas na forma de um sistema de equações diferenciais de primeira ordem, e os MNs podem então ser executados.

Se para cada uma das equações do sistema valer o Teorema de Lipschitz, então o PVI tem solução única.

Se a ED for linear, recai-se em um sistema linear. Usando-se as funções residentes dos MNs do Matlab 7, a transformação do PVI para um sistema de equações deve ser feita pelo usuário antes da aplicação do MN . O *SAM* faz essa transformação automaticamente, em conformidade com a ordem da ED , utilizando ferramentas simbólicas do Matlab 7, para equações diferenciais de até quarta ordem, lineares ou não.

Por exemplo, o PVI de segunda ordem:

$$y'' = y' + 2y - x^2 \quad x \in [0, 1], \text{ com } y(0) = 1 \text{ e } y'(0) = 0, \text{ cuja solução analítica é}$$

$$y(x) = \frac{1}{4}(e^{2x} + 2x^2 - 2x + 3) \quad [\text{Campos, 2007}], \text{ pode ser escrito na forma:}$$

$$\begin{cases} y_1' = y_2 \\ y_2' = y_2 + 2y_1 - x^2 \end{cases}, \text{ com } y_1(0) = 1 \text{ e } y_2(0) = 0.$$

Os gráficos e as margens de erro, utilizando-se o RK-Dormand & Prince (ode45.m), são apresentados a seguir.

Gráfico 2. 4. Solução numérica do *PVI* de 2ª ordem.

Função	Erro máximo
D0	2e-14
D1	4e-14
D2	8e-14

Capítulo 3. Fundamentação Analítica

Este capítulo apresenta a fundamentação analítica para a busca de soluções polinomiais ou trigonométricas nas soluções das *EDOs*.

Neste trabalho considera-se que a função $f(x)$ seja contínua em um intervalo fechado $[a, b]$, com derivadas também contínuas, e que seja de quadrado integrável; em outras palavras, exige-se que a integral do quadrado da função no intervalo considerado seja finita. A secção 4.1 demonstra a necessidade dessa exigência. Dessa forma, o conjunto das funções que satisfazem as condições descritas perfaz um espaço de Hilbert [Arfken & Weber, 1995, p. 566].

Além disso, este trabalho considera apenas funções reais de variáveis reais, embora extensões possam ser feitas para funções de variáveis complexas.

O primeiro aspecto a ser observado é se essa busca proposta no 1º parágrafo é **lícita**, ou seja, se é possível construir, a partir de um conjunto de potências de x , ou outras famílias de funções, uma seqüência que convirja **uniformemente** para representar determinada função contínua no intervalo $[a, b]$.

O teorema apresentado na secção seguinte garante a validade da busca e a convergência desejada.

3.1 Teorema de Weierstrass

Em 1891, Karl Weierstrass demonstrou que os polinômios possuem a propriedade de mimetizar muito bem as funções contínuas em intervalos finitos.

Conforme enfatiza Moura [2002, p. 182]:

... por melhor que seja o instrumento de medição de que disponhamos, dada uma função contínua arbitrária no intervalo $[a, b]$, existe um polinômio que consegue confundir nosso instrumento, o qual não vai poder distinguir entre esse polinômio e a função inicialmente introduzida. Como a avaliação numérica de um polinômio exige apenas operações algébricas (soma, multiplicação), facilmente processadas em um computador, este resultado parece indicar um caminho insuperável para a construção de algoritmos de aproximação.

O instrumento citado por Moura pode ser o próprio computador, já que, se a aproximação tiver uma margem de erro menor que o limite de casas decimais do computador utilizado, para esse instrumento a função e a aproximação serão indistinguíveis.

O parágrafo seguinte apresenta o enunciado formal do teorema de Weierstrass.

Se $f(x)$ é uma função contínua no intervalo fechado $[a, b]$, nesse intervalo existe uma seqüência de polinômios $P_n(x)$ tal que: $\lim_{n \rightarrow \infty} P_n(x) = f(x)$, uniformemente no intervalo $[a, b]$. [Byron & Fuller, 1992, p.229].

O critério de convergência uniforme garante que: qualquer que seja $\varepsilon > 0$, existe um número N , independente de x no intervalo $[a, b]$, tal que $|f(x) - P_n(x)| < \varepsilon$ para todo $n \geq N$. [Arfken e Weber, 1995, p. 309]

Isso significa que não importa o quão pequeno se queira ε , pode-se sempre escolher n suficientemente grande de forma a satisfazer $|f(x) - P_n(x)| < \varepsilon$, em todo o intervalo. A figura seguinte ilustra a visão geométrica desse confinamento de $P_n(x)$, discutido por Arfken & Weber [*ib.*, p.310].

Gráfico 3. 1. O polinômio $P_n(x)$ é uma aproximação de $f(x)$, no intervalo $[a, b]$, com erro menor do que ε .

3.2 Ortogonalidade e completude

O teorema de Weierstrass exige que a função seja contínua. Uma redução nas exigências a respeito da função pode levar a aplicações mais gerais. Por exemplo, exigir que a função seja apenas seccionalmente contínua leva ao critério da **convergência em média**, no intervalo considerado.

Conforme Byron [Byron & Fuller, 1992, p. 220], por definição, uma seqüência de funções $P_n(x)$ **converge em média** para $f(x)$ se $\lim_{n \rightarrow \infty} \int_a^b \left(f(x) - \sum_{k=0}^n a_k \cdot P_k(x) \right)^2 \cdot dx = 0$.

Se x for um ponto de descontinuidade de $f(x)$, com essa aproximação, a seqüência $P_n(x)$ converge para a média aritmética dos valores de $f(x)$ à direita e à esquerda da descontinuidade, ou seja, $\sum_{k=0}^n a_k \cdot P_k(x) = \frac{f(x_+) + f(x_-)}{2}$.

Sem dúvida, a convergência uniforme acarreta a convergência em média, mas a recíproca não é verdadeira, já que, na convergência em média, podem existir pontos de descontinuidade, e a condição imposta para a convergência uniforme não será verificada.

Ortogonalidade

Um conjunto de funções reais, de variáveis reais é ortogonal em relação a uma função “peso” ($w(x)$), não-negativa, no intervalo $[a, b]$ se:

$$\text{Eq. \{ 7 \}} \quad (f_n \cdot f_m) = \int_a^b f_n \cdot f_m \cdot w \cdot dx = \begin{cases} 0 & \text{se } m \neq n \\ g(m) & \text{se } m = n \end{cases}.$$

Usualmente, o conjunto de funções é normalizado de forma que $g(m) = 1$, mas isso não é imperativo.

Contudo, supondo as funções normalizadas e usando o delta de Kronecker $\delta_{mn} = \begin{cases} 0 & \text{se } m \neq n \\ 1 & \text{se } m = n \end{cases}$, é então possível se escrever:

$$\text{Eq. \{ 8 \}} \quad (f_n \cdot f_m) = \int_a^b f_n \cdot f_m \cdot w \cdot dx = \delta_{mn} \quad (\text{funções normalizadas}).$$

No caso da série de Fourier, $g(n) = \frac{1}{\pi}$ e $[a, b] = [-\pi, \pi]$, para os polinômios de Legendre $g(m) = \frac{2}{2m+1}$ e $[a, b] = [-1, 1]$, como mostram as secções 3.4 e 3.3, respectivamente.

Tanto na série de Fourier, quanto na de Legendre, a função peso é a unidade.

Completeness

Um conjunto de funções (P_n) é completo se, para qualquer função $f(x)$ bem-comportada (seccionalmente contínua, pelo menos) no intervalo $[a, b]$, vale a relação:

$$\text{Eq. \{ 9\}} \quad \lim_{n \rightarrow \infty} \int_a^b \left(f(x) - \sum_{k=0}^n a_k \cdot P_k(x) \right)^2 \cdot dx = 0 \quad [\text{Arfken, 1995, p. 566}].$$

Admitindo válida a igualdade anterior, diz-se que o conjunto P_n é uma base na qual se escreve $f(x)$, em analogia com a forma de escreverem-se os vetores tridimensionais. A analogia é a mesma para o termo “completeness”. Por exemplo, não é possível descrever um vetor tridimensional qualquer usando como base somente dois vetores.

De fato, Hilbert demonstrou que o conjunto das funções seccionalmente contínuas, de quadrado integrável, em intervalos limitados, é um espaço vetorial de dimensão infinita, mesmo para funções de variáveis complexas; mas o aprofundamento e o detalhamento dessa discussão estão fora dos objetivos deste trabalho, mesmo porque ela se encontra largamente registrada na literatura. Ver, por exemplo, referências [Byron & Fuller, 1992; Arfken & Weber, 1995].

Propriedades

Neste tópico, as propriedades seguintes serão obtidas supondo-se que as bases funcionais consideradas sejam ortogonais normalizadas (ortonormais) e completas no intervalo $[a, b]$.

1. Cálculo dos coeficientes

Como a base é completa, tem-se:

$$\text{Eq. \{ 10\}} \quad f(x) = \sum_{i=0}^{\infty} a_i \cdot P_i.$$

Considerando um vetor da base P_k , pode-se escrever:

$$(f \cdot P_k) = \int_a^b P_k \cdot f = \int_a^b \sum_{i=0}^{\infty} a_i \cdot P_i \cdot P_k = \sum_{i=0}^{\infty} a_i \cdot \delta_{ik} = a_k, \text{ portanto:}$$

$$\text{Eq. \{ 11\}} \quad a_k = \int_a^b P_k \cdot f. \text{ (base normalizada), conforme Byron [Byron \& Fuller, 1992, p.}$$

Neste ponto, é importante observar que o valor de a_k não depende dos valores dos outros coeficientes. Esse é um grande trunfo da ortogonalidade. Se por um processo qualquer for possível melhorar algum dos coeficientes de uma aproximação e isso não afetar os outros coeficientes, o processo pode convergir muito mais rapidamente, ou seja, o uso de bases ortogonais impede que a melhora de um coeficiente piore os outros.

2. Mínimos quadrados

Supondo que se esteja aproximando uma função $f(x)$ até a ordem k , quais seriam os valores dos coeficientes b_i que tornariam mínima a expressão:

$$\text{Eq. \{ 12\}} \quad M = \int_a^b \left(f(x) - \sum_{i=0}^k b_i \cdot P_i(x) \right)^2 \cdot dx$$

Em termos de aplicações da Engenharia, ou da Física-Matemática, a pergunta seria: quais são os valores dos b_i que atendem ao critério dos mínimos quadrados?

Desenvolvendo o quadrado, obtém-se:

$$M = \int_a^b \left(f^2 - 2f \sum_{i=0}^k b_i \cdot P_i + \sum_{i,j=0}^k b_j \cdot P_j \cdot \sum_{i=0}^k b_i \cdot P_i \right) \cdot dx$$

Como $\int_a^b P_i \cdot P_j = \delta_{ji}$ e $a_i = \int_a^b P_i \cdot f$, fazendo a substituição, tem-se:

$$M = \int_a^b (f^2) - 2 \sum_{i=0}^k b_i \cdot a_i + \sum_{i=0}^k b_i^2 \cdot \int_a^b P_i^2 \cdot dx$$

Agora, soma-se a M e subtrai-se o termo $\sum_{i=0}^k a_i^2$, resultando em:

$$M = \int_a^b (f^2) + \sum_{i=0}^k (b_i - a_i)^2 - \sum_{i=0}^k a_i^2, \text{ adaptado de Byron [Byron \& Fuller, 1992, pp.221-222]}$$

Como os a_i são constantes obtidas pela Eq. \{11\}, o mínimo para expressão anterior vai ocorrer quando $b_i = a_i$, ou seja, o critério que utiliza a **função peso unitária em um intervalo** $[a,b]$, usando uma base ortonormal completa, obedece ao critério dos mínimos quadrados.

Duas são de particular interesse em computação e engenharia: a base trigonométrica de Fourier, do intervalo $[-\pi, \pi]$ e a dos polinômios de Legendre, no intervalo $[-1, 1]$.

Comentários

1. Além da convergência uniforme e da convergência em média, há a **convergência no ponto**. As duas primeiras são otimizadoras globais da função, mas sacrificam alguns pontos, enquanto a convergência no ponto é um otimizador local. A série de Taylor é um exemplo de otimização no ponto, como mostra o 7º exemplo do capítulo 6.
2. Em computação, os cálculos das funções trigonométricas são feitos utilizando polinômios de alto grau, mas, mesmo assim, truncados. Por essa razão a preferência deste trabalho na escolha da base ortogonal recaiu sobre os polinômios de Legendre.
3. Como apontado, a não ortogonalidade da base canônica tem como decorrência que a melhora de um coeficiente pode piorar todos os outros. Isso acaba por demandar maior esforço computacional na procura de uma aproximação para a função.

3.3 Polinômios de Legendre

Os polinômios de Legendre constituem uma base ortogonal completa no intervalo $[-1, 1]$. As referências [Boas, 1983; Arfken & Weber, 1995; Byron & Fuller, 1992] demonstram essas propriedades detalhadamente.

Uma das formas de se obter um polinômio de Legendre de grau n (P_n) é pela aplicação direta da expressão:

$$\text{Eq. \{ 13\}} \quad P_{L,n}(x) = \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \frac{(2n-2k)!}{2^n k!(n-k)!(n-2k)!} x^{n-2k},$$

considerando que: $[n/2] = \begin{cases} \frac{n}{2} & \text{para } n \text{ par} \\ \frac{n-1}{2} & \text{para } n \text{ ímpar} \end{cases}$. [Arfken & Weber, 1995, p. 696].

Entretanto, esses polinômios apresentam um vasto conjunto de propriedades que abreviam vários processos algébricos e computacionais. A secção seguinte apresenta algumas delas.

3.3.1 Propriedades

Há algumas relações entre os P_n consecutivos e também entre os P_n e suas derivadas que permitem obter um novo P_n ou relacionar derivadas, a partir de outros. Essas relações são conhecidas como relações de recorrência. O tópico seguinte apresenta algumas delas.

Relações de recorrência

Apresentam-se a seguir algumas relações de recorrência [Boas, 1983, p. 491].

$$\text{Eq. \{ 14\}} \quad nP_n = (2n-1) \cdot u \cdot P_{n-1} - (n-1) \cdot P_{n-2}$$

$$\text{Eq. \{ 15\}} \quad uP_n' - P_{n-1}' = nP_n$$

$$\text{Eq. \{ 16\}} \quad (2n+1) \cdot P_n = P_{n+1}' - P_{n-1}'$$

Como $P_0 = 1$ e $P_1 = u$, a relação de recorrência {14} permite a obtenção rápida de $P_2 = 1,5u^2 - 0,5$, a seguir $P_3 = \frac{5}{2}u^3 - \frac{3}{2}u$, e assim por diante. Geralmente, esse é o processo usado em computação, para se evitarem os fatoriais de ordem alta que podem ultrapassar a quantidade de dígitos suportada pelo processador.

A tabela seguinte apresenta os coeficientes das potências de u para os polinômios de Legendre até ordem 5. A obtenção desses coeficientes pode ser feita pela Eq. {13}, ou, como dito no parágrafo anterior, pela aplicação sucessiva da recorrência da Eq. {14}.

Tabela 3. 1. Coeficientes do Polinômio de Legendre de ordem n na base canônica.

	u^0	u^1	u^2	u^3	u^4	u^5	u^6	u^7	u^8	u^9
P_0	1	0	0	0	0	0	0	0	0	0
P_1	0	1	0	0	0	0	0	0	0	0
P_2	$-1/2$	0	$3/2$	0	0	0	0	0	0	0
P_3	0	$-3/2$	0	$5/2$	0	0	0	0	0	0
P_4	$3/8$	0	$-15/4$	0	$35/8$	0	0	0	0	0
P_5	0	$15/8$	0	$-35/4$	0	$63/8$	0	0	0	0
P_6	$-5/16$	0	$105/16$	0	$315/16$	0	$-693/16$	0	$429/16$	0
P_7	0	$-35/16$	0	$315/16$	0	$-693/16$	0	$429/16$	0	0
P_8	$35/128$	0	$-315/32$	0	$3465/64$	0	$-3003/32$	0	$6435/128$	0
P_9	0	$315/128$	0	$-1155/32$	0	$9009/64$	0	$-6435/32$	0	$12155/128$

Máximos e mínimos

No intervalo $[-1, 1]$, no qual os polinômios de Legendre perfazem uma base completa, o máximo valor que eles assumem é 1 e o menorante⁹ do mínimo é -1 . (Arfken & Weber, p. 706).

A secção seguinte demonstra que $P_n(1) = 1$, qualquer que seja n , logo, o extremo superior do domínio é um máximo global. Para n par, o extremo inferior do domínio também é um ponto de máximo global, pois $P_n(-1) = (-1)^n$. Para n ímpar o extremo inferior é um ponto de mínimo global.

⁹ Se o polinômio de Legendre for de ordem par, o mínimo no intervalo em que são definidos é maior do que -1 . Por exemplo, o mínimo para o polinômio de Legendre de ordem 2 ocorre em $x = 0$ e vale $-0,5$, ou seja, é maior do que -1 .

Gráfico 3. 2. Polinômios de Legendre da ordem 0 até 5. Sendo n a ordem do polinômio, ele tem n raízes reais.

Gráfico 3. 3. Polinômios de Legendre de ordem 6 a 10.

Figura 3. 1. Polinômio de Legendre de ordem 21. À medida que aumenta a ordem do polinômio, os valores que ele assume entre os extremos do intervalo vão ficando cada vez menores, “preenchendo” blocos cada vez menores da função aproximada.

3.3.2 Integral de uma função expressa na base de Legendre

A aproximação de funções por séries de polinômios de Legendre, ou outras bases ortogonais, demonstrou ser uma ferramenta efetiva na resolução de *PVIs*, como denotam várias publicações, por exemplo, as referências [Babolian & Fattahzadeh, 2007], [Elbarbary, 2002], [Sezer & Kesan, 2000], [Psihoyios & Simos, 2003].

Para resolução de alguns *PVIs* de quarta ordem, Elbarbary [2002] utilizou um método no qual um polinômio de Legendre pode ser escrito como combinação linear das derivadas de polinômios de Legendre, e as sucessivas integrais de um polinômio de Legendre como combinação linear de elementos da própria base, usando a base normalizada.

Nesta tese escolheu-se obter a derivada de um polinômio de Legendre em função dos próprios polinômios da base, cuja demonstração é feita no tópico seguinte, e para a integral propõe-se uma demonstração mais simples, baseada nas relações de recorrência e com os polinômios não normalizados. O fato de os polinômios não serem normalizados evita o cálculo prematuro com elementos irracionais na matriz que será construída e os decorrentes erros de truncamento.

Considerando-se a Eq. {15}, pode-se escrever:

$$P_n = \frac{P'_{n+1} - P'_{n-1}}{2n+1}, \text{ com } n \in \mathbb{N}^*.$$

Integrando-se a expressão acima entre -1 e x , obtém-se:

$$\int_{-1}^x P_n(\xi) d\xi = \frac{P_{n+1} - P_{n-1}}{2n+1} \Big|_{-1}^x$$

Considerando que a integral seja de -1 até 1 , observando que $P_k(1) = 1$, e ainda

que $\int_{-1}^1 P_k(\xi) \cdot d\xi = 0$, qualquer que seja $k \in \mathbb{N}^*$, obtém-se:

$$0 = \frac{P_{n+1}(-1) - P_{n-1}(-1)}{2n+1}, \text{ ou seja, } P_{n+1}(-1) = P_{n-1}(-1), \text{ qualquer que seja } n \in \mathbb{N}^*.$$

Explicitamente, como $P_0(-1) = 1$ e $P_1(-1) = -1$, obtemos:

$$P_n(-1) = (-1)^n, \text{ agora com } n \in \mathbb{N}.$$

Finalmente, pode-se escrever:

$$\text{Eq. \{ 17\}} \quad \int_{-1}^u P_n(\xi) \cdot d\xi = \frac{P_{n+1}(u) - P_{n-1}(u)}{2n+1} \quad n \in \mathbb{N}^*.$$

Para $n=0$, faz-se a integral diretamente:

$$\text{Eq. \{ 18\}} \quad \int_{-1}^u P_0(\xi) \cdot d\xi = \int_{-1}^u 1 \cdot d\xi = u + 1 = P_1(u) + P_0.$$

Considerando agora uma função $f(u)$ expressa por uma série de polinômios de Legendre até ordem k , pode-se escrever:

$$f(u) = \sum_{n=0}^k c_n P_n(u)$$

Com a relação de recorrência obtida, a função pode ser integrada diretamente,

$$\text{termo a termo, por: } \int_{-1}^u f(\xi) \cdot d\xi = \sum_{n=0}^k c_n \int_{-1}^u P_n(\xi) \cdot d\xi.$$

A secção 3.5.2 mostra como transformar essa operação em um cálculo matricial simples pela matriz de integração para séries de Legendre (M_{LI}).

3.3.3 Derivada de uma função expressa na base de Legendre

A derivada de um polinômio de Legendre é uma combinação linear de polinômios de Legendre de ordens menores, conforme descrito pelo teorema seguinte, obtido neste trabalho.

Teorema da derivada de Legendre

Sendo $P'_n(u)$ a derivada de um polinômio de Legendre de ordem n , então

$$P'_n = \sum_{k=0}^{n-1} \frac{(2k+1)}{2} \cdot [1 - (-1)^{k+n}] \cdot P_k \quad n \in \mathbb{N}^*$$

Prova

Seja $f(u)$ uma expressão polinomial qualquer.

Os coeficientes de Legendre para $f(u)$ são:

$$c_k = \int_{-1}^1 f(u) \cdot P_k(u) \cdot du \cdot \frac{(2k+1)}{2}.$$

Se $f(u) = P'_n(u)$, a função é um polinômio de grau $n-1$.

Fazendo a integral por partes, lembrando que o polinômio de Legendre de grau n é ortogonal a todos os polinômios de graus inferiores a n , de Legendre ou não¹⁰, tem-se:

$$\frac{2c_k}{2k+1} = \int_{-1}^1 \underbrace{P_k}_{u} \cdot \underbrace{P'_n}_{dv} \cdot d\xi = P_k \cdot P_n \Big|_{-1}^1 - \int_{-1}^1 \underbrace{P_n \cdot P'_k}_{(\text{ortogonalidade})} \cdot d\xi = 0$$

Dessa forma, como $P_j(1) = 1$ e $P_j(-1) = (-1)^j$, para qualquer j , tem-se:

$$\frac{2c_k}{2k+1} = 1 - (-1)^k \cdot (-1)^n, \text{ o que acarreta:}$$

$$\text{Eq. \{ 19\} } \quad c_k = \frac{(2k+1)}{2} [1 - (-1)^{k+n}],$$

o que se queria provar.

A secção 3.5.2 mostra como obter a matriz de diferenciação a partir desse último resultado, analogamente ao que será feito para a integração.

¹⁰ As expressões polinomiais de graus menores do que n que não sejam de Legendre podem ser escritas como combinação linear de polinômios de Legendre de graus menores do que n , todos ortogonais à P_n , dessa forma, o produto escalar dessa combinação por P_n será nulo, garantindo a ortogonalidade.

Integração do quadrado de uma função

Uma exigência que se faz para aproximar uma função por polinômios de Legendre é que ela seja de quadrado integrável. Este tópico expõe como calcular a integral do quadrado de uma função expressa por coeficientes de Legendre para que se possa entender a razão dessa exigência,

Seja $f(u) = \sum_{k=0}^n c_k P_k$ a função expressa por uma combinação linear de polinômios de Legendre.

Queremos obter a integral do quadrado dessa função, portanto:

$$\int_{-1}^1 f^2(u) \cdot du = \int_{-1}^1 \left(\sum_{k=0}^n c_k P_k \right)^2 \cdot du$$

Podemos explorar a ortogonalidade dos polinômios de Legendre para simplificar o termo à direita da igualdade, pois $\int_{-1}^1 P_k P_j \cdot du = 0$, se $k \neq j$, assim:

$$\int_{-1}^1 f^2(u) \cdot dx = \int_{-1}^1 \sum_{k=0}^n c_k^2 P_k^2 \cdot du, \text{ ou seja, com a expansão do quadrado, todos os}$$

termos mistos serão nulos após a integração.

Agora fazemos a integral termo a termo:

$$\int_{-1}^1 f^2(u) \cdot dx = \sum_{k=0}^n c_k^2 \int_{-1}^1 P_k^2 \cdot du, \text{ e como } \int_{-1}^1 P_k^2 \cdot dx = \frac{2}{2k+1}, \text{ escrevemos:}$$

$$\text{Eq. \{ 20\} } \quad \int_{-1}^1 f^2(u) \cdot dx = \sum_{k=0}^n \frac{2c_k^2}{2k+1}.$$

Se a integral **não** for limitada, pelos menos um dos coeficientes será **não** limitado também, pois todos os termos do somatório são não negativos.

Em muitos casos, pode-se abreviar a verificação de que a função é de quadrado integrável utilizando uma condição **suficiente** para que isso aconteça, qual seja, de que a função seja limitada. De fato, se $f(u)$ é limitada no intervalo $[a, b]$, tem-se:

$$|f(u)| < M, \text{ logo } \int_a^b f^2(u) \cdot du < \int_a^b M^2 \cdot du = (b-a)M^2, \text{ isto é, se a função for limitada no}$$

intervalo $[a, b]$ a integral não diverge, conforme Boas [1983, pp. 313-314].

3.3.4 Erro nos coeficientes

Consideremos agora a função $f^*(u) = \sum_{k=0}^n c_k^* P_k$ como sendo uma aproximação para função $f(u)$, em que cada coeficiente c_k está afetado por um erro δ_k . Dessa forma, podemos escrever:

$$f^*(u) = \sum_{k=0}^n (c_k + \delta_k) \cdot P_k.$$

A integral do erro quadrático no intervalo $[-1; 1]$ será, então:

$$E_{\text{Int}} = \int_{-1}^1 (f^*(u) - f(u))^2 \cdot du = \int_{-1}^1 \sum_{k=0}^n \delta_k^2 \cdot P_k^2 \cdot du.$$

Como $\int_{-1}^1 P_k^2 du = \frac{2}{2k+1}$, obtém-se:

$$E_{\text{Int}} = \sum_{k=0}^n \frac{2\delta_k^2}{2k+1}, \text{ ou seja, } E_{\text{Int}} = \frac{2\delta_0^2}{1} + \frac{2\delta_1^2}{3} + \frac{2\delta_2^2}{5} \dots + \frac{2\delta_n^2}{2n+1}.$$

Como todos os termos da série são não negativos, o máximo valor possível para um determinado δ_k^2 ocorre quando todos os outros forem nulos. Assim: $\frac{2\delta_{k,\text{máx}}^2}{2k+1} = E_{\text{Int}} \Rightarrow$

$$\text{Eq. \{ 21\} } \quad \delta_{k,\text{máx}} = \sqrt{\frac{2k+1}{2} \cdot E_{\text{Int}}}$$

Importância da ortogonalidade

A secção 2.2.2 deste trabalho apresenta uma pergunta sobre a expansão de uma função $f(x)$ em série de potências. A questão é: não seria possível incorporar pequenas diferenças nos coeficientes das potências menores de forma a compensar o truncamento da série na potência n ?

A argumentação que segue supõe que o domínio da função seja o intervalo $[-1, 1]$. Caso não seja, a secção 3.7 expõe como esse domínio pode ser transposto para obedecer a essa condição.

Supondo que a série de potências seja a série de Legendre, pode-se escrever:

$f(u) = c_0 P_0 + c_1 P_1 + \dots + c_n P_n + R(u)$, onde $R(u)$ é o resíduo da aproximação e contém somente polinômios de Legendre de grau superior a n .

Se o critério de erro no intervalo for o dos mínimos quadrados, então o resíduo pode ser decomposto em polinômios de Legendre¹¹, conforme explicitado na secção 3.3. Os coeficientes δ_k ($k = 0, 1, 2, \dots, n$) dos polinômios de Legendre que irão “preencher” o resíduo

$$\text{serão: } \delta_k = \frac{2k+1}{2} \int_{-1}^1 R(u) \cdot P_k \cdot du = \frac{2k+1}{2} \int_{-1}^1 \left(\sum_{j=n+1}^{\infty} c_j P_j \right) \cdot P_k \cdot du .$$

Rearranjando o somatório e observando-se a ortogonalidade, obtém-se:

$$\delta_k = \frac{2k+1}{2} \sum_{j=n+1}^{\infty} c_j \underbrace{\int_{-1}^1 P_j P_k du}_{(\text{ortogonalidade})=0} \Rightarrow \delta_k = 0 .$$

Esse resultado mostra que não há como “melhorar” os coeficientes dos polinômios de menor ordem de forma a compensar o truncamento da série, se a expansão foi feita em polinômios de Legendre. Por outro lado, se a expansão **não** for a de Legendre, como a de Taylor, por exemplo, as diferenças de coeficientes entre esta última expansão, seja ela qual for, e a de Legendre serão os δ_k . Enfatizando, esta é a conclusão sob o critério de erro dos *MQ*, porém, mudando-se para o critério *LG*, torna-se possível esses coeficientes “evolüirem”.

Esse resultado é de extrema importância quando se pensa em métodos iterativos para aproximação de uma função em determinado intervalo. O resultado mostra que, seja qual for a ordem do método, se ele for polinomial e o critério for o *MQ*, a melhor aproximação será pelos polinômios de Legendre.

3.3.5 Mudança de base: Legendre $\times$ Canônica

Considerando $f(u) = \sum_{k=0}^n c_{k,L} P_k$ uma função escrita na base de Legendre até ordem n , pode-se escrevê-la também na base canônica, por uma operação matricial.

Sem dúvida, é possível expandir a função considerando-se cada dos coeficientes da série e os polinômios de Legendre, para depois agrupar os termos na base canônica, mas isso poderia ser trabalhoso e envolver uma seqüência de operações nas quais os erros de truncamento iriam se acumulando.

¹¹ Se o intervalo fosse $[-\pi, \pi]$ o critério seria obedecido ao se decompor a função em série de Fourier, cuja base é também ortogonal e completa. [Arfken & Weber, 1995, p.810]

Na forma matricial de **dimensões finitas**, pode-se escrever: $f(u) = C_L \cdot P_L$, onde $C_L = [c_{L,0}, c_{L,1}, \dots, c_{L,n}]$ e $P_L = P_L(u)$ é o vetor coluna $P_L = [P_0, P_1, \dots, P_n]^T$, obtendo-se o escalar¹² $f(u)$, cujo valor depende de u .

Logo, pode-se afirmar que:

Eq. { 22} $P_L = L \cdot B_C$, onde $L = L_{n+1,n+1}$ é obtido pela Eq. {13} e B_C é a matriz coluna da base canônica: $B_C = [x^0, x^1, \dots, x^n]^T$.

Considerando-se que $C_C = [c_{C,0}, c_{C,1}, \dots, c_{C,n}]$ seja o vetor dos coeficientes na base canônica, impõe-se: $C_L P_L = C_C B_C = \text{escalar}$.

Considerando-se a Eq. {22}, obtêm-se: $C_L (LB_C) = C_C B_C$.

Como a transposta de um escalar é o próprio escalar, obtêm-se:

$$\begin{aligned} C_L LB_C &= C_C B_C \\ (LB_C)^T C_L^T &= C_C B_C \\ B_C^T (L^T C_L^T) &= B_C^T C_C^T \end{aligned}$$

Como todos os elementos de B_C^T são não nulos, para que a igualdade seja verdadeira é necessário que:

Eq. { 23} $L^T C_L^T = C_C^T$.

A matriz L é não-singular, logo, é válida também a relação:

Eq. { 24} $C_L^T = (L^T)^{-1} C_C^T$, que faz a transformação inversa.

Como exemplo, apresentam-se as matrizes para séries de ordem 4 (5 coeficientes), que fazem a transformação da base de Legendre para a Canônica, e a respectiva inversa.

¹²Sem esquecer que, no espaço infinito-dimensional de Hilbert, $f(u)$ é um vetor.

$$C_C^T = \begin{bmatrix} 1 & 0 & \frac{-1}{2} & 0 & \frac{3}{8} \\ 0 & 1 & 0 & \frac{-3}{2} & 0 \\ 0 & 0 & \frac{3}{2} & 0 & \frac{-15}{4} \\ 0 & 0 & 0 & \frac{5}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{35}{8} \end{bmatrix} \cdot C_L^T \text{ e}$$

$$C_L^T = \begin{bmatrix} 1 & 0 & \frac{1}{3} & 0 & \frac{1}{5} \\ 0 & 1 & 0 & \frac{3}{5} & 0 \\ 0 & 0 & \frac{2}{3} & 0 & \frac{4}{7} \\ 0 & 0 & 0 & \frac{2}{5} & 0 \\ 0 & 0 & 0 & 0 & \frac{8}{35} \end{bmatrix} \cdot C_C^T$$

3.4 Base ortonormal de Fourier

Em várias situações computacionais, principalmente em fenômenos periódicos, utiliza-se a expansão da função em séries de Fourier.

Sendo $f^*(u)$ uma aproximação por expansão em série de Fourier de $f(x)$, no intervalo¹³ $[-\pi, \pi]$ tem-se: $f^*(u) = \frac{a_0}{2} + \sum_1^k (a_k \cos ku + b_k \sin ku)$, expressão em que:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(u) \cdot du, \quad a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(u) \cdot \cos(ku) \cdot du \text{ e}$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cdot \sin(ku) \cdot du, \text{ com } k \in \mathbb{N}^*, \text{ conforme Boas [1983, pp. 309-310]. A}$$

base de Fourier, na forma que estão definidos os coeficientes, é normalizada, pois

¹³ Com maior generalidade, o intervalo pode ser $[a, b]$ a e $b \in \mathbb{R}$, com uma simples transposição de domínio por mudança de variável, como detalhado na secção 3.7.

$\frac{1}{\pi} \int_{-\pi}^{\pi} \cos^2(ku) \cdot du = \frac{1}{\pi} \int_{-\pi}^{\pi} \sin^2(ku) \cdot du = 1$. Além disso, como visto pelo cálculo dos coeficientes, a função peso é unitária. Conforme exposto na secção 3.2, essa base também obedece ao critério dos *MQ*.

3.4.1 Derivada de uma função expressa na base de Fourier

Seja $f(u)$ uma função expressa por uma combinação linear de Fourier:

$$\text{Eq. \{ 25 \}} \quad f(u) = \frac{a_0}{2} + \sum_1^k (a_k \cos ku + b_k \sin ku).$$

A pergunta, em termos da própria base de Fourier, é: — Quem é a derivada de f ?

Se a expansão de Fourier for até a ordem k , a base tem $2k + 2$ elementos.

Sendo B um elemento qualquer dessa base, tem-se:

$$\text{Eq. \{ 26 \}} \quad \frac{d}{du} B_{2n} = -B_{2n+1} \cdot n \quad \text{e} \quad \frac{d}{du} B_{2n+1} = B_{2n} \cdot n, \quad \text{com } n = 0, 1, \dots, k.$$

3.5 Matrizes de integração e diferenciação (M_I e M_D)

Nas secções anteriores, este capítulo mostrou que é possível escrever a derivada ou a integral de um elemento da base completa — não necessariamente ortogonal — como combinação linear dos outros elementos da base. Dessa forma, estabelecida essa combinação linear, ela pode ser usada na forma matricial para diferenciar ou integrar uma função expressa pelos seus coeficientes em relação a uma base, utilizando-se as matrizes de mudança de base.

Teorema da matriz operacional¹⁴

Seja $Z = Z_{n+1, n+1}$ a matriz quadrada que descreve os coeficientes resultantes de uma operação linear qualquer α na base genérica $B_G = [b_0, b_1, \dots, b_n]^T$ em função da própria base e $f(u)$ uma série cujo vetor de coeficientes é $C = [c_0, c_1, \dots, c_n]$, e sendo $V = [v_0, v_1, \dots, v_n]$ o vetor dos coeficientes resultantes da operação linear α aplicada à série, então:

$$V^T = Z^T \cdot C^T$$

Prova

Considerando $f(u) = \sum_{n=0}^k c_n B_{Gn}(u)$, como a operação é linear, ela pode ser inserida no somatório, isto é: $\alpha[f(u)] = \sum_{n=0}^k c_n \alpha B_{Gn}(u)$. Observe-se ainda que, considerando o espaço das matrizes finitas $M_{i,j}$, $i \in \{1, 2, \dots, n+1\}$ e $j \in \{1, 2, \dots, n+1\}$, $\alpha[f(u)]$ é um escalar¹⁵ cujo valor depende de u .

A operação aplicada à série é: $\alpha(C \cdot B_G) = C \alpha B_G$, pelo exposto no parágrafo anterior. Como $\alpha B_G = Z \cdot B_G$, pode-se escrever:

$$V \cdot B_G = C(ZB_G) = \text{escalar}.$$

Como a transposta de um escalar é o próprio escalar, obtém-se:

$$V \cdot B_G = C(ZB_G)$$

$$B_G^T \cdot V^T = (ZB_G)^T C^T$$

$$B_G^T \cdot V^T = B_G^T (Z^T C^T)$$

Como B_G^T é uma base genérica e C é arbitrário, para que a igualdade acima seja sempre válida é necessário que $V^T = Z^T C^T$.

Desta forma, para construírem-se as matrizes das operações lineares integração ou diferenciação, a tarefa central é determinar a matriz Z , para em seguida transpô-la.

3.5.1 Base canônica

Os tópicos seguintes expõem como obter as matrizes de integração e diferenciação na base canônica.

Matriz de integração

Se um polinômio de grau n está escrito na base canônica, da menor potência para a maior, tem-se:

$$Q(u) = a_0 + a_1 u + \dots + a_n u^n + 0 \cdot u^{n+1}.$$

¹⁴ Originalmente, o teorema foi deduzido nesta tese somente para as operações de integração e diferenciação. Meus agradecimentos ao Prof. José Roberto Castilho Piqueira, pela observação de que a única hipótese que estava sendo usada era que o operador fosse linear, ou seja, que o teorema tem validade mais geral. Por exemplo, a mudança de base obtida na secção 3.3, que é uma operação linear, se torna uma operação imediata com o uso deste teorema.

¹⁵ Sem deixar de ser, no espaço de Hilbert, um vetor.

O acréscimo do termo com u^{n+1} é necessário para que o polinômio a ser integrado tenha o mesmo número de coeficientes que a sua integral.

A integral do polinômio é:

$$\int_0^u Q(\xi) d\xi = 0 + a_0 \cdot u + a_1 \frac{u^2}{2} + \dots + a_n \frac{u^{n+1}}{n+1}.$$

Sendo o vetor linha $C = (a_i)$, $i = 0, \dots, n$, deseja-se obter M_{CI} , tal que:

$$\beta \cdot B_C = \int_0^u Q(\xi) d\xi, \text{ onde } \beta = [\beta_0, \beta_1, \dots, \beta_{n+1}] \text{ são os coeficientes da série obtidos}$$

após a integração, expressa na própria base canônica, e B_C é o vetor coluna da base canônica,

isto é, $B_C = \begin{bmatrix} x^0 \\ x \\ \vdots \\ x^{n+1} \end{bmatrix}.$

Monta-se a matriz Z , de forma a representar as integrais da base em função da própria base:

$$\int_0^u \begin{bmatrix} u^0 \\ u \\ \vdots \\ 0 \cdot u^{n+1} \end{bmatrix} \cdot dx = Z \cdot \begin{bmatrix} u^0 \\ u \\ \vdots \\ u^{n+1} \end{bmatrix}, \text{ com } \begin{cases} \int_0^u \xi^0 \cdot d\xi = u, \\ \dots, \\ \int_0^u \xi^n \cdot d\xi = \frac{1}{n+1} u^{n+1} \end{cases}.$$

$$\text{Assim, } \int_0^u Q(\xi) d\xi = C \cdot (Z \cdot B_C), \text{ de forma que:}$$

$Z_{i,i+1} = \frac{1}{i}$, $i = 1, 2, \dots, n$ sendo $Z_{n+1,n+1} = \frac{1}{n+1}$ e $Z_{i,j} = 0$ para todos os outros elementos dessa matriz.

Pelo teorema da matriz operacional $\beta^T = Z^T \cdot C^T$, ou seja:

$$\text{Eq. \{ 27\} } \quad M_{CI} = Z^T.$$

O código para gerar a matriz M_{CI} é apresentado a seguir. A transposição de Z é feita no próprio código, trocando-se o índice da linha pelo da coluna.

```

% Integração na base canônica.
function c_int = Can_Int(c)
%% Garante que c é vetor coluna
c = c(:); n = length(c); c = [c; 0];

%% Pré-aloca e define M
M = zeros (n+1, n+1);
for k0 = 1:n
    M(k0+1, k0) = 1/k0;
end
M(n+1, n+1)=1/(n+1);
M=sym(M); end
c_int = (M*c).'; % A saída é um vetor linha.

%% Fim

```

Figura 3. 2. Código em Matlab para obtenção da Matriz de Integração na base canônica.

Matriz de diferenciação

Para derivar, pode-se seguir o mesmo raciocínio da integração, isto é, procurando-se escrever a derivada do polinômio na base canônica, em função da própria base, conforme exposto a seguir.

A derivada do polinômio é:

$$\frac{d[Q(x)]}{dx} = 0 + a_1 \cdot x + 2 \cdot a_2 x^2 + \dots + n \cdot a_n x^{n-1}.$$

Sendo $C = (a_i)$, $i = 0, \dots, n$, deseja obter M_{CD} , tal que:

$$\frac{d[Q(x)]}{dx} = (M_{CD} \cdot C^T)^T \cdot B_C.$$

Monta-se a matriz Z , de forma a representar as derivadas em função da própria base:

$$\frac{d}{dx} \begin{bmatrix} x^0 \\ x \\ \vdots \\ x^n \end{bmatrix} = Z \cdot \begin{bmatrix} x^0 \\ x \\ \vdots \\ x^n \end{bmatrix}, \text{ com } \begin{cases} \frac{d}{dx}[x^0] = 0, \\ \dots, \\ \frac{d}{dx}[x^n] = n \cdot x^{n-1} \end{cases}.$$

Assim, $\frac{d[Q(x)]}{dx} = C \cdot (Z \cdot B_C)$, de forma que:

$$\text{Eq. \{ 28\} } \quad Z_{i,i-1} = i - 1, \quad i = 2, \dots, n-1.$$

Todos os outros elementos dessa matriz são nulos.

Pelo teorema da matriz operacional:

$$\text{Eq. \{ 29\}} \quad M_{CD} = Z^T.$$

Tanto a matriz de integração quanto a de diferenciação são singulares, isto é, não inversíveis. Isto pode ser entendido pelo fato de a integração ter um limite inferior, de certa forma, arbitrário, e a de diferenciação corresponder à derivada de vários polinômios que diferem por uma constante. Mas, mesmo assim, é interessante observar-se o produto delas:

$M_{CI} \cdot M_{CD} = F$, tal que $F_{1,1} = 0$ e $F_{j,j} = 1$, ou seja, a matriz F difere da identidade apenas pelo elemento $F_{1,1} = 0$, que seria a constante arbitrária de integração.

3.5.2 Base de Legendre

As propriedades dos polinômios de Legendre permitirão que a integral e a derivada de uma função expressa por uma série desses polinômios possa ser escrita em função da própria base de Legendre.

Matriz de Integração (M_{LI})

A secção 3.3.2 demonstra que a integral de um polinômio de Legendre pode ser expressa também na base de Legendre, pela expressão:

$$\int_{-1}^u P_n(\xi) \cdot d\xi = \frac{P_{n+1}(u) - P_{n-1}(u)}{2n+1}.$$

A obtenção da Matriz de Integração será feita de duas formas, para se verificar a consistência dos resultados.

Primeira forma

Sendo $f(u) = \sum_{k=0}^n c_k P_k$, com a relação de recorrência obtida, a função pode então

ser integrada diretamente, termo a termo, por: $\int f du = \sum_{k=0}^n c_k \int P_k$.

Genericamente, para os termos centrais do somatório (K_k , tal que $0 \neq k \neq n$),

$$\text{obtem-se: } K_k = c_k \int P_k = c_k \frac{P_{k+1} - P_{k-1}}{2k+1}.$$

O polinômio de Legendre de ordem k só aparecerá nessa integração para K_{k-1} e K_{k+1} , pois, pela Eq. {11}, ele só depende do sucessor e do antecessor. Pode-se então determinar qual será seu coeficiente (b_k), tendo-se em conta somente K_{k-1} e K_{k+1} .

No sucessor, ter-se-á:

$$K_{k+1} = c_{k+1} \frac{P_{k+2} - P_k}{2(k+1)+1} = c_{k+1} \frac{P_{k+2} - P_k}{2k+3}.$$

Ou seja, o coeficiente da parcela correspondente à P_k é: $\frac{-c_{k+1}}{2k+3}$.

Para o antecessor, ter-se-á:

$$K_{k-1} = c_{k-1} \frac{P_k - P_{k-2}}{2(k-1)+1} = c_{k-1} \frac{P_k - P_{k-2}}{2k-1}.$$

Ou seja, o coeficiente da parcela correspondente à P_k é: $\frac{c_{k-1}}{2k-1}$.

Assim:

$$\text{Eq. \{ 30\} } \quad b_k = \frac{c_{k-1}}{2k-1} - \frac{c_{k+1}}{2k+3}$$

Para o coeficiente extremo superior, a Eq. {30} continua válida, observando-se que $c_{n+1} = 0$.

Para obter-se b_0 , pode-se efetuar: $K_0 = c_0 \int P_0 = \frac{c_0}{1} u + \frac{c_0}{1} = c_0 \left(\frac{P_1}{1} + P_0 \right)$ e

$K_1 = c_1 \int P_1 = c_1 \left(\frac{P_2 - P_0}{2+1} \right)$, que corresponde a considerar-se $P_{-1} = -1$, que mantém a

alternância nos valores de $P_k(-1) = (-1)^k$.

$$\text{Portanto, } b_0 = \frac{c_0}{1} - \frac{c_1}{3}.$$

O sistema linear que descreve todas essas operações é, portanto:

$$\begin{aligned}
1 \cdot c_0 + \left(\frac{-1}{3}\right)c_1 + \dots 0 \dots \dots \dots &= b_0 \\
\frac{1}{1} \cdot c_0 + 0 \cdot c_1 + \left(\frac{-1}{5}\right)c_2 + 0 \dots \dots \dots + 0 \cdot c_{n+1} &= b_1 \\
0 \dots \frac{1}{2k-1} \cdot c_{k-1} + 0 \cdot c_k - \frac{1}{2k+3} + 0 \dots + 0 \cdot c_{n+1} &= b_k \\
0 \dots \dots \dots \frac{1}{2n+1} c_k + 0 \cdot c_{n+1} &= b_{n+1}
\end{aligned}$$

Na forma matricial¹⁶, pode-se escrever [Razzaghi & Yousefi, 2001]:

$$\begin{bmatrix}
1 & \frac{-1}{3} & \dots & 0 \dots \dots \\
\frac{1}{1} & 0 & \frac{-1}{5} & \dots & 0 \dots \dots \\
0 \dots & \dots \frac{1}{2k-1} & 0 & \frac{-1}{2k+3} \dots & 0 \dots \\
\vdots & & & & \\
\dots & \dots & \dots & 0 \dots \dots & \dots \\
0 \dots \dots \dots & \dots & \frac{1}{2n+1} & 0 & \dots
\end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n+1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n+1} \end{bmatrix}$$

Sendo $B = [b_0, b_1, \dots, b_{k+1}]$, obtém-se, pela matriz apresentada acima:

$$M_{LI} \cdot C^T = B^T.$$

Segunda forma: recorrência da integração

Pela Eq. {17} têm-se: $\int_{-1}^x P_n(\xi) \cdot d\xi = \frac{P_{n+1}(u) - P_{n-1}(u)}{2n+1} \quad n \in \mathbb{N}^*, \text{ e,}$

pela Eq. {18}, $\int_{-1}^u P_0(\xi) \cdot d\xi = P_1(u) + P_0.$

Com essas relações, monta-se a matriz Z , de forma que:

¹⁶ Razzaghi & Yousefi [2001] desenvolveram uma matriz similar para Legendre, mas utilizando polinômios normalizados no intervalo [0,1]. Além disso, Babolian & Fattahzadeh [2007] usaram a mesma idéia, mas com polinômios de Chebishev, obtendo aproximações ligeiramente melhores, por motivos que serão discutidos no capítulo 6 – Resultados e Análises.

$$\int_{-1}^x \begin{bmatrix} P_0 \\ P_1 \\ \dots \\ 0.P_{n+1} \end{bmatrix} = Z \cdot \begin{bmatrix} P_0 \\ P_1 \\ \dots \\ P_{n+1} \end{bmatrix}.$$

A matriz Z é a matriz quadrada $Z_{n+2,n+2}$, tal que:

$$Z_{1,1} = 1, Z_{1,2} = 1, \text{ para obedecer à Eq. \{18\},}$$

$$Z_{k,k+1} = \frac{-1}{2k-1}, Z_{k,k-1} = \frac{1}{2k-1} \quad k = 2, 3, \dots, n+1, \text{ para obedecer à Eq. \{17\} até a}$$

$$\text{penúltima linha da matriz e } Z_{n+2,n+1} = \frac{-1}{2(n+2)-1} = \frac{-1}{2n+3}, \text{ também para obedecer à Eq.}$$

\{17\}, observando-se que não existe o elemento $Z_{n+2,n+3}$, pois P_{n+1} não foi integrado.

Explicitamente, como exemplo, apresenta-se a matriz de integração para uma série de Legendre até ordem 3 (Z^T), portanto, com 4 coeficientes, que após a integração resultará em um vetor de coeficientes com 5 elementos ($n = 4$).

$$M_{LI} = \begin{bmatrix} 1 & -1/3 & 0 & 0 & 0 \\ 1 & 0 & -1/5 & 0 & 0 \\ 0 & 1/3 & 0 & -1/7 & 0 \\ 0 & 0 & 1/5 & 0 & -1/9 \\ 0 & 0 & 0 & 1/7 & 0 \end{bmatrix}$$

Matriz de Diferenciação

Esta secção apresenta duas formas de se obter a matriz de diferenciação, com a intenção de assegurar a validade dos resultados.

Primeira forma: matriz inversa

A matriz operacional de integração de Legendre é inversível. Essa condição permite obter a derivada de uma função descrita pela soma de série de polinômios de

Legendre, usando-se a operação inversa. Sendo $f(u) = \sum_{n=0}^k c_n P_n(u)$, tem-se: $\frac{df}{du} = \sum_{n=0}^{k-1} d_n P_n$,

onde o vetor D é obtido com: $D^T = M_{LI}^{-1} \cdot C^T$.

Entretanto, houve uma condição imposta no processo de integração que precisa ser observada. Uma vez que $F(u) = \int_{-1}^u f(\xi) \cdot d\xi$, está imposto que $F(-1) = 0$, já que

$$F(-1) = \int_{-1}^{-1} f(\xi) \cdot d\xi, \text{ portanto } \sum_{k=0}^{n+1} c_k P_k(-1) = 0.$$

$$\text{Além disso, } P_k(-1) = (-1)^k, \text{ logo } \sum_{k=0}^{n+1} c_k \cdot (-1)^k = 0.$$

A derivada de $c_0 P_0$ é nula, qualquer que seja c_0 , já que $P_0 = 1 = \text{constante}$. Assim, aproveitando-se desse fato, pode-se (e deve-se) impor um c_0 modificado (c_0^M) de forma que $c_0^M + \sum_{k=1}^{n+1} c_k \cdot (-1)^k = 0$, isto é, $c_0^M = -\sum_{k=1}^{n+1} c_k \cdot (-1)^k$, sem alterar a derivada da função. Essa imposição deve ser feita ao vetor de entrada antes do cálculo da derivada com a inversa da matriz de integração.

Segunda forma: recorrência da derivada

A secção 3.3.1 mostra que, sendo $b_i, i = 0, \dots, n$ os coeficientes de uma série de Legendre da função $f(u)$, podem-se obter os coeficientes de Legendre da série que corresponde à derivada da função por: $c_k = \frac{(2k+1)}{2} [1 - (-1)^{k+n}]$, $k = 0, \dots, n-1$, isto é,

$$P'_n(u) = \sum_{k=0}^{n-1} \frac{(2k+1)}{2} \cdot [1 - (-1)^{k+n}] \cdot P_k(u).$$

É possível obter a matriz de diferenciação a partir desse último resultado.

Como efetuado na integração, $Z = Z_{n,n}$, tal que:

$$\text{Eq. \{ 31 \}} \quad Z_{i+1,j+1} = [1 - (-1)^{i+j}] \cdot \frac{2j+1}{2}, \text{ com } i = 1, \dots, n-1, j = 0, \dots, i.$$

Todos os outros elementos dessa matriz são nulos.

A matriz de diferenciação é obtida então por: $M_{LD} = Z^T$.

A transposição já é implementada no código com a troca dos índices.

```

function [MD clin] = Leg_Der(c);
%% Entradas
c = c(:); % Garante que c é vetor coluna
n = length(c);
%% Matriz MD - Já é a transposta de Z porque os índices estão trocados
MD = zeros(n,n);
for i0 = 1:n-1
    for j0 = 0:i0
        MD(j0+1,i0+1)=sym((1-(-1)^(i0+j0))*(2*j0+1)/2);
    end
end

clin = (MD*c).'; %O vetor de saída é vetor linha.
%% Verificar se a matriz M foi também pedida.
if nargin < 2
    MD = clin.';
end
%% F I M -

```

Figura 3. 3. Código em Matlab para obter a matriz de diferenciação na base de Legendre.

Como exemplo, apresenta-se a matriz de diferenciação para uma série com 5 coeficientes ($k = 4$).

$$M_{LD} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

É interessante notar que:

$$M_{LI} \cdot M_{LD} = \begin{bmatrix} 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ ou seja, é } \mathbf{quase} \text{ a matriz identidade.}$$

Exemplo

Como exemplo, considera-se no intervalo $[-1,1]$ a função:

$$f(u) = \sum_{n=0}^4 (n+1) \cdot P_n, \text{ ou seja, } f(u) = [1 \ 2 \ 3 \ 4 \ 5] \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

A transposição para base canônica pode ser feita com a matriz apresentada na secção 3.3.3, resultando em $C_C = [\frac{11}{8}, -4, \frac{-57}{4}, 10, \frac{175}{8}]$, isto é:

$$f(u) = \frac{11}{8} - 4u - \frac{57}{4}u^2 + 10u^3 + \frac{175}{8}u^4.$$

A integral dessa função é, então: $\int_{-1}^u f = \sum_{n=0}^{k+1} b_n P_n$.

Como $B^T = M_{LI} \cdot C^T$, obtém-se:

$$B_L = [\frac{1}{3}, \frac{2}{5}, \frac{2}{21}, \frac{2}{45}, \frac{4}{7}, \frac{5}{9}], \text{ que na forma canônica é:}$$

$$B_C = [\frac{1}{2}, \frac{11}{8}, -2, \frac{-19}{4}, \frac{5}{2}, \frac{35}{8}].$$

Para derivar $f(u)$ usando-se a inversa da matriz de integração, deve-se impor

$c_0^M = -\sum_{k=1}^{n+1} c_k \cdot (-1)^k$, o que não altera a derivada, conforme descrito no tópico **Matriz de**

Diferenciação, logo, $c_0^M = -(-2 + 3 - 4 + 5) = -2$ e obtém-se:

$$D_L^T = M_{LI}^{-1} \cdot \begin{bmatrix} -2 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}, \text{ isto é, } D_L = [6, 24, 20, 35],$$

ou, na forma canônica: $D_C = [-4, -57/2, 30, 175/2]$.

3.5.3 Base de Fourier

Nas secções anteriores obteve-se, para determinadas bases, a matriz de diferenciação e a de integração. Este tópico procura fazer o mesmo com a base de Fourier. A expansão de uma função em série de Fourier é $f(u) = \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos ku + b_k \sin ku)$, que explicitamente resulta em:

$$f(u) = [a_0, b_0, a_1, b_1, \dots, a_n, b_n] \cdot \begin{bmatrix} 1/2 \\ \sin(0 \cdot u) \\ \cos(1 \cdot u) \\ \sin(1 \cdot u) \\ \dots \\ \cos(nu) \\ \sin(nu) \end{bmatrix},$$

Matriz de diferenciação

Para obter a matriz de diferenciação de uma função expressa na base de Fourier com termos até ordem k , basta observar o resultado da secção 3.4.1.

Sendo C o vetor linha dos coeficientes da série de Fourier, o que se busca é:

$$\frac{df}{dx} = M_{FD} \cdot C^T, \text{ onde } M_{FD} = M_{FD(2n+2, 2n+2)}.$$

Se a expansão de Fourier for até a ordem n , a base tem $2n+2$ elementos.

Sendo B_k um elemento qualquer dessa base, tem-se:

$$\text{Eq. \{ 32\}} \quad \frac{d}{dx} B_{2k} = -B_{2k+1} \cdot n \quad \text{e} \quad \frac{d}{dx} B_{2k+1} = B_{2k} \cdot n, \text{ com } k = 0, 1, \dots, n.$$

Dessa forma, todas as derivadas da base podem ser escritas em função da própria base, com a matriz Z .

$$\frac{dB}{dx} = Z \cdot B, \text{ em que } Z = Z_{2n+2, 2n+2}, \text{ com}$$

$$\text{Eq. \{ 33\}} \quad Z_{2k+1, 2k+2} = -k \quad \text{e} \quad Z_{2k+2, 2k+1} = k, \quad k = 0, 1, \dots, n.$$

O que se deseja é o vetor dos coeficientes de Fourier da derivada pelo produto matricial: $D^T = M_{FD} \cdot C^T$.

Pelo Teorema da Matriz Operacional, têm-se: $M_{FD} = Z^T$.


```

% Função que deriva uma função expressa pelos coeficientes
% de Fourier
function [M clin] = Fou_Der(c)
%% Dados
n = (length(c)-2)/2;
c = c(:); %Garante que c é vetor coluna.

%%
M = zeros(2*n+2);
for k = 0:n

M(2*k +1,2*k+2)=-k;
M(2*k+2,2*k +1) =k;

end

%% Transpor -
M = M.';
clin = (M*c).';
%% F I M -

```

Figura 3. 4. Código em Matlab para obter a Matriz de Diferenciação na base de Fourier.

Matriz de integração

Considere-se a função a ser integrada como $f(u) = \frac{a_0}{2} + \sum_{k=1}^n (a_k \cos ku + b_k \sin ku)$,

que explicitamente é:

$$f(x) = [a_0, b_0, a_1, b_1, \dots, a_k, b_k] \cdot \begin{bmatrix} 1/2 \\ \sin(0 \cdot x) \\ \cos(1 \cdot x) \\ \sin(1 \cdot x) \\ \dots \\ \cos(kx) \\ \sin(kx) \end{bmatrix}.$$

Analogamente ao que foi feito com outras bases, pode-se tentar integrar a função termo a termo. As integrais dos senos e co-senos serão expressas, sem problemas, em termos de senos e co-senos também. No entanto, a integral de $\frac{1}{2}$, o primeiro elemento da base, é

$\int_0^x \frac{1}{2} dx = \frac{x}{2}$. Não há como expressar esse resultado como uma combinação finita de senos e co-

senos e, portanto, para essa base, não há como se obter a matriz de integração, sem o truncamento da série. Sendo assim, para esse propósito, não se mostra conveniente o uso da série de Fourier.

3.6 Relação entre os coeficientes e as condições iniciais

Uma exigência que este trabalho considerou importante na evolução das soluções polinomiais das *EDs* foi a preservação das condições iniciais durante o processo evolutivo.

Os tópicos seguintes apresentam como isso pode ser feito quando se usa a base canônica e a de Legendre.

3.6.1 Condição inicial – base canônica

Esta pesquisa começou trabalhando com a forma mais comum de se expressar um polinômio, qual seja, a base canônica:

$$Q(u) = a_0 + a_1u + a_2u^2 \dots + a_nu^n.$$

Sendo u a variável independente de uma *ED*, para garantir que a influência das potências de maior grau fosse cada vez menor, escolheu-se um intervalo de trabalho em que ficasse obedecida a condição: $\lim_{n \rightarrow \infty} u^n < K$. Para tanto, é necessário e suficiente que $|u| \leq 1$.

Assim, qualquer subintervalo $[c; d]$ de $[-1; 1]$ poderia ser objeto da escolha.

Sem impor essa condição, não temos como garantir que o truncamento da série não seja desastroso, pois os termos de maior grau poderiam ser mais significativos que os de menor grau.

A imposição das *CI* fica relativamente simples quando se utiliza em conjunto com a base canônica um subintervalo de $[-1; 1]$ que comece em $u = 0$, pois, como

$Q(u) = a_0 + a_1u + a_2u^2 \dots + a_nu^n$, obtemos:

$$Q(0) = a_0, \quad Q^{(1)}(0) = a_1, \quad Q^{(2)}(0) = 2 \cdot a_2, \quad Q^{(3)}(0) = 3 \cdot 2 \cdot 1 \cdot a_3, \quad \text{e, genericamente,}$$

$$Q^{(n)}(0) = n!a_n = y^{*(n)}(0), \text{ ou seja:}$$

$$\text{Eq. \{ 34\} } \quad a_n = \frac{y^{*(n)}(0)}{n!} \quad \text{para } n = 0 \text{ até a ordem da } ED.$$

Durante o processo evolutivo, para que as condições iniciais se mantenham, é, pois, necessário que $a_n = \frac{y^{*(n)}(0)}{n!}$, ou seja, se a *ED* for de ordem n , os n primeiros coeficientes do polinômio ficarão fixados, obedecendo à Eq. {34}.

A utilização dessa base (canônica) não se revelou adequada. A evolução é muito lenta. Como não é uma base ortogonal, alterações em um coeficiente vão afetar os outros e, à medida que cresce o grau do polinômio utilizado, esse aspecto vai-se revelando cada vez mais grave.

No entanto, como mostra a Eq. {34}, a fixação das *CI* nessa base, com o subintervalo começando em $u = 0$, é simples e direta.

O problema muda um pouco de aspecto quando se trabalha em outro subintervalo de $[-1;1]$, por exemplo, $[c, d]$, que não começa em $u = 0$.

Como $Q(u) = a_0 + a_1u + a_2u^2 \dots + a_nu^n$, têm-se:

$$Q(c) = a_0 + a_1c + a_2c^2 \dots + a_nc^n = y_0(1) \text{ e}$$

$$Q^{(1)}(c) = a_1 + 2 \cdot a_2c \dots + n \cdot a_nc^{n-1} = y_0(2).$$

De qualquer forma, outras propriedades e a dedução do intervalo de busca de coeficientes melhores para essa base são apresentadas na secção 3 do capítulo 3 deste trabalho.

A secção 4 do capítulo 3 discute tanto a questão das condições iniciais, como as propriedades e a matriz de integração e derivação quando se usa como base os polinômios de Legendre. Essas matrizes vão determinar em que intervalo se dará a busca por coeficientes melhores, o que responde à questão 4, qual seja, como preservar as condições iniciais no processo evolutivo.

3.6.2 Condição inicial – base de Legendre

Usando-se os polinômios de Legendre, a manutenção das condições iniciais impostas depende de uma adequada combinação dos coeficientes c_ℓ que deve ser mantida ao longo do processo de evolução. Essa combinação leva a um sistema linear possível e indeterminado.

Para obter os coeficientes das derivadas em função dos coeficientes da função, usa-se a matriz de diferenciação.

Por exemplo, trabalhando com 7 coeficientes em uma equação qualquer de 3ª ordem, no intervalo $[-1,1]$, obtêm-se, após a diagonalização do sistema:

$$c_1 = 5c_4 - 21c_5 + 56c_6 - 120c_7 - \frac{2}{3}y_0^{(2)} - y_0^{(1)} - y_0^{(0)}$$

$$c_2 = 9c_4 - 35c_5 + 90c_6 - 189c_7 - y_0^{(2)} - y_0^{(1)}$$

$$c_3 = 5c_4 - 15c_5 + 35c_6 - 70c_7 - \frac{1}{3}y_0^{(2)},$$

sendo $\vec{y}_0 = [y_0, y_0^{(1)}, y_0^{(2)}]$ o vetor das condições iniciais.

A diagonalização é importante para que c_1 não dependa de c_2 e c_3 , c_2 , não dependa de c_1 e c_3 e também que c_3 não dependa de c_1 e c_2 , ou seja, para que o cálculo de qualquer um desses três elementos não afete os outros dois.

Assim, após a mutação ou *crossing-over* dos cromossomos, os valores de c_1 , c_2 e c_3 devem ser re-atribuídos de acordo com os outros novos coeficientes.

Sem a imposição do sistema linear, o polinômio evolui para erros menores, mas se afasta das condições iniciais e, em muitos casos, converge para a solução trivial $y = 0$, quando ela é também solução da ED.

Relação entre uma função e sua derivada, considerando condição inicial.

Vamos considerar a função: $F^*(x) = \sum_{k=0}^n b_k^* P_k$, definida no intervalo $[-1; 1]$,

com a condição $F^*(-1) = \alpha$.

Por outro lado, vamos considerar que $F(u) = \int_{-1}^u f(\xi) \cdot d\xi$, e ainda que $f(u)$ seja expressa por uma combinação de polinômios de Legendre, ou seja:

$$f(u) = \sum_{k=0}^{n-1} c_k P_k \text{ e, além disso, } F^*(u) - F(u) = \alpha.$$

É importante observar que: $F(-1) = \int_{-1}^{-1} f(u) \cdot du = 0$

As duas funções diferem apenas por uma constante e, portanto, têm a mesma derivada $f(u)$.

Parece imediato que, para que $F^*(-1) = \alpha$, basta acrescentar α à b_0 , fazendo-se $b_0^* = b_0 + \alpha$, que se consegue o objetivo; mas ocorre que todos os outros polinômios de Legendre de ordem par também têm termo independente, e essa suposição precisa ser provada.

Para obtermos os coeficientes da expansão em polinômios de Legendre da função $F^*(u)$, podemos avaliar as integrais:

$$b_n^* = \frac{2}{2n+1} \int_{-1}^1 F^*(u) \cdot P_n du = \frac{2}{2n+1} \int_{-1}^1 (F(u) + \alpha) P_n \cdot du.$$

Desta forma, obtêm-se:

$$b_n^* = \underbrace{\frac{2}{2n+1} \int_{-1}^1 F(u) \cdot P_n du}_{(I)} + \underbrace{\frac{2}{2n+1} \int_{-1}^1 \alpha \cdot P_n du}_{(II)}$$

A integral (I) representa os coeficientes b_n , da expansão em Legendre de $F(u)$.

Já a integral (II) só não se anula se $n = 0$.

Assim, para $n \neq 0$, temos $b_n^* = b_n$ e para $n = 0$:

$$b_0^* = b_0 + \frac{1}{2}(1 - (-1)) \cdot \alpha \Rightarrow b_0^* = b_0 + \alpha.$$

Consegue-se assim, dada uma condição inicial para a função e também a sua derivada pela expansão em Legendre, fazer-se a integração diretamente pela matriz, apenas acrescentando o valor inicial ao termo b_0 . Isso pode ser feito sucessivas vezes, dependendo da ordem da equação diferencial. Realizado esse procedimento, comparam-se então os coeficientes obtidos pela integração com os coeficientes calculados e pode-se avaliar a dispersão.

Como exemplo, vamos considerar a equação já discutida: $y'' + 4y' + 13y = 20e^{-x}$, com as condições iniciais: $y(0) = 1$ e $y'(0) = 3$, comparando os coeficientes para analisar a dispersão.

A tabela a seguir apresenta os coeficientes da expansão em Legendre, até 17º grau da função e os obtidos pela integração de suas derivadas, impondo-se as condições iniciais. A última linha é o desvio padrão simples¹⁷.

¹⁷ A integral do quadrado do polinômio de Legendre de ordem k é $2/(2k+1)$, portanto, a participação dos polinômios na composição da função vai diminuindo à medida que aumenta a sua ordem. Para uma comparação mais “justa” da dispersão, é interessante comparar o valor da dispersão em relação à raiz quadrada de $2/(2k+1)$, o que seria uma dispersão relativa. Isso será discutido no capítulo 5.

Tabela 3. 2. Coeficientes da série de Legendre da função e das derivadas integradas conforme as *CI* e os respectivos desvios padrão.

1º	2º	3º	4º	5º	6º
,641E-01	8,340E-01	,295E-02	,700E-01	3,099E-01	,328E-02
,641E-01	8,339E-01	,294E-02	,700E-01	3,099E-01	,328E-02
,640E-01	8,340E-01	,294E-02	,700E-01	3,099E-01	,328E-02
,58E-06	,218E-06	,27E-06	,71E-07	,095E-06	,54E-07
7º	8º	9º	10º	11º	12º
,117E-02	2,548E-02	,197E-03	,112E-04	4,364E-04	,050E-04
,117E-02	2,548E-02	,198E-03	,101E-04	4,363E-04	,053E-04
,117E-02	2,548E-02	,198E-03	,098E-04	4,361E-04	,053E-04
,34E-07	,583E-07	,7E-07	,08E-07	,067E-07	,55E-07
13º	14º	15º	16º	17º	18º
5,267E-06	2,699E-06	,299E-07	3,535E-07	1,365E-07	,073E-07
5,329E-06	2,587E-06	,445E-07	1,317E-07	,444E-08	,136E-08
5,376E-06	2,675E-06	,592E-07	5,302E-08	1,114E-10	,352E-09
,488E-08	,841E-08	,42E-08	,272E-07	,696E-08	,17E-08

3.7 Transposição de domínios nas EDs

Considerando que a *ED* seja definida em um intervalo $[a;b]$ qualquer, cuja variável independente seja x , e que se queira transpô-la para o intervalo $[c;d]$, na variável u , obtendo ED_t , em outro intervalo em que se possa expandir a função em série, e que a transformação $u = T(x)$ — linear ou não-linear — faça essa transposição, esta secção deduz como obter a equação diferencial transformada (ED_t) e as transformações das condições iniciais, transpondo o *PVI*, considerando até a quarta ordem. Isso não foi feito nas abordagens de *PVIs* com polinômios ortogonais, mesmo em publicações recentes¹⁸, sendo esta secção uma das contribuições deste trabalho.

3.7.1 Transposição Genérica

Embora essa transposição genérica aparente um aspecto por demais formal, ela é importante na análise de várias transformações particulares.

¹⁸ Ver, por exemplo, [Doha & Bhrawy, 2008], [Babolian & Fattahzadeh, 2007] e [Khellat & Yousefi, 2006].

A história da Física-Matemática tem demonstrado que, em vários casos, a solução do problema geral, às vezes, é mais direta, objetiva e obtível do que as de um conjunto de problemas particulares.

Sendo a equação diferencial: $ED(y, y^{(1)}, \dots, y^{(n)}, x) = 0$, no domínio $[a; b]$, e a transformação $u = T(x)$ que transpõe o domínio para $[c; d]$, de forma bijetora¹⁹ tal que $T(a) = c$ e $T(b) = d$, a pergunta é como se deve reescrever a ED e suas condições iniciais de forma a corresponderem ao problema original?

Transformação da equação

Em uma EDO comparecem a função, suas derivadas e a variável independente. Com a transposição de intervalo, impõe-se que: $y(x) = y_t(u)$, onde y_t é a função transformada para o intervalo de trabalho $[c; d]$, isto é, para transpor a ED , basta substituir y por y_t na transposição de domínio. Entretanto, essa substituição direta não é válida para as derivadas. Nos parágrafos seguintes, esta tese deduz como efetuar a substituição das derivadas, considerando o problema até derivadas de ordem 4.

Considerando que $u = T(x)$ seja o mapa genérico bijetor que leva x até u , como a função `tx2u.m` e `zi.m`²⁰ a sua inversa, o código, em Matlab, para transposição das funções até ordem 4 é:

```
%% Substituir a variável da equação por u.
syms x u;
c = subs(tx2u,x,a); d = subs(tx2u,x,b);
g = diff(zi); % Derivada da inversa.
f = 1/g; f = simple(f);

%% Modificar as derivadas.
EQ = subs(EQ,D1,D1*f); EQ = simple(EQ);
EQ = subs(EQ,D2,D2*f^2 + f*diff(f)*D1);
EQ = subs(EQ,D3,D3*f^3 + 3*D2*diff(f)*f^2 + ...
((diff(f))^2 + f*diff(f,2))*D1*f );
EQ = subs(EQ,D4,...
D4*f^4 + D3*6*f^3*diff(f) + ...
D2*(4*diff(f,2)*f^3 + 7*diff(f)^2*f^2) + ...
D1*f*(diff(f)^3 + 4*f*diff(f)*diff(f,2) + f^2*diff(f,3)) );

%% Modificar as Condições iniciais
f0 = subs(f,x,c);
f1 = subs(diff(f),x,c);
f2 = subs(diff(f,2),x,c);
```

¹⁹ Esta imposição é importante para garantir que não haja dois valores de u que correspondam ao mesmo x .

²⁰ Há a função `finverse.m` que faria isso automaticamente, mas nem sempre a estrutura simbólica consegue obter essa transformação. A maneira mais simples, aqui constatada, foi explicitá-la de antemão.

```

y0t(1) = y0(1);
y0t(2) = y0(2)/f0;
y0t(3) = (y0(3)-f0*f1*y0t(2))/f0.^2;
y0t(4) = (y0(4)-(3*y0t(3)*f1*f0.^2+...
(f0*f1.^2+f0.^2*f2)*y0t(2)))/f0.^3;
%% F I M

```

Figura 3. 5. Código em Matlab para efetuar a transposição de domínios a partir de uma transformação genérica (tx2u) e sua inversa (zi).

Para justificar esse código proposto, devem-se observar as conclusões seguintes.

Com relação à primeira derivada, tem-se: $y(x) = y_t(u)$. Logo, a 1ª derivada deve ser substituída conforme a igualdade:

$$y^{(1)}(x) = y_t^{(1)}(u) \cdot \frac{du}{dx}.$$

A derivada $\frac{du}{dx}$ é função de x , mas na equação transformada a variável é u . Como

a derivada da inversa é a inversa da derivada, é conveniente a substituição:

$$\frac{du}{dx} = \frac{1}{dx/du} = \lambda, \text{ assim, a primeira derivada será substituída conforme a igualdade:}$$

$$\text{Eq. \{ 35\} } \quad D_1 = D_{1t} \cdot \lambda.$$

Para a segunda derivada, observando-se as regras de diferenciação, obtém-se:

$$\frac{d(y^{(1)})}{dx} = \frac{d(y_t^{(1)} \cdot \lambda)}{du} \cdot \frac{du}{dx}, \text{ logo:}$$

$$\text{Eq. \{ 36\} } \quad D_2 = D_{2t} \cdot \lambda^2 + D_{1t} \cdot \frac{d\lambda}{du} \cdot \lambda.$$

Diferenciando a expressão acima, pode-se obter a transformação para a terceira derivada, como segue.

$$D_3 = \frac{d(D_2)}{dx} = \frac{d}{du} \left(D_{2t} \cdot \lambda^2 + D_{1t} \cdot \frac{d\lambda}{du} \cdot \lambda \right) \cdot \frac{du}{dx};$$

$$D_3 = \left(D_{3t} \cdot \lambda^2 + 2\lambda \frac{d\lambda}{du} D_{2t} + D_{2t} \cdot \frac{d\lambda}{du} \cdot \lambda + D_{1t} \cdot \frac{d^2\lambda}{d^2u} \cdot \lambda + D_{1t} \cdot \left(\frac{d\lambda}{du} \right)^2 \right) \cdot \lambda;$$

$$\text{Eq. \{ 37\} } \quad D_3 = D_{3t} \cdot \lambda^3 + 3D_{2t} \lambda^2 \frac{d\lambda}{du} + D_{1t} \lambda \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right).$$

A substituição da quarta derivada é trabalhosa, e o cálculo será feito separadamente para cada uma das parcelas; e depois os termos serão reagrupados e multiplicados por $\lambda = \frac{du}{dx}$.

$$D_4 = \frac{d(D_3)}{dx} = \frac{d}{du} \left(D_{3t} \cdot \lambda^3 + 3D_{2t}\lambda^2 \frac{d\lambda}{du} + D_{1t}\lambda \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) \right) \cdot \frac{du}{dx}$$

$$\frac{d(D_{3t} \cdot \lambda^3)}{du} = D_{4t}\lambda^3 + 3\lambda^2 \frac{d\lambda}{du} D_{3t}. \quad (1^a \text{ parcela})$$

$$\frac{d}{du} \left(3D_{2t}\lambda^2 \frac{d\lambda}{du} \right) = 3D_{3t}\lambda^2 \frac{d\lambda}{du} + 3 \cdot 2 \cdot \lambda \frac{d\lambda}{du} D_{2t} \frac{d\lambda}{du} + 3D_{2t}\lambda^2 \frac{d^2\lambda}{du^2}, \text{ isto é:}$$

$$\frac{d}{du} \left(3D_{2t}\lambda^2 \frac{d\lambda}{du} \right) = 3D_{3t}\lambda^2 \frac{d\lambda}{du} + 6D_{2t}\lambda \left(\frac{d\lambda}{du} \right)^2 + 3D_{2t}\lambda^2 \frac{d^2\lambda}{du^2} \quad (2^a \text{ parcela}).$$

$$\begin{aligned} \frac{d}{du} \left[D_{1t}\lambda \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) \right] &= D_{2t}\lambda \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) + D_{1t} \frac{d\lambda}{du} \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) + \dots \\ &+ \dots D_{1t}\lambda \frac{d}{du} \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) \end{aligned} \quad (3^a \text{ parcela}).$$

$$\text{Sendo que: } D_{1t}\lambda \frac{d}{du} \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{d^2u} \right) = D_{1t}\lambda \left(3 \frac{d\lambda}{du} \frac{d^2\lambda}{du^2} + \lambda \frac{d^3\lambda}{du^3} \right).$$

Agrupando-se os termos, obtêm-se:

Termo com D_{4t}

$$\text{Eq. \{ 38\} } \quad D_{4t}\lambda^4$$

Termos com D_{3t}

$$\text{Eq. \{ 39\} } \quad D_{3t} \left(3\lambda^2 \frac{d\lambda}{du} + 3\lambda^2 \frac{d\lambda}{du} \right) \lambda = 6D_{3t}\lambda^3 \frac{d\lambda}{du}.$$

Termos com D_{2t}

$$D_{2t} \left(6\lambda \left(\frac{d\lambda}{du} \right)^2 + 3\lambda^2 \frac{d^2\lambda}{du^2} + \lambda \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{du^2} \right) \right) \lambda =$$

$$\text{Eq. \{ 40\}} = D_{2t} \lambda^2 \left(7 \left(\frac{d\lambda}{du} \right)^2 + 4\lambda \frac{d^2\lambda}{du^2} \right).$$

Finalmente, os termos com D_{1t} . Após o agrupamento são três termos.

$$D_{1t} \left[\frac{d\lambda}{du} \left(\left(\frac{d\lambda}{du} \right)^2 + \lambda \frac{d^2\lambda}{du^2} \right) + \lambda \left(3 \frac{d\lambda}{du} \frac{d^2\lambda}{du^2} + \lambda \frac{d^3\lambda}{du^3} \right) \right] \lambda =$$

$$= D_{1t} \lambda \left[\left(\frac{d\lambda}{du} \right)^3 + \lambda \frac{d\lambda}{du} \frac{d^2\lambda}{du^2} + 3\lambda \frac{d\lambda}{du} \frac{d^2\lambda}{du^2} + \lambda^2 \frac{d^3\lambda}{du^3} \right] =$$

$$\text{Eq. \{ 41\}} = D_{1t} \lambda \left[\left(\frac{d\lambda}{du} \right)^3 + 4\lambda \frac{d\lambda}{du} \frac{d^2\lambda}{du^2} + \lambda^2 \frac{d^3\lambda}{du^3} \right].$$

Transformação das condições iniciais

Todas as transformações anteriores podem ser aproveitadas para a transposição das condições iniciais pensando agora na transformação de u para x , isto é, como $u = T(x)$, têm-se $x = T^{-1}(u)$ e $\frac{dx}{du} = \frac{1}{du/dx} = \rho$. Ou seja, aproveitam-se as fórmulas anteriores, substituindo-se λ por ρ e u por x .

Obtêm-se:

$$\text{Eq. \{ 42\}} \quad y_{0t} = y_0;$$

$$\text{Eq. \{ 43\}} \quad y_{1t} = y_1 \cdot \rho;$$

$$\text{Eq. \{ 44\}} \quad y_{2t} = y_2 \cdot \rho^2 + y_1 \cdot \frac{d\rho}{dx} \cdot \rho \text{ e}$$

$$\text{Eq. \{ 45\}} \quad y_{3t} = y_3 \cdot \rho^3 + 3y_2 \rho^2 \frac{d\rho}{dx} + y_1 \rho \left(\left(\frac{d\rho}{dx} \right)^2 + \rho \frac{d^2\rho}{dx^2} \right).$$

Alternativamente, como ao calcular y_{3t} o valor de y_{2t} já foi calculado, pode-se então exprimi-lo em função desse último, como foi feito no código.

3.7.2 Transposição Linear

Com uma transposição de domínio linear, do tipo $u = mx + n$, se a série no intervalo de trabalho for um polinômio, esse resultado será também um polinômio no domínio em que a função está definida.

Considerando que a função esteja definida no intervalo $[a, b]$ e que se queira transpô-la para o intervalo $[c, d]$, a igualdade $\frac{u - c}{d - c} = \frac{x - a}{b - a}$, tal que

$$u = T(x) = \left(\frac{d - c}{b - a} \right) x + \left(\frac{bc - ad}{b - a} \right), \text{ faz a transposição.}$$

Essa igualdade pode ser entendida geometricamente considerando-se semelhanças de triângulos, conforme ilustra a figura seguinte.

Figura 3. 6. A semelhança de triângulos permite obter: $\frac{u - c}{d - c} = \frac{x - a}{b - a}$.

Obtém-se $y(x) = y_t(u)$ e, para a 1ª derivada:

$$y^{(1)}(x) = y_t^{(1)}(u) \cdot \frac{du}{dx}.$$

Pondo-se $\frac{d - c}{b - a} = k$, vê-se que $T^{(1)}(x) = \frac{du}{dx} = k$ e, ainda,

$T^{(2)}(x) = 0$, ou seja,

$$y^{(1)}(x) = y_t^{(1)}(u) \cdot k$$

Para as derivadas de maior ordem, obtém-se:

$$y^{(2)}(x) = \frac{d}{dx} \left(y_t^{(1)}(u) \cdot T^{(1)}(x) \right) = y_t^{(2)} k^2, \text{ e, genericamente,}$$

$$\text{Eq. \{ 46\}} \quad y^{(n)}(x) = y_t^{(n)} k^n \Rightarrow y_t^{(n)} = \frac{y^{(n)}(x)}{k^n}.$$

Esta última igualdade pode ser aproveitada também para a transposição de todas as condições iniciais, considerando que $y^{(0)}$ seja a própria função.

3.7.3 Transposição trigonométrica

Considerando-se uma função na variável x , em um intervalo real qualquer, primeiramente a transpomos para um intervalo em que u seja a tangente de um ângulo (v) por uma transformação **não** linear.

Sendo $[a; b]$ o intervalo de origem, tem-se:

$$v = \text{atan}(x)$$

$$v_1 = \text{atan}(a) \text{ e } v_2 = \text{atan}(b).$$

Em seguida, o intervalo $[v_1; v_2]$ será transposto para o intervalo $[c, d]$ na variável u , pela transformação linear, entre u e v :

$$\frac{u - c}{d - c} = \frac{v - v_1}{v_2 - v_1} \Rightarrow$$

$$\text{Eq. \{ 47\}} \quad u = \frac{(d - c)(\text{atan}(x) - v_1)}{v_2 - v_1} + c$$

cuja inversa é:

$$\text{Eq. \{ 48\}} \quad x = \tan \left(\frac{(v_2 - v_1)(u - c)}{d - c} + v_1 \right).$$

3.7.4 Transposição Irracional

A transposição seguinte foi obtida a partir da idéia do seno do arco-tangente de um número real.

Considerando $u = \frac{x}{\sqrt{1+x^2}}$, verifica-se que, enquanto x varia de $-\infty$ a $+\infty$, o

valor de u fica compreendido entre -1 e 1 .

Gráfico 3. 4. Transposição não-linear que confina os valores de u entre -1 e 1 .

No caso de ser um intervalo limitado $[a,b]$ a ser transposto para o intervalo $[c,d]$

faz-se, analogamente à secção anterior: $v_1 = \lim_{x \rightarrow a} \frac{a}{\sqrt{1+a^2}}$, $v_2 = \lim_{x \rightarrow b} \frac{b}{\sqrt{1+b^2}}$ e ainda

$\frac{u-c}{d-c} = \frac{x-v_1}{v_2-v_1}$, concluindo a transformação, cuja forma final será:

$$\text{Eq. \{ 49\}} \quad u = T(x) = \frac{\left(\frac{x}{\sqrt{1+x^2}} - \frac{\frac{a}{\sqrt{1+a^2}} + \frac{b}{\sqrt{1+b^2}}}{(d-c)} \right)}{\frac{b}{\sqrt{1+b^2}} - \frac{a}{\sqrt{1+a^2}}}.$$

Sendo $p = \frac{b}{\sqrt{1+b^2}} - \frac{a}{\sqrt{1+a^2}}$ e $q = \frac{b}{\sqrt{1+b^2}} + \frac{a}{\sqrt{1+a^2}}$, a inversa pode ser escrita

na forma:

$$\text{Eq. \{ 50\}} \quad x = \frac{pu + q}{\sqrt{(d - c)^2 - (pu + q)^2}}.$$

3.8 Integração de Gauss-Legendre

Há muitas situações em que a integral de uma função não é expressável por combinação finita de funções elementares, como $\int_a^b e^{-x^2} dx$, ou $\int_a^b e^{\sin(x)} dx$, por exemplo. Nesses casos, lança-se mão de uma integração numérica.

Em geral, os processos de integração numérica avaliam os valores da função $f(x)$ em determinados pontos do intervalo $[a, b]$, x_i , e atribuem um “peso” (w_i) a esses valores para depois efetuarem um somatório, isto é, simbolicamente:

$$\int_a^b f(x) dx = \sum_{i=1}^n w_i f(x_i) + E, \text{ onde } E \text{ é o erro cometido na aproximação} \quad [\text{Wilson,}$$

Turcotte & Halpern, 2003].

Em muitos casos, a função $f(x)$ não é conhecida, mas apenas os seus valores em determinadas abscissas x_i , ao longo do intervalo, e a expressão acima continua aplicável.

Campos [2007, pp. 211-246] discute e compara os métodos numéricos de integração de Newton-Cotes, do qual o de Simpson é um caso particular, e a Quadratura de Gauss-Legendre (*QGL*), considerando o número de pontos, grau do polinômio de aproximação e complexidade computacional.

Os resultados mostram que a *QGL* é francamente superior [Campos, *ib.*, p. 246] em termos de precisão, com baixo custo computacional, e de programação relativamente simples.

O Matlab possui uma função residente para integração numérica (*quadl.m*), baseado no método adaptativo de Lobatto [The Mathworks, 2006], também de alta precisão, mas, como mostram Wilson, Turcotte & Halpern [2003, secção 5.3], a *QGL* ainda é ligeiramente superior na maioria dos casos por eles analisados.

Mesmo em pontos próximos de singularidades, a *QGL* se mostrou, com algumas adaptações, um método extremamente preciso [Lutz, 1992] nas aplicações em geral, até, por

exemplo, em cálculos de antenas, quando expressões analíticas integráveis não estão disponíveis [Duan, 1997].

Recentemente, Mughal, Ye & Iqbal [2006] discutiram a possibilidade de usar polinômios de Legendre de ordens mais altas, explorando a simplificação pelo uso de suas raízes e os avanços na precisão e velocidade dos processadores, mostrando os ganhos de precisão na *QGL*.

A *QGL* se baseia nos polinômios de Legendre. Se, por exemplo, se estipula que a quadratura deva ser feita com polinômios de Legendre de grau até 21, o resultado é exato para funções polinomiais grau até 41 ($2n-1$) [Wilson, Turcotte & Halpern, *ib.*, 5.2]. O algoritmo usado nesta tese provém dessa referência (secção 5.2), mas está minuciosamente detalhado em Campos [2007, pp. 234-236].

Se, para maior precisão, opta-se por analisar 126 pontos da função em determinado intervalo, usando-se a *QGL* com grau 21, então o intervalo deverá ser dividido em 6 subintervalos, como se procedeu em vários pontos deste trabalho, ao se usar a *QGL*, por exemplo, na análise dos cromossomos de determinada geração no capítulo 5, secção 1. Nesse caso, o algoritmo responderá com o valor exato da integral para funções polinomiais também até grau 41, mas que podem mudar de um intervalo para outro.

A título de exemplo, com os parâmetros do parágrafo anterior, analisa-se a integral de 3 funções e o erro percentual em relação ao valor exato, conforme tabela seguinte, no intervalo $[0,1]$.

Tabela 3.3. Precisão da *QGL* para diferentes funções

Função	Integral exata	Erro Percentual (%)
$f_1(x) = \sqrt{x}$	2/3	1e-4
$f_2(x) = \ln(x)$	-1	2e-2
$f_3(x) = e^{10x} \cos(10\pi x)$	202,7	5e-2

A escolha dessas funções [Wilson, Turcotte & Halpern, 2003] visa verificar a precisão da quadratura em casos de funções com singularidades na derivada (f_1), funções não limitadas (f_2) e funções altamente oscilantes (f_3) não periódicas. Pelas margens de erro observadas, vê-se que mesmo em casos singulares a *QGL* apresenta ótimos resultados.

Capítulo 4. Algoritmo Genético

Genericamente, no darwinismo, considera-se que a capacidade de sobrevivência dos seres vivos em seus ambientes seja o resultado de, basicamente, três fatores:

- seleção natural;
- hereditariedade;
- mutações.

A seleção natural é também conhecida como a luta pela sobrevivência. Sobrevivem e se reproduzem os mais aptos, geração a geração. Após várias gerações, a população remanescente reunirá o conjunto de características necessárias à sobrevivência no ambiente considerado. Conforme Darwin [1859], não foram os organismos que se adaptaram, mas sim o ambiente que se encarregou de eliminar os menos aptos.

Embora Darwin não tenha se atido à explicação da hereditariedade e das mutações, isso foi agregado mais tarde à Teoria da Evolução com a descoberta do DNA, dando origem ao que hoje se conhece como neodarwinismo.

Os três fatores da teoria de Darwin podem ser implementados em simulações computacionais com propósitos diversos como, por exemplo, resolver computacionalmente problemas cuja solução matemática seja inacessível, por algum motivo, mas para o qual uma população de seres vivos encontraria na prática uma resposta. Talvez não a melhor, já que não sabemos quando a evolução termina, talvez não a única, mas uma resposta. Os algoritmos computacionais que se utilizam desses fatores são os algoritmos genéticos.

A Computação Evolutiva, da qual o Algoritmo Genético (0) é um ramo, foi introduzida em 1960 por I. Rechenberg com seu trabalho “Estratégias de Evolução” (*Evolutionstrategy* no original). Mais tarde, John Holland, cujos trabalhos resultaram na publicação do livro “*Adaptation in Natural and Artificial Systems*” [Holland, 1975], introduziu o termo Algoritmo Genético [Krink, Thiemo & Paterlini, 2006, p.1227].

Mais tarde, em 1992, John Koza usou algoritmos genéticos para desenvolver programas para realizar certas tarefas. Ele chamou seu método de “Programação Genética” (PG) [Obitko, 1998].

A seção 5.1.1 do capítulo apresenta mais detalhes sobre esse método, suas variações, assim como a forma com que ele foi implementado na resolução das *EDs*.

4.1 Características dos algoritmos genéticos

Os tópicos a seguir irão detalhar como a inspiração que a Biologia trouxe para a computação será implementada nas otimizações.

4.1.1 Hereditariedade simulada

Nos organismos vivos reais a hereditariedade reside nos cromossomos. Há em cada célula de um mesmo ser vivo um mesmo conjunto de cromossomos. Os cromossomos, por sua vez, são conjuntos de genes que são blocos de DNA.

Cada gene codifica uma determinada proteína cujo conjunto irá resultar nas características totais do indivíduo. Por exemplo, no código genético de um humano normal já está fixado que ele terá dois braços, duas pernas etc. Há genes que codificam cor do cabelo, cor da pele, cor dos olhos, forma do queixo etc. e variam de indivíduo para indivíduo. Essa variedade resulta na diversidade que encontramos dentro de uma mesma espécie.

A posição de cada gene na estrutura cromossômica é chamada de *locus*. O conjunto completo de material genético, ou seja, todos os cromossomos, é chamado genoma. Por outro lado, conjuntos particulares de genes são chamados de genótipos. O resultado manifesto dos genótipos é chamado de fenótipo. No entanto, a correspondência não é biunívoca. Fenótipos iguais podem provir de genótipos diferentes.

Nas simulações computacionais, cada vetor (cromossomo) é um indivíduo da população, o *locus* é o índice do componente de um vetor, e os conteúdos dos *loci* são simulados por números, trechos de programação, sentenças condicionais etc., simulando assim os genes [Michalewicz, 1996].

A hereditariedade é então simulada tomando-se dois vetores (pai e mãe) que resultarão em um descendente. Esse descendente terá partes do pai e partes da mãe, mas será um vetor de mesma dimensão que os seus geradores. Tantos para os seres vivos reais como para os virtuais, os critérios para se determinar em cada *locus* se o gene ali alojado será paterno ou materno são diversos [Mitchell, 1997]. Alguns são descritos a seguir.

4.1.2 *Crossing-over*

Nesse processo toma-se a determinado *locus* um bloco inteiro de material genético de um dos pais, para em seguida terminá-lo com o material genético do outro. Nos seres vivos reais, a probabilidade de ocorrer o *cross-over* em determinado *locus* é diretamente proporcional à distância deste *locus* ao extremo do cromossomo [Obitko, 1998].

4.1.3 Elitismo

Esse procedimento consiste em conservar para a geração seguinte o melhor ou alguns melhores indivíduos da população. Isso evita pioras no melhor da população, mas pode, em alguns casos, levar o processamento para um mínimo local, atingindo uma estagnação evolutiva. Com isso, o processamento deixaria de procurar mínimos menores, fora da região em que a estagnação ocorreu [Obitko, 1998]. No entanto, a imposição do elitismo impede retrocessos no processo evolutivo e se revelou importante em várias aplicações [Krink, Thiemo & Paterlini, 2006, p.1227].

4.1.4 Seleção por roleta viciada

Se simplesmente impedimos que os indivíduos menos adaptados se reproduzam, podemos estar perdendo habilidades valiosas desses seres. Se, por outro lado, eles tiverem a mesma probabilidade de reprodução que os outros, estamos negando a seleção natural. No critério da elegibilidade para reprodução por roleta, os mais aptos têm maior probabilidade de serem sorteados, mas não é impossível que o menos apto se reproduza.

Vício da roleta

É muito freqüente, particularmente em computação evolutiva, postular-se determinada probabilidade de que um evento aconteça. Por exemplo, que a probabilidade de acasalamento seja maior para os melhores da população, ou que a probabilidade do *crossing-over* seja mais concentrada nos genes centrais.

Este tópico deduz uma relação entre a função de distribuição de probabilidades $g(x)$ e o gerador de números aleatórios do Matlab, entre 0 e 1, a função `rand.m`, com alguns exemplos que serão usados pelo *SAM* no processo evolutivo.

Dado um algoritmo que gere números aleatórios (r) uniformemente distribuídos entre 0 e 1, o valor gerado é a integral da distribuição $g(x) \geq 0$, ou seja:

$$r = \int_{-\infty}^x g(x) dx = G(x).$$

Dessa forma, para obter a função $g(x)$, basta observar que:

$$\text{Eq. \{ 51\}} \quad g(x) = \frac{dG}{dx} = \frac{dr}{dx}.$$

Sendo $x = T(r)$, obtém-se $r = T^{-1}(x)$, ou seja, as funções T e G são inversas.

Exemplos

1. Para sortear números aleatórios abrangendo todo o campo dos reais, foi proposta a função:

$$\text{Eq. \{ 52\}} \quad T(r) = \tan\left(\pi\left(r - \frac{1}{2}\right)\right) = x, \text{ com } 0 < r < 1.$$

$$\text{Tem-se, portanto, } r = \frac{\arctan(x)}{\pi} + \frac{1}{2}.$$

$$\text{Como } g(x) = \frac{dr}{dx}, \quad g(x) = \frac{d}{dx}\left(\frac{\arctan(x)}{\pi} + \frac{1}{2}\right)$$

$$\text{Eq. \{ 53\}} \quad g(x) = \frac{1}{\pi} \frac{1}{(1+x^2)}.$$

O histograma a seguir, gerado em Matlab 7, verifica a concordância dos resultados, com 100.000 amostragens.

Gráfico 4. 1. Histograma dos coeficientes gerados. A linha pontilhada é a função de distribuição calculada.

Neste caso, a escolha da função tem como objetivo gerar números aleatórios com maior concentração em torno do zero, para evitar coeficientes muito altos em polinômios aleatórios, quando estes forem o ponto de partida do processo evolutivo.

2. A probabilidade de o *crossing-over* ocorrer em determinado gene de um cromossomo é proporcional às distâncias que ele tem dos pontos extremos. Considerando que o cromossomo tenha n genes, pode-se então escrever:

$g(x) = \alpha \cdot (n - x)x$, onde α é a constante de proporcionalidade.

Como $\int_{-\infty}^{+\infty} g(x) dx = 1$ e a função é nula fora do intervalo entre os cromossomos, tem-se:

$$\int_0^n \alpha \cdot (n - x)x \cdot dx = 1, \text{ logo } \alpha \frac{n^3}{6} = 1 \Rightarrow \alpha = \frac{6}{n^3}.$$

$$\text{Portanto, } g(x) = \frac{6 \cdot (n - x)x}{n^3}.$$

Para obter-se a função geratriz dessa distribuição, faz-se:

$$\int_0^x \frac{6 \cdot (n - x)x}{n^3} dx = r \Rightarrow \frac{x^2(3n - 2x)}{n^3} = r.$$

$$\text{Dessa forma: } x = n \cdot \sin \left[\frac{\arcsin(2r - 1)}{3} \right] + \frac{n}{2}.$$

O histograma a seguir, gerado em Matlab 7, verifica a concordância dos resultados, com 100.000 amostragens, considerando-se $n = 40$.

Gráfico 4. 2. Histograma de números gerados que determinam o *locus* do *crossing-over*. A linha pontilhada corresponde aos valores assumidos pela função de distribuição $g(x)$.

3. Deseja-se uma distribuição de probabilidades semelhante à anterior (parabólica com máximo central), mas na qual os resultados estejam distribuídos no intervalo $[-1,1]$, para, ao sortear um número aleatório que será responsável por uma mutação, os mais próximos de zero sejam mais prováveis. Por qual razão isto é feito?

A secção 5.1.1 vai mostrar como se pode confinar cada vez mais a faixa em que podem variar os coeficientes da série que aproxima a função. Considerando-se que já se tenha em mãos uma boa população, as novas gerações de coeficientes devem estar próximas da atual. Sendo δ_k a faixa de confinamento das variações do coeficiente da série c_k , o coeficiente mutante será: $c_k^M = c_k + x \cdot \delta_k$, onde x é um número aleatório obtido com a distribuição postulada.

Tem-se, então, fatorando a expressão pelas suas raízes:

$$g(x) = k(x-1)(x+1) \text{ e } \int_{-1}^1 g(x) dx = 1 \Rightarrow k = -\frac{3}{4}.$$

$$\text{Portanto, } g(x) = \frac{3}{4}(1-x^2).$$

$$\text{Pelo exposto no início desta secção: } \int_{-1}^x \frac{3}{4}(1-x^2) = r.$$

$$\text{Esta última equação tem 3 soluções, mas a única aceitável, que para } 0 \leq r \leq 1 \Rightarrow -1 \leq x \leq 1 \text{ é } x = 2 \sin\left(\frac{\arcsin(2r-1)}{3}\right).$$

A figura seguinte compara os resultados dessa previsão com o obtido sortearo 10000 números aleatórios.

Gráfico 4.3. Distribuição normalizada dos números obtidos a partir da função geratriz deduzida.

Essa distribuição será usada em uma das fases do processo evolutivo adaptativo na construção de uma nova geração, conforme será descrito na secção 5.1.1.

Ela torna a busca em torno do valor atual mais fecunda, mas não impede que os extremos da região de confinamento sejam visitados, garantindo a diversidade.

4. Supondo que se queira uma distribuição na qual os melhores classificados em um ranking tenham maior probabilidade de serem sorteados, uma proposta possível é ter-se:

$$P(k) = \frac{\alpha}{k}, \text{ onde } k = 1, 2, \dots, n \text{ é a posição no ranking de } n \text{ indivíduos que o candidato ocupa.}$$

Quanto menor o valor de k , maior a probabilidade de ele ser sorteado.

Imaginando que n seja muito grande, pode-se aproximar o problema discreto para o contínuo.

A integral da função de distribuição deve totalizar 100%, logo:

$$\int_{-\infty}^{\infty} \frac{\alpha}{k} dk = \int_1^n \frac{\alpha}{k} dk = 1, \text{ portanto, } \alpha \cdot \ln n = 1 \Rightarrow \alpha = \frac{1}{\ln n}.$$

Conforme expõe a secção 4.1.4 – Roleta viciada, $P(k)$ é o vício da roleta.

A secção 4.1.4 deduz que, para obter a função geratriz desse vício, impõe-se:

$$r = \int_{-\infty}^x P(z) \cdot dz = \int_1^x \frac{1}{\ln n \cdot z} \cdot dz \Rightarrow r = \frac{1}{\ln n} \ln z, \text{ onde } r \text{ é um número aleatório, com distribuição}$$

uniforme entre 0 e 1, gerado pelo Matlab.

Obtém-se então: $\ln n^r = \ln z \Rightarrow z = n^r$. (função geratriz).

Impondo essa condição a cada sorteio, os melhores do *ranking* terão maior probabilidade de serem sorteados, mas não são descartados nem os intermediários e nem os piores. Essa é a função geratriz do sorteio na escolha dos “noivos” que vão acasalar. As noivas são escolhidas ao acaso, tentando manter a diversidade.

O gráfico seguinte ilustra a concordância entre a distribuição prevista usando a função geratriz $z = n^r$, a distribuição construída com 10000 sorteios, para uma população de 40 cromossomos. Como z é natural, usa-se o comando *floor* do Matlab para fazer o arredondamento para o natural imediatamente inferior ao resultado obtido.

Gráfico 4.4. Resultados de uma roleta viciada na qual quanto menor for o número, maior será a probabilidade de ele ser sorteado.

5. Neste caso, de acordo com a importância relativa de cada coeficiente na “construção” da função de aproximação, este trabalho levou em conta o tamanho de cada “tijolo”, qual seja, a raiz quadrada da integral do quadrado do elemento da série.

Em termos de polinômios de Legendre, tem-se: $\int_{-1}^1 P_k^2(u)du = \frac{2}{2k+1}$, $k = 0, 1, \dots, n$,

logo, para cada cromossomo, o resultado dos *cross-over* nos coeficientes de ordem alta tem menos importância do que aqueles que ocorrem nas ordens mais baixas. Em outras palavras, se houver diversidade na população, os *cross-over* nos coeficientes de ordem mais baixa terão mais impacto nas mudanças do que os de maior ordem. Pondo-se a proporção, pode-se escrever a probabilidade de que o *cross-over* ocorra nos coeficientes de ordem k , como sendo:

$$p(k) = \frac{\alpha}{\sqrt{2k-1}}, \text{ agora com } k = 1, 2, \dots, n, \text{ onde } n-1 \text{ é o grau do polinômio de}$$

aproximação. Observe-se que o *cross-over* não pode ocorrer no último gene ($k = n$), e nem antes do primeiro ($k = 0$), o que significaria não haver o processo. Portanto, o índice k representa o *locus* do gene imediatamente anterior ao ponto em que vai ocorrer a operação.

Considerando-se um número muito grande de genes, pode-se fazer uma aproximação considerando k um elemento do intervalo contínuo $[1, n]$, de forma que:

$$\int_1^n g(k) \cdot dk = 1, \text{ ou seja, } \alpha \int_1^n \frac{dk}{\sqrt{2k-1}} = 1 \Rightarrow \alpha = \frac{1}{\sqrt{2n-1}-1}.$$

Portanto, a distribuição de probabilidades é descrita por:

$$g(k) = \frac{1}{(\sqrt{2n-1}-1)\sqrt{2k-1}}, \text{ com } k \in [1, n+1].$$

Como $r = \int_1^k g(k)dk$, obtém-se:

$$\text{Eq. \{ 54\}} \quad k = 1 + r^2 \left(n - \sqrt{2n-1} \right) + r \left(\sqrt{2n-1} - 1 \right), \text{ com } r \in [0, 1].$$

Como $k \in \mathbb{N}^*$, a função *floor* do Matlab fará o arredondamento para o natural inferior mais próximo do número sorteado.

Por exemplo, considerando que sejam 26 coeficientes ($n = 25$), ter-se-á:

$$k = 1 + r^2 (25 - 7) + r (7 - 1) = 1 + 6r + 18r^2$$

O histograma seguinte, com 100000 sorteios, corrobora a análise.

A Eq. \{54\}, com o comando *floor*, será utilizada nos sorteios dos *loci* de *cross-over*.

Gráfico 4. 5. Distribuição das probabilidades dos loci de *cross-over*.

Mutação

Por pequenos erros na cópia dos genes dos pais, o descendente pode ter elementos de seu DNA ligeiramente modificados. Nos seres vivos reais isso pode acontecer também devido à incidência de radiações ionizantes. Nos AGs também se implementa a mutação. Assim como na vida real, a intenção é provocar variedade. Variedades que trazem em si mesmas mecanismos adicionais de sobrevivência podem emergir desse processo, como também podem emergir indivíduos altamente vulneráveis.

Altas taxas de mutação poderiam produzir uma nova geração totalmente aleatória, pondo por terra todo trabalho anterior da seleção natural. Por outro lado, a ausência de mutação leva à estagnação da diversidade. Em geral, nos AGs essa taxa fica em torno de 1,0 %, mas pode ser reduzida gradativamente à medida que a população evolui.

Com essas considerações, vê-se que a mutação em si não é boa ou ruim. É a seleção natural que se encarrega de aceitar ou rejeitar os indivíduos mutantes.

Fazendo-se um paralelo com a Termodinâmica, a seleção natural seria o Diabo de Maxwell²¹.

Figura 4. 1. O diabo de Maxwell controlando o fluxo de partículas e produzindo um ambiente mais “organizado”.

4.1.5 Espaço de soluções e Inteligência Artificial

Suponha-se que determinada espécie se alimentasse dos frutos caídos de árvores. Outras espécies poderiam ter o mesmo tipo de alimentação. A escassez de frutos no solo tornaria mais aptos à sobrevivência os indivíduos capazes de chegar aos pontos altos, para colher os frutos antes da queda. Em época de fartura essa seria uma aptidão inútil. O conjunto de todas as possíveis soluções para esse problema é o espaço de soluções. Mas quais são elas?

A primeira que nos vem à mente é que o indivíduo tenha membros capazes de se agarrar aos troncos e galhos para realizar sua escalada. No entanto, isso não seria necessário se ele pudesse voar, ou ainda se ele fosse forte o suficiente para chacoalhar a árvore a ponto de os frutos caírem, ou ainda hábil em atirar pedras para derrubar os frutos.

Vê-se, por exemplo, que na diversidade habitam soluções talvez nem imaginadas na fase em que se delineava o problema. Nesse sentido, os AGs podem se tornar elementos da Inteligência Artificial. Ele pode apresentar soluções inesperadas, um conceito normalmente denominado de **emergência**. O exemplo 2 do capítulo 6 ilustra uma situação em que isto acontece nos *PVIs*.

²¹ Ser imaginário concebido por Maxwell que controlaria a passagem de partículas em um orifício que interligasse dois gases a temperaturas diferentes. Se ele permitisse que apenas as partículas mais velozes do gás mais frio passassem para o lado do gás mais quente, enquanto somente as mais lentas do mais quente passassem para o mais frio, a entropia do sistema estaria sendo reduzida sem dispêndio de energia. O paradoxo fica explicado se admitirmos que até mesmo o Diabo precisa de energia para funcionar e fazer as escolhas.

4.1.6 Aplicações dos AGs

Problemas nos quais o número de iterações para se chegar a uma solução possam ser descritos por um polinômio, como, por exemplo, ordenar uma tabela com n números, são chamados em computação de problemas polinomiais [Campos, 2007].

Os AGs se aplicam a problemas cujos processamentos não seriam polinomiais. Essa classe de problemas é chamada de problemas não-polinomiais (NP). Um exemplo desse tipo é o problema logístico do caixeiro-viajante [Baraglia *et al.*, 2001]. À medida que aumenta o número de cidades a serem visitadas, a busca pelo melhor trajeto aumenta exponencialmente.

Existem muitos problemas importantes em que é muito difícil encontrar uma solução, mas que, uma vez encontrada, é fácil de ser verificada. São chamados de problemas NP-completos. Por exemplo, a busca de uma solução polinomial exata para um *PVI* não linear, para a qual não se conheça um método analítico, usando computação evolutiva é um problema NP-completo, pois, encontrada uma solução exata, ela pode ser verificada.

Um aspecto importante a ser observado quando da aplicação dos AGs é que ele não deva ter a intenção de esgotar o espaço de soluções. Isso levaria o processamento para a classe não-polinomial, o que se quer evitar quando da sua aplicação.

Os AGs vão visitar alguns pontos do espaço de soluções, que muitas vezes tem infinitos elementos, e **evoluir** para uma determinada solução.

Otimização no design de CPUs [Chen *et al.* 2008], contaminação da água no subsolo [Sayeed & Mahinthakumar, 2005], controle de fluxo em tubo cilíndrico rotativo [Sengupta *et al.*, 2007], modelos dinâmicos para controles florestais [Tietjen & Huth, 2006], são alguns exemplos em publicações recentes que fazem uso dos AGs.

4.2 Variedades de Algoritmos Genéticos

Conforme o problema a ser otimizado e o seu espaço soluções, o AG é aplicado com algumas variações. Esta secção descreve a seguir duas variações que se aplicam especialmente a problemas cujo espaço de soluções é contínuo.

Uma comparação de diferentes tipos de AGs foi desenvolvida por Hrstka *et al.* [2003], Krink *et al.* [2006] e Coelho [2008].

4.2.1 Algoritmo Genético Diferencial (*dAG*)

Desde a publicação seminal sobre evolução diferencial “Differential Evolution – A simple and efficient heuristic for global Optimization over continuos spaces”, [Storn & Price, 1997], essa abordagem tem sido utilizada com bons resultados em vários problemas *NP* de espaço de soluções contínuos.

Neste tipo de algoritmo, o intervalo contínuo em que pode variar o valor de cada gene é dividido em n subintervalos. A seguir, uma nova população é construída com n cromossomos. Se cada cromossomo tem p genes, o número de possibilidades (N_p) para um determinado cromossomo é, pelo princípio fundamental da contagem:

$$N_p = \underbrace{n \cdot n \cdot \dots \cdot n}_{p \text{ vezes}} = n^p.$$

Por exemplo, se o algoritmo está buscando os coeficientes de um polinômio de grau 15, em que cada coeficiente pode variar em intervalos com 20 subintervalos, o espaço de soluções terá $N_p = 15^{20}$ elementos, ou seja, é um problema computacional tipicamente não polinomial.

Partindo, por exemplo, de uma população aleatória, O *dAG* aborda esse problema gerando, a partir do melhor da população, uma nova população na qual cada gene é uma soma do melhor da população anterior com uma pequena variação aleatória (para mais ou para menos), determinada por um parâmetro F , chamado de parâmetro de escala [Kaelo & Ali, 2007]. Esta é a fase das mutações. Em seguida, um pacote de gerações é processado, nas quais os cromossomos se acasalam produzindo descendentes com *crossing-over*, e, a cada geração, uma nova classificação é efetuada, até que se atinja determinada condição pré-estipulada de parada.

Algumas modificações no *dAG* têm sido propostas, de acordo com as mudanças em problemas abordados [Hrstka & Kucerovala, 2004], [Bergey & Ragsdale, 2005]. Em geral, essas alterações têm seu foco na fase de mutação e no parâmetro de escala.

Uma comparação de eficiência e desempenho do *dAG* em relação a outros métodos de computação evolutiva mostrou, pelo menos para os problemas de espaços de soluções contínuos analisados, que o *dAG* foi melhor [Hrstka *et al.*, 2006].

Uma variante do *dAG* foi proposta por Bergey e Ragsdale [2005]. Melhorias nos operadores de *cross-over* também foram propostas por Hrstka e Kucerovala [2004], e as equações diferenciais para controle de um robô foram também objeto de abordagem com o

dAG [Rout & Mittal, 2008]. Recentemente, a obtenção de parâmetros para sistemas dinâmicos usando o *dAG* diferencial também foi utilizada, com algumas modificações por Liu *et al.* [2008].

Neste trabalho, como se tornou possível confinar a pequenos intervalos as possíveis variações nos valores dos coeficientes a serem evoluídos, a escolha recaiu sobre o *dAG*, com algumas modificações que serão descritas no capítulo 5, seção 1.

4.2.2 Algoritmo Genético Compacto e Particle Swarm

Em relação ao algoritmo genético compacto, há duas características importantes que merecem ser mencionadas.

A primeira é “leveza”. Por trabalhar com um código relativamente simples e poucas variáveis, ele ocupa pouca memória e tem um processamento relativamente rápido. A segunda é a demonstração de sua convergência [Rastegar & Hariri, 2006].

Outra abordagem em computação evolutiva, mas que não é um algoritmo genético é o “Particle Swarm”.

Nesse tipo de algoritmo, tem-se uma população de partículas de tal forma que cada partícula é um objeto móvel que “sobrevoa” o espaço de soluções e é atraída por pontos previamente visitados com alto *fitness* [Krink *et. al.*, 2003, p. 1229].

Nesta tese, como a escolha recaiu sobre o *dAG*, essas outras possibilidades de computação evolutiva foram deixadas para análises futuras.

Capítulo 5. Os *PVIs* e o *SAM*

Frank & Ernest Bob Thaves

Seria possível, com o poder de computação atual, gerar coeficientes aleatórios para um polinômio de grau n , e chegar a uma²² solução exata de um *PVI*?

Tanto quanto se pode processar ao longo de três anos trabalhando com os *dAG* aplicado às *EDs*, a resposta foi **não**. Em uma busca sem restrições, o espaço de soluções é sempre visitado em regiões aleatórias, sem uma convergência.

No entanto, com algumas modificações no *dAG*, as publicações recentes têm denotado a aplicabilidade do *dAG* em vários tipos particulares de *PVIs* [Liu, 2008], [Sengupta *et al.*, 2007], [Bryden *et al.*, 2006], [Cao *et al.*, 2000], incluindo fenômenos biológicos de relativa complexidade [Moscovitch *et al.*, 2004], [Sayeed & Mahinthakumar, 2005] e da Mecânica Quântica [Nakanishi & Sugawara, 2000], [Chaudhury & Bhattacharyya, 1998].

Como expõe o capítulo 1, a busca neste trabalho foi por um método geral para resolução de *PVIs* até quarta ordem, em intervalos fechados quaisquer.

Paralelamente, o caráter multidisciplinar do grupo de estudos de Ciências Cognitivas convidou esta tese a considerar dois importantes ingredientes dos processos heurísticos, determinantes em suas eficiências:

- Restrição (seleção natural) e
- Aprendizado.

Sem a restrição, não há seleção. Com a busca em um espaço de soluções não delimitado, qualquer que seja o poder computacional disponível, essa busca deságua no

²² O exemplo 2 do cap. 6 mostra que pode haver uma família de soluções para o *PVI*, conforme ele for proposto.

binômio convergência-aleatoriedade. A pergunta imediata é: em qual região devemos procurar os melhores candidatos?

Quanto mais restritiva for a resposta, maiores serão as possibilidades de se encontrar o melhor. Em outras palavras, quanto menos restrições se impõem, mais “perdido” o algoritmo de busca se torna. Diferentes restrições têm sido propostas, conforme o problema abordado, mostrando o ganho em tempo de computação e na precisão dos resultados obtidos [Sengupta *et al.*, 2007], [Liu, 2008].

A questão do aprendizado está ligada à memória, às buscas anteriores. Considerando cada busca do programa por melhores coeficientes como uma nova busca, sem passado, que probabilidade ele teria de chegar ao melhor candidato? Certamente, a mesma que teria de ir para o pior. De fato, o *SAM* só pôde evoluir quando foi capaz de obter a margem de erro dos coeficientes do melhor de cada geração, como descreve a secção seguinte.

Finalmente, como ponto favorável dos *AGs*, algumas vezes rivalizando com os processos analíticos deterministas, os processos heurísticos podem responder com a **emergência**. Há a possibilidade de se obterem soluções que não provenham de um silogismo clássico, mas que possam se apresentar mais gerais que outras conhecidas, ou até mesmo inesperadas, como mostra o exemplo 2 do capítulo 6.

5.1 Processo evolutivo-adaptativo na resolução das *EDs*

Considerando-se que um polinômio de grau n seja o melhor candidato até então encontrado para resolver o *PVI* (critério *LG*), é natural perguntar-se o quão longe ele está da solução.

À primeira vista, essa pergunta parece ingênua, já que se fosse conhecido exatamente o quão longe ele está da solução, saber-se-ia a solução, mas não é esse o caso, como descreve a secção seguinte.

5.1.1 Dispersão dos coeficientes de Legendre

Considere-se a derivada de maior ordem da equação diferencial escrita em função das outras derivadas, da função e da variável independente, na forma:

$$\text{Eq. \{ 55\}} \quad \frac{d^q y_t}{du^q} = G(y_t, y_t^{(1)}, \dots, y_t^{(q-1)}, u).$$

Com a *ED* em mãos, a igualdade acima pode ser obtida rapidamente. No Matlab a função *solve* automatiza o processo²³. Essa tarefa está programada no *SAM*, e funciona, por assim dizer, nos bastidores.

Em seguida, substitui-se o melhor polinômio candidato de determinada geração de um processo evolutivo em $G(y_t, y_t^{(1)}, \dots, y_t^{(q-1)}, u)$. Sem dúvida, se nessa substituição, o

resultado obtido for exatamente $\frac{d^q y_t}{du^q}$, o candidato é a solução exata. Mas, e se não for?

Nesse caso, avalia-se G ao longo do intervalo e obtêm-se os coeficientes de Legendre relativos a essa função. A seguir, integram-se esses coeficientes q vezes, com a matriz de integração M_{LL} , para obter os coeficientes da função que gerou essa derivada, impondo-se as *CI*. No caso da solução exata, os coeficientes assim obtidos seriam iguais ao do melhor candidato. Se não forem, pode-se verificar a diferença entre eles e então procurar novos candidatos entre os dois extremos.

Dispersão simples

Considerando o coeficiente genérico c_k da série de Legendre que aproxima a função e o coeficiente obtido conforme descrito ao final da subsecção anterior $\overline{c_k}$, tem-se: $\delta_k = |c_k - \overline{c_k}|$. Obtém-se assim uma faixa em que se irá procurar a melhora dos coeficientes. O desvio padrão entre os coeficientes foi denominado dispersão simples.

Essa pesquisa verificou que tal faixa de busca, aqui denominada como “envelope”, vai ficando cada vez mais estreita, contraindo o volume do espaço de soluções, até estacionar em valores muito pequenos na estagnação evolutiva.

Com esse recurso, as novas gerações vão aprendendo com as antigas, a seleção vai descartando os piores, os melhores vão produzindo bons descendentes, e os *crossing-over* abrem a oportunidade de se pegar o melhor de cada um. Além disso, pequenas mutações varrem os vãos entre os candidatos. O processo pode continuar até uma precisão pré-estabelecida pelo usuário, ou até que se observe a estagnação evolutiva.

²³ Poucas são as *EDs* implícitas e, dentre elas, poucas são aquelas em que não conseguimos isolar a derivada de maior ordem explicitamente. Entretanto, se isso não for possível, como é o caso, por exemplo, da *ED*: $\ln y^{(2)} = y^{(1)} \cdot y^{(2)}$, o algoritmo não consegue aplicar o processo adaptativo. Nessa condição, o intervalo se mantém fixo e a evolução é extremamente lenta.

Dispersão relativa

Neste trabalho, os polinômios de Legendre foram utilizados **não normalizados**, pelo fato de isso se mostrar desnecessário.

Entretanto, como $\int_{-1}^1 [P_k(u)]^2 du = \frac{2}{2k+1}$, à medida que aumenta a ordem do polinômio de Legendre a integral anterior diminui, ou seja, em termos relativos, um mesmo erro absoluto no valor do coeficiente para os coeficientes de ordens maiores significa um erro percentualmente maior no valor desse coeficiente em relação à base.

Por esta razão, para maior clareza na avaliação das dispersões, esta tese optou por exibir a dispersão relativa dos coeficientes²⁴, definida de forma que:

$$\text{Eq. \{ 56\}} \quad \sigma_k^{\text{rel}} = \sigma_k \sqrt{\frac{2k+1}{2}},$$

uma vez que a dispersão simples dos coeficientes de maior ordem vai ficando com visualização cada vez mais difícil à medida que se aumenta a ordem da série que aproxima a função.

Para melhor esclarecer todo o processo, apresenta-se a seguir um exemplo, em um *PVI* que poderia ser resolvido de forma direta analiticamente.

A explanação do processo evolutivo nesta tese desenvolvido será descrita na secção 5.4.

Exemplo

Supondo que se queira uma boa aproximação, com um polinômio de grau 9, para o *PVI*: $(1-x^2)y^{(2)} - 2xy^{(1)} + 30y = 0$, com $x \in [-1,1]$, sob as *CI* $\vec{y}_0 = [-1, 15]$, cuja solução

analítica é o polinômio de Legendre de ordem 5: $P_5(x) = \frac{63}{8}x^5 - \frac{35}{4}x^3 + \frac{15}{8}x$.

O *SAM* apresenta a equação e as condições iniciais transformadas para o intervalo de Legendre que, neste caso, é o intervalo original do *PVI* e não há alterações.

$$(1-u^2)y_t^{(2)} - 2uy_t^{(1)} + 30y_t = 0, \text{ com } u \in [-1,1].$$

Isolando-se a derivada de maior ordem, tem-se:

²⁴ Pensando-se na base de Legendre como tijolos de diferentes tamanhos que compõem a função, isso é equivalente a comparar-se o erro na quantidade em relação ao tamanho do tijolo, isto é: $\sigma_k^{\text{rel}} = \sigma_k / \sqrt{\frac{2}{2k+1}}$, que leva ao mesmo resultado.

$$y_t^{(2)} = \frac{2uy_t^{(1)} - 30y_t}{1-u^2} \text{ com } u \in [-1,1].$$

Inicialmente, esta pesquisa tomava com ponto de partida os pontos gerados por métodos numéricos consagrados, que são funções residentes do Matlab. Como descrito na secção 2.2.6, em termos de evolução a idéia foi chegar ao *homo sapiens* já partindo de um primata. Mas a pergunta nº 7, se o processo evolutivo funcionaria partindo da estaca zero, ou seja, da sopa primordial, parecia não calar.

A precisão e a rapidez dos métodos numéricos já existentes permitiram um balizamento do processo evolutivo. Foram muitos ajustes: taxas de mutação, *crossing-over*, número de cromossomos etc.

Quando as curvas obtidas começaram praticamente a se superpor, com polinômios de grau relativamente baixo, parecia a hora de tentar e isto foi feito neste exemplo.

Partindo de um polinômio nulo, o ajuste das condições para manutenção das condições iniciais vai impor que dois dos coeficientes sejam as adequadas combinações lineares dos outros, levando em conta as condições iniciais, conforme explana a secção 3.6.2. Dessa forma, o polinômio de partida já não é mais nulo, mas sim um que preserva as *CI* para a função e suas derivadas.

Após 60 gerações, com a evolução, obtiveram-se, para o intervalo de trabalho, os seguintes coeficientes da série de Legendre, com 3 algarismos significativos.

Tabela 5. 1. Coeficientes da série de Legendre após 60 gerações, partindo do polinômio nulo.

Ordem	0	1	2	3	4	5	6	7	8	9
Coef.	0,00E+00	0,00E+00	0,00E+00	0,00E+00	-1,79E-01	6,72E-01	-1,49E-01	0,00E+00	0,00E+00	0,00E+00

Qual é o resultado que se obtém ao substituir o polinômio candidato na equação

$$G(u, y_t, y_t^{(1)}) = \frac{2uy_t^{(1)} - 30y_t}{1-u^2} ?$$

Se ele fosse a solução exata, iria corresponder aos valores de $y_t^{(2)}$, como determina a *EDO*. Sejam o que forem, pode-se considerar o gráfico dos valores de G , produzidos por esse candidato, no domínio de trabalho.

Nos extremos do intervalo ($x = \pm 1$), a função G tem singularidades, mas a *QGL* não utiliza esses pontos como nodos, logo, eles sequer são considerados. De qualquer modo, para exibir o gráfico incluindo os extremos, esses pontos foram extrapolados por um polinômio de Lagrange de grau 5 [Campos, 2007, p.128], considerando os seis pontos mais próximos do extremo.

Podem-se obter os coeficientes de Legendre para o resultado da substituição desses valores — os coeficientes de Legendre de G — como se faz para qualquer conjunto de ordenadas ao longo de um intervalo.

A figura acima exhibe os valores da derivada de maior ordem que o polinômio candidato produziu no intervalo de trabalho. A partir deles será obtida uma série de Legendre que, depois de integrada duas vezes, será comparada com o próprio candidato.

Agora, analisam-se as ordenadas ao longo do intervalo e obtemos para elas a série de Legendre até a ordem $(n - q)$, onde q é a ordem da ED , conforme suas abscissas. O motivo de se impor o grau $(n - q)$ é que esse polinômio será depois integrado pela matriz de integração M_{LI} , q vezes, e comparado com o polinômio candidato. Nesse processo de integração as condições iniciais são obedecidas, e a imposição do grau $(n - q)$ garante que os vetores a serem comparados tenham o mesmo número de coeficientes.

Neste exemplo, $G(u)$ será integrada duas vezes, impondo-se as condições iniciais, conforme explicita a secção 3.6.2.

Feita a integração, se não houver diferença, a solução é exata; se houver diferenças entre os coeficientes, pode-se avaliar de quanto é essa diferença.

O gráfico seguinte ilustra a dispersão²⁵ relativa (σ_{rel}) dos coeficientes neste estágio do processo evolutivo.

Gráfico 5. 1. Dispersão relativa entre os coeficientes obtidos pela matriz de integração e os coeficientes do polinômio candidato.

Para as futuras gerações, as variações dos coeficientes ficarão confinadas por essa dispersão e, após certo pacote de gerações, a dispersão é novamente verificada e um novo envelope é definido. Neste exemplo, os pacotes foram inicialmente de 30 gerações. Nas primeiras gerações é conveniente usarem-se pacotes pequenos para rapidamente obter-se um novo **envelope**, mais constrito.

O gráfico seguinte ilustra a dispersão normalizada após mais 300 gerações, com cerca de 2 min. de processamento.

²⁵ Embora seja um abuso de linguagem, uma vez que não se está lidando com um conjunto de erros acidentais, optou-se pelo uso do termo “dispersão”, já que o gráfico não nos mostra de fato o erro nos coeficientes, mas sim o quanto um diverge do outro.

Gráfico 5. 2. Com a evolução, após 300 gerações, diminuiu a dispersão nos coeficientes.

Gráfico 5. 3. Resíduo decorrente da substituição da série aproximadora na ED.

Como a dispersão agora é baixa e o resíduo também, é razoável querer se comparar os gráficos da função e suas derivadas com os valores exatos, conforme ilustra a figura seguinte.

Gráfico 5. 4. O SAM encontrou um polinômio de grau 9 que mimetiza a função e sua derivada ao longo de todo intervalo. No limite da resolução do gráfico, os pontos estão todos superpostos. A margem de erro é menor do que um milionésimo de *pixel*.

A dispersão possibilita também obter o limite superior do erro, como vai descrever a secção 5.2.2. É importante enfatizar que, na maioria dos casos analisados, a margem de erro ficou bem menor, isto é, esse limite superior considera o extremo das piores possibilidades.

Neste caso, os valores obtidos foram:

```
Majorante do erro em D0 : 1.41e-013
Majorante do erro em D1 : 1.88e-012
Majorante do erro em D2 : 2.99e-011
```

A tabela seguinte apresenta os coeficientes da série, com três algarismos significativos. No entanto, vale ressaltar que, como foi estipulado ao Matlab para trabalhar com 180 dígitos na representação dos números, esses coeficientes estão armazenados na resposta final como a razão entre dois inteiros, e assim devem ser usados para manter a precisão alcançada.

A mudança de base da série de Legendre para a forma canônica é feita por uma matriz simbólica²⁶, de forma a preservar a precisão. Por exemplo, o coeficiente do termo de ordem 5 na forma canônica é:

20444668880480902419815530881970047/2596148429267413814265248164610048.

A diferença entre esse número e o valor analítico $(63/8)$ é $7,1 \cdot 10^{-15}$.

Tabela 5. 2. Coeficientes da série de Legendre depois de 400 gerações.

Ordem	0	1	2	3	4	5	6	7	8	9
Coef.	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

Gráfico 5. 5. Erro ao longo do intervalo na função e nas derivadas na comparação com a solução analítica. Note-se que os valores majorantes calculados são de fato bem maiores que os erros máximos.

Como processador não tem precisão ilimitada, as divisões e potenciações, por exemplo, carregarão uma pequena margem de erro, principalmente porque os nodos da *QGL* são expressões com grande número de algarismos significativos.

Curiosamente, neste exemplo, essa margem de erro fez com que a solução exata deste *PVI* ficasse em quinto lugar no *ranking* dos polinômios candidatos, ou seja, os coeficientes de ordem 10^{-13} ou inferiores são “ruídos” do processo numérico.

²⁶ Operação feita pela função *Leg2Can.m*, construída aqui usando o Teorema da Matriz Operacional, que, dados os coeficientes de Legendre na forma de um vetor coluna, obtém o vetor coluna da série canônica.

Em termos de seleção natural, a melhor resposta obtida pelo *SAM* soa como: — Esta é a melhor resposta para o computador que você tem. Algo que faz parte do ambiente em que o processo evolutivo-adaptativo aconteceu. No ambiente do processamento, o que o analítico considera como ruído é um ingrediente, às vezes decisivo, da seleção natural. Em outro computador, com outras limitações decimais, a resposta poderia ser diferente.

Esta é uma breve descrição para responder à pergunta nº 6. A secção 5.1 descreve mais detalhadamente o processo, acompanhada de um exemplo.

Particularmente interessante foi o 3º exemplo da secção 6.1, uma equação diferencial de 3ª ordem, não linear, para a qual a função ode45 do Matlab não inicializava, impossibilitando a solução numérica. Como a equação foi montada a partir de uma solução exata conhecida, por assim dizer, de uma engenharia reversa, era possível testá-la com o *SAM*. A implementação resultou em erros cada vez menores, até que, após cerca de 2 min. e 600 gerações, o *SAM* encontrou praticamente a solução exata.

Além disso, o capítulo 6 apresenta mais 3 exemplos para os quais os métodos numéricos (*MN*) não conseguiram dar a partida, pelo fato de a matriz de massa ser singular, para os quais o *SAM* obteve aproximações com margens de erro máximo para a função e suas derivadas da ordem de 10^{-12} . Isso é possível porque as abscissas da *QGL* não incluem os extremos do intervalo, logo, se neles houver uma singularidade, ela será desconsiderada no processo de integração, como já discutido. Para a função e as derivadas até a ordem $q-1$, esses valores já estão definidos pelas *CI*. Para a última derivada, esse ponto é obtido com um polinômio de Lagrange de grau 5, obtido pelas 5 coordenadas mais próximas do ponto analisado (um dos extremos ou ambos), apenas para exibição do gráfico.

5.2 Majorantes das margens de erro

Uma questão importante em qualquer método numérico é saber qual é a sua margem de erro. Esta secção mostra como pode ser feita uma estimativa do majorante do erro para a série que aproxima a solução do *PVI* e também para suas derivadas.

5.2.1 Majorante do erro da série em relação à função

Primeiramente, vai-se considerar a função $f(u)$ e suas derivadas aproximadas por uma série de Legendre no intervalo de trabalho, pela função $f^*(u)$ até a ordem n . Tem-se:

$$f(u) = \sum_{k=0}^{\infty} c_k P_k \quad \text{e} \quad f^*(u) = \sum_{k=0}^n (c_k + \delta_k) \cdot P_k, \quad \text{onde } \delta_k \text{ é o erro absoluto que afeta o}$$

coeficiente c_k .

Como descrito na secção 3.3.2, no tópico Importância da Ortogonalidade, não há como incorporar pequenas diferenças nos coeficientes de Legendre de menor ordem de forma a compensar o truncamento da série. Dessa forma, esse trabalho considerou que os δ_k advém da dispersão externa dos coeficientes, pois, caso a ordem da aproximação levasse à solução exata, essa dispersão seria nula. Isto é, após o processo evolutivo, o truncamento se reflete na dispersão, assim vão-se considerar então os coeficientes que se têm em mãos, e não os que faltam. Outro olhar que levaria à mesma hipótese é que, considerando que serão usados polinômios de graus suficientemente altos para uma boa aproximação, os erros acumulados dos termos de ordem maior do que a série são desprezíveis diante do calculado. Com essa hipótese, pode-se escrever:

$$\delta f(u) = f^*(u) - f(u) = \sum_{k=0}^n (c_k + \delta_k - c_k) \cdot P_k = \sum_{k=0}^n \delta_k \cdot P_k.$$

$$\text{Eq. \{ 57 \}} \quad \delta f(u) = \sum_{k=0}^n \delta_k \cdot P_k \leq \sum_{k=0}^n |\delta_k \cdot P_k| \leq \sum_{k=0}^n |\delta_k \cdot P_{k,\text{máx}}| \quad (\text{expressão majorante}).$$

Quando essa soma será máxima?

Considerando o pior dos cenários, todos os δ_k agirão no sentido de aumentar o erro em cada ponto e, exatamente, nos pontos em que P_k é máximo. Na verdade, alguns afetarão o erro para mais, outros para menos e, não necessariamente nos pontos em que P_k é máximo. De qualquer forma, a expressão permite determinar²⁷ um majorante para o erro. O valor máximo de P_k , qualquer que seja k , é 1.

²⁷ Considerando válida a hipótese descrita no parágrafo anterior.

Assim, o erro máximo que aproximação pelo polinômio pode acarretar na avaliação da função no domínio de trabalho é:

$$\text{Eq. \{ 58\}} \quad |\delta f|_{\text{máx}} = \sum_{k=0}^n |\delta_k|.$$

Pode-se argumentar que o erro assim calculado está excessivamente majorado, mas, como os valores de δ_k , após a evolução, são muito exíguos, mesmo com essa majoração o majorante é relativamente pequeno, como mostrarão os exemplos do capítulo 6.

5.2.2 Majorante do erro nas derivadas da série

O cálculo do erro máximo para cada uma das derivadas é análogo ao cálculo feito para função.

Considera-se a diferença entre as derivadas da função e as derivadas da aproximação, até a ordem da *EDO*, e encontra-se a expressão majorante.

$$\text{Eq. \{ 59\}} \quad \delta f^{(j)}(u) = f^{*(j)}(u) - f^{(j)}(u), j = 1, \dots \text{ até a ordem da } EDO.$$

Esse cálculo é tedioso, particularmente se o polinômio de aproximação tiver grau muito alto. Como resultado, obtém-se os δ_k das derivadas, que serão combinações lineares dos δ_k da função.

No entanto, esse cálculo pode ser feito com a matriz de diferenciação (M_{LD}) de forma rápida automática, já considerando que $P_{k.\text{máx}} = 1$. O expoente na matriz de diferenciação representa o número de vezes que ela foi aplicada ao vetor de coeficientes.

$$\text{Eq. \{ 60\}} \quad \delta f_k^{(j)} = M_{LD}^j(c_k) - M_{LD}^j(c_k + \delta_k), j = 1, \dots, \text{ordem da } EDO, k = 0, \dots, n.$$

Em seguida, monta-se a expressão majorante:

$$\text{Eq. \{ 61\}} \quad |\delta f^{(j)}|_{\text{máx}} = \sum_{k=0}^n |\delta_k^j|.$$

Como foi constatado e era de se esperar, geralmente²⁸, as margens de erro nas derivadas são um pouco maiores, pois são aproximadas por polinômios de menor grau. Além

²⁸ Houve casos em que a margem de erro na derivada ficou menor do que a margem de erro na função. Nesses casos, o processo evolutivo continha ainda margens de erro muito altas nos coeficientes de ordem baixa, que vão desaparecendo nos cálculos das derivadas de ordens mais altas, conforme mostram as expressões ao final desta seção.

disso, o processo de diferenciação traz para os coeficientes o expoente da potência de u que se está derivando; mas, mesmo assim, se o processo numérico for bastante preciso, as margens de erro continuam baixas, e valores altos nessas margens devem denunciar a falta de precisão do processo.

A título de exemplo, se o polinômio que aproxima a função for de grau 7 e os erros que afetam seus coeficientes forem δ_i ($i = 0, 1, \dots, 7$), os erros máximos para a primeira derivada e a segunda serão:

$$\delta f_{\text{máx}}^{(1)} = \delta_1 + 3 \cdot \delta_2 + 6 \cdot \delta_3 + 10 \cdot \delta_4 + 15 \cdot \delta_5 + 21 \cdot \delta_6 + 28 \cdot \delta_7;$$

$$\delta f_{\text{máx}}^{(2)} = 3 \cdot \delta_2 + 15 \cdot \delta_3 + 45 \cdot \delta_4 + 105 \cdot \delta_5 + 210 \cdot \delta_6 + 378 \cdot \delta_7.$$

5.2.3 Transposição dos majorantes de erros

Os tópicos anteriores discutiram os erros máximos no intervalo de trabalho, mas o que se deseja são os erros máximos no domínio original da função.

A subsecção 3.7.1 mostra como transformar a função, derivadas e o *PVI* do domínio original para o domínio de trabalho em transposições genéricas, lineares ou não.

As expressões podem ser aproveitadas para se fazer a transposição do erro. Por exemplo, para a transposição do erro na primeira derivada, usamos a Eq. {29}:

$$D_1 = D_{1t} \cdot \lambda, \text{ onde } \frac{du}{dx} = \frac{1}{dx/du} = \lambda, \text{ e escrevemos}$$

$$\text{Eq. \{ 62\} } \quad \delta_1 = \delta_{1t} \cdot \lambda.$$

Para a segunda derivada, usa-se a Eq. {30},

$$D_2 = D_{2t} \cdot \lambda^2 + D_{1t} \cdot \frac{d\lambda}{du} \cdot \lambda, \text{ logo:}$$

$$\text{Eq. \{ 63\} } \quad \delta_2 = \delta_{2t} \cdot \lambda^2 + \delta_{1t} \cdot \frac{d\lambda}{du} \cdot \lambda.$$

Para as derivadas de ordens superiores o processo é o mesmo, usando-se as equações da subsecção 3.7.1.

5.3 Qualidade de um método numérico de integração

Uma pergunta natural para qualquer problema envolvendo uma *ED* é: conhecida a solução numérica de *ED*, qual é a margem de erro em cada ponto?

A idéia de verificar a dispersão nos coeficientes de Legendre, descrita na secção 5.1.1, pelo menos nos casos analisados, foi capaz de realizar essa tarefa e é também uma das contribuições deste trabalho.

5.3.1 Dispersão nos coeficientes de uma solução numérica

Para acompanhamento do raciocínio aqui utilizado, considera-se um exemplo simples para ilustrar cada um dos passos.

Seja a *EDO* de 3ª ordem:

$y^{(3)} + y^{(1)} + 10y = 0$, definida no intervalo $[0, 4]$, com as seguintes condições iniciais $y(0) = 1$; $y^{(1)}(0) = 1$; $y^{(2)}(0) = -3$, cuja solução analítica é: $y = e^x \cdot \cos(2x)$.

Essa equação será resolvida por um método numérico Runge-Kutta de baixa ordem (2–3), com a função residente do Matlab `ode23`, com as opções:

```
options = odeset('RelTol', 1e-2, 'AbsTol', 1e-3)
```

Em seguida, a mesma análise será feita usando para a equação o método de Adams-Bashforth-Moulton, cuja função no Matlab é `ode113.m`, com a máxima precisão que ele permite:

```
options = odeset('RelTol', 2.22045e-014, 'AbsTol', 1e-21);
```

Pensando em uma análise que levasse em conta algo em torno de mil pontos, nos dois casos o domínio foi dividido em 989 pontos, para permitir uma integral de Gauss-Legendre com polinômio de grau 47, com 21 subintervalos e mais os dois pontos dos extremos ($21 \cdot 47 + 2 = 989$).

Os passos seguintes serão executados pela opção de análise do *SAM* de forma automática.

Transposição do domínio

Com uma transposição linear, descrita na secção 3.7.2, os pontos do domínio $[a, b]$ são transpostos para o intervalo de Legendre $[c, d] = [-1, 1]$. Nessa transposição, os valores

de y permanecem os mesmos, mas as derivadas serão transpostas obedecendo às equações seguintes.

$$u = T(x) = \left(\frac{d-c}{b-a} \right) x + \left(\frac{bc-ad}{b-a} \right)$$

Considerando-se que $\frac{d-c}{b-a} = k_0$, como descreve a secção 3.7.2, obtém-se:

$$y^{(n)}(x) = y_t^{(n)}(u) \cdot k_0^n \Rightarrow y_t^{(n)}(u) = \frac{y^{(n)}(x)}{k_0^n}.$$

Esse procedimento já transforma também as condições iniciais.

A equação é transformada como descreve a secção 3.7.1, e o *SAM* a apresenta, juntamente com as condições iniciais, também transpostas.

```
=====
Equação no intervalo de trabalho [c; d].
Equação transformada:
.125*D3+.500*D1+10.*D0 = 0
Vetor de cond inics p/ o Método Numérico: y(0); y'(0) ...
0      0      -12
=====
```

Na verdade, como há outras transposições que podem ser consideradas (trigonométrica e irracional), a rotina do *SAM* utiliza a transposição genérica, bem como a sua inversa, deduzidas na secção 3.7, mas a linear é suficiente neste exemplo.

Obtenção dos coeficientes da série de Legendre

A secção 3.3.1 descreve como obter os coeficientes de Legendre para uma expansão em série de uma função $h(u)$, $u \in [-1,1]$, até a ordem n , pela expressão:

$$\text{Eq. \{ 64\}} \quad c_i = \frac{2i+1}{2} \int_{-1}^1 h(u) P_i(u) \cdot du, \quad i = 0, \dots, n.$$

A análise se estende até polinômios de grau 40 (41 coeficientes). A integral de Gauss-Legendre utiliza um polinômio de grau 47 e, portanto, é exata se a função for um polinômio de grau menor do que 93 ($2n-1$), conforme expõe a secção 3.8.

Neste exemplo, como a *EDO* é de terceira ordem, o polinômio atribuído à terceira derivada será de grau 38, para a segunda, grau 39 e 40 para primeira. Após as integrações sucessivas, todos eles terão o mesmo grau que o polinômio que aproxima a função e poderão ser comparados em todos os termos.

Exemplo com o método do Runge-Kutta de baixa ordem

Seguindo com o exemplo do *PVI* do início desta secção, as ordenadas da derivada de maior ordem, neste caso, a terceira, serão o que resta para igualar a equação a zero quando nela se substituem as outras derivadas e a função. Em suma, como descrito na secção 2.3, a derivada de maior ordem é o que lhe resta ser para anular a expressão.

Se a integração sucessiva desta última derivada, considerando as condições iniciais, levar exatamente ao polinômio que aproxima a função, a solução é exata, caso contrário, ter-se-á em mãos a margem de erro.

Dispersão Externa

Convencionou-se aqui denominar-se a dispersão descrita no parágrafo anterior de **Dispersão Externa**, para diferenciá-la da outra que será descrita no tópico seguinte, mas também para salientar que esse cálculo é feito por um juiz externo ao método numérico que obteve os pontos, qualquer que tenha sido esse método.

Além disso, no caso do polinômio obtido pelo *SAM*, essa dispersão vai permitir uma estimativa conservadora do erro máximo que a aproximação acarreta para a função e cada uma de suas derivadas, como descreve a secção 5.2.2.

Gráfico 5. 6. Dispersão dos coeficientes, em relação à derivada de maior ordem, usando o método de integração ode23.m.

Dispersão Interna

Supondo que a solução buscada para a *EDO* seja um polinômio de grau menor ou igual a 40 e que o método numérico de integração seja tão bom que os pontos obtidos estejam muito próximos da solução exata; ao integrar sucessivas vezes os polinômios obtidos para as derivadas, dever-se-ia obter o polinômio que seja a própria função.

Essa coerência interna é um dos primeiros critérios para analisar a confiabilidade de uma solução numérica. Por exemplo, procura-se verificar se nos máximos e mínimos da função no intervalo a 1ª derivada se anula, depois se comparam os máximos e mínimos da 1ª com os zeros da segunda derivada e assim sucessivamente.

Nesse aspecto, a solução polinomial pelo *SAM* tem uma grande vantagem: sua dispersão interna é nula para todos os coeficientes, pois todos são filhotes do mesmo polinômio.

Voltando à solução numérica, quanto melhores forem os pontos, mais próximos serão os coeficientes e menor será dispersão.

No caso do polinômio obtido pelo *SAM*, como a dispersão interna é nula, o erro máximo é avaliado pela dispersão externa; no caso da análise de solução numérica, o erro máximo é avaliado pela máxima dispersão, em cada um, quando se consideram as duas.

Gráfico 5. 7. Coerência entre os coeficientes da função e os das derivadas obtidas pelo ode23.m com baixa precisão.

Como, neste exemplo que descreve o processo, a transformação usada foi linear,

tem-se: $y_t^{(n)} = \frac{y^{(n)}}{k^n}$, e assim:

$$\text{Eq. \{ 65\}} \quad \delta y^{(n)} = \delta y_t^{(n)} \cdot k^n.$$

No caso de transposições não lineares, usam-se as fórmulas genéricas descritas na secção 3.7.1.

```
=====
Majorantes do erro em D0 : 1.59e+000
Majorantes do erro em D1 : 3.16e+001
Majorantes do erro em D2 : 3.36e+003
Majorantes do erro em D3 : 3.36e+005
=====
```

Neste caso, a confiança na função obtida é razoável²⁹, mas as derivadas têm margem de erro muito grande.

Mesmo correndo o risco de ser detalhista demais, é preciso insistir que a margem de erro **não** é o erro esperado, mas sim o limite superior de erro do método analisado. Contudo, a dispersão, esta sim mostra falhas da qualidade do método.

Exemplo usando o Método de Adams

Voltando ao exemplo, considere-se agora o Método de Adams, usando a função ode113.m com as seguintes opções extremas que o Matlab permite:

```
options = odeset('RelTol',2.22045e-014,'AbsTol',1e-21);
```

²⁹ Considerando em termos relativos, pois em $X = \pi$, a função vale 23,14.

Gráfico 5.8. O aumento da precisão do método numérico diminuiu a dispersão externa.

Gráfico 5.9. O aumento da precisão do método reduziu muito também a dispersão interna.

O *SAM* apresenta os erros máximos no domínio.

```
=====
Majorantes do erro em D0 : 3.09e-012
Majorantes do erro em D1 : 4.86e-010
Majorantes do erro em D2 : 6.92e-008
Majorantes do erro em D3 : 7.39e-006
=====
```

5.3.2 Limites da análise

Os coeficientes obtidos na análise são frutos de uma integração de Gauss-Legendre que, embora seja muito precisa em relação a outros processos numéricos de integração, **não** é exata, tem uma margem de erro da ordem de 10^{-12} , que irá se propagar em outros cálculos subsequentes. Além disso, há a própria limitação numérica computacional, principalmente em polinômios cujo grau é muito alto. Por exemplo, o arquivo de ajuda do próprio Matlab já alerta que, ao calcular-se $\sin(\pi)$ não se obtém exatamente 0, mas sim $1,22 \cdot 10^{-16}$.

Considerando os fatores supracitados, dispersões menores do que 10^{-11} podem resultar de erros inerentes aos limites do analisador. Isto é, a análise vai obter um majorante para o limite do erro máximo que o processo de integração pode ter cometido, mas, de fato, os erros máximos são bem menores, podendo até ser nulos.

Com a intenção de testar essa observação, este tópico analisou o conjunto de pontos que se obteria com a solução exata para verificar a dispersão que o método acusaria, conforme ilustram os gráficos seguintes.

Gráfico 5. 10. Dispersão externa acusada pelo método ao analisar os pontos da solução exata.

Gráfico 5. 11. Dispersão interna dos coeficientes obtidos com o método ao analisar a solução exata.

Embora se pudesse ter uma expectativa de erros menores, esses gráficos nos apontam que o método, ao calcular os coeficientes, tem acuidade da ordem de 10^{-12} , ou seja, o usuário deve ser alertado que dispersões de ordens menores do que 10^{-12} podem ser alarmes falsos, diante de uma solução exata.

```
=====
Majorantes do erro em D0 : 3.09e-012
Majorantes do erro em D1 : 4.86e-010
Majorantes do erro em D2 : 6.92e-008
Majorantes do erro em D3 : 7.39e-006
=====
```

Deve-se observar que os valores dos erros máximos são próximos aos obtidos na análise do método de Adams; na verdade, alguns são até maiores.

Haveria como melhorar esses coeficientes afetados por pequenos erros inerentes aos processos numéricos e à limitação do número de casas dos processos computacionais, obtendo-se uma expressão analítica, uma série ou polinômio confiáveis?

Ao longo de toda secção seguinte este trabalho vai procurar mostrar que sim, usando como juíza, para classificar os polinômios candidatos, a própria *EDO*, enquanto o *SAM* vai impondo a evolução.

Outra observação importante se refere ao limite de precisão na construção de um polinômio de Legendre.

Sendo s_k o coeficiente de u^k quando o polinômio de Legendre é escrito na base canônica, tem-se: $P_n(u) = \sum_0^n s_k u^k$.

Como $P_n(1) = 1$ e $1^k = 1 \quad \forall n \text{ e } k \in \mathbb{N}$, obtém-se:

$$P_n(1) = \sum_0^n s_k 1^k \Rightarrow \sum_0^n s_k = 1.$$

Isto é, o somatório dos coeficientes de um polinômio de Legendre escrito na base canônica é 1, qualquer que seja $n \in \mathbb{N}$.

Essa propriedade pode ser um instrumento de teste para verificar até que ordem o Matlab pode produzir um polinômio de Legendre, expresso na base canônica, com os coeficientes corretos, visto ser uma condição necessária para a exatidão do polinômio gerado. Os gráficos seguintes ilustram, conforme a ordem do polinômio, o quanto a soma dos coeficientes na base canônica diferem da unidade. Até a ordem 24 a diferença é rigorosamente nula.

Gráfico 5.12. Diferença em relação à unidade da soma dos coeficientes. A limitação da quantidade de algarismos provoca pequenos erros nos coeficientes de um polinômio de Legendre escrito na base canônica, que aumentam conforme o aumenta grau do polinômio.

Pelo exposto, vê-se que os polinômios de base, para ordens muito altas, não serão rigorosamente ortogonais. Entretanto, é nessa base que o processo vai evoluir, e é com essa mesma base que é feita a transformação para a base canônica. Em outras palavras, a ortogonalidade acelera o processo evolutivo, mas não é imperativa.

5.4 Evolução de uma função expandida em série

Obtida a solução de um *PVI*, pode-se associar aos pontos uma série, por exemplo, polinomial. Em seguida, esse polinômio é testado ao longo do domínio da função e os erros podem ser computados, ponto a ponto.

O próximo passo é determinar a dispersão externa desse candidato, uma vez que, como descrito na secção anterior, sua dispersão interna é nula.

Essa dispersão vai balizar a evolução do cromossomo, limitando diferentemente, para cada gene, a faixa em que lhe é permitido variar. É como se fosse uma seleção natural antes de ele nascer. Se aparecer fora da faixa determinada, já provocaria dispersões maiores e seria descartado na classificação.

Os pontos de partida podem ser obtidos por um método de baixa ordem, ou até mesmo o polinômio nulo, pois, após a primeira geração, já se terá um “envelope” de dispersão, de valores variáveis para cada coeficiente (gene), ainda que possivelmente grosseiro, mas não ilimitado.

Parte-se para uma nova geração, os coeficientes sofrem mutações, acasalam com *crossing-over* e são testados pela equação visando-se uma nova classificação.

Com a intenção de mostrar que, mesmo a partir de uma solução numérica extremamente precisa, como a do método de Adams, é possível reduzir a dispersão dos coeficientes do polinômio, esse será o ponto de partida do exemplo.

5.4.1 Primeira geração

Uma das primeiras escolhas é o grau do polinômio a ser evoluído. Para o caso de *EDO* que envolvam senos, co-senos, tangentes, logaritmos ou qualquer outra função cuja expansão em série é infinita, seja no corpo da equação ou na suspeita de solução, é interessante já se começar com um polinômio de grau alto, por exemplo, 21. Por outro lado, se a expectativa é de uma solução polinomial, parece ser mais interessante ir aumentando o grau gradativamente, pois quanto menor o grau do polinômio, mais rápido as gerações se sucedem, e, eventualmente, podem-se ter margens de erros já aceitáveis com polinômios de grau relativamente baixo.

A cada nova retomada de processamento, o menu de entrada apresenta a opção ao usuário de aumentar, reduzir ou manter o número de coeficientes.

Outra escolha que o programa oferece é o tipo de transposição de intervalo: linear, trigonométrica ou irracional, mas isto na fase de montagem da base de dados. Apenas a primeira vai resultar em um polinômio no domínio original da função, quando se fizer a transposição inversa, como discutido na secção 3.7.

A idéia de ofertar outras transposições além da linear foi considerar que, em alguns casos, a transformação possa resultar em uma equação mais fácil de ser trabalhada, além de permitir o trabalho com domínios ilimitados, como mostram os exemplos 5 e 6 do capítulo 6.

No caso em que o ponto de partida seja uma solução numérica e não de uma série aleatória, a função orthofit.m ³⁰ [Garcia, 2008], baseada nos polinômios de Legendre,

³⁰ [Nougier, 2001], apud D. Garcia [2008].

determina o melhor polinômio que atende aos pontos da última derivada. Sendo q a ordem da equação diferencial e n o grau do polinômio escolhido para aproximar a função, os pontos da última derivada serão aproximados por um polinômio de grau $(n - q)$. O motivo, como foi exposto na secção 5.2.1, é que esse polinômio será integrado q vezes, pela matriz de integração, considerando as condições iniciais, para, enfim, ter-se o primeiro candidato.

Isola-se a última derivada na equação e substitui-se no membro à direita o polinômio candidato. O que restar deveria ser a última derivada. Para comparar, obtêm-se os coeficientes de Legendre do que restou após a substituição, integra-se, usando-se a quadratura de Gauss-Legendre tendo-se em conta as CI , e compara-se o resultado com os coeficientes do candidato, determinando-se a dispersão — o **envelope**. O envelope vai delimitar as mutações das próximas gerações.

Em seguida, valores aleatórios de variações são sorteados para cada gene, confinados pelo envelope, e cria-se uma população com 66 cromossomos³¹ e obtém-se o melhor da população, com o critério LG . A razão da escolha desse tamanho da população é que a população a ser construída terá três grupos de 22 elementos, como descrito a seguir:

- 1º grupo - O cromossomo elemento central é o melhor da geração anterior, ou no caso da primeira geração, a tentativa inicial³². Esse elemento é envolto por 20 cromossomos cujos genes apresentam pequenas variações em relação aos do elemento central (em média, dez um pouco maiores e dez um pouco menores) sorteados pela função de distribuição descrita na secção 4.1.4 – exemplo 3.
- 2º grupo - O cromossomo elemento central é o elemento de saída do mapa descrito na secção 5.5.1., envolto por 20, da mesma forma descrita no item anterior e mais um que é o oposto do central. Esse último é o erro, como se fosse uma mutação, para manter a possibilidade de busca fora do espaço de soluções previsto.
- 3º grupo - O cromossomo elemento central é o elemento de saída do mapa descrita na secção 5.5.2, e também envolto por 21, como descrito no grupo 1.

³¹ Obtiko [1998] recomenda, para o AG em sua versão original, trabalhando com parâmetros discretos, uma população entre 20 e 40 cromossomos. Obtiko também destaca que, em geral, populações muito grandes não aceleram a convergência e tornam o processamento muito mais lento.

³² No caso de não ter sido um método numérico para, a partir dele, obter-se os coeficientes de partida, a tentativa inicial pode ser o polinômio nulo ou um polinômio aleatório.

Está quase montada a primeira geração. Falta ainda a imposição de que as condições iniciais se mantenham, como expõe a secção 3.6.2. O *SAM* aplica agora essa condição para todos os cromossomos da 1ª geração, e continuará aplicando a todos os cromossomos das gerações futuras.

Esse processo será repetido a cada pacote de gerações, sendo o melhor da última geração do pacote o novo pivô da nova população.

Se for um caso em que os *MN* não conseguem inicializar, parte-se do polinômio nulo³³. A preservação das *CI*, aplicada a esse candidato, já vai alterá-lo. Em seguida, o envelope é construído, assim como a primeira geração, e parte-se para o processo evolutivo, com as mutações e *cross-over*, tal como no caso anterior.

Entre os pacotes, a evolução se processa descartando-se os 20% piores de cada geração, recombina-se com *cross-over* os restantes e preservando-se os 4 melhores da geração (elitismo). Além disso, acontecem pequenas mutações confinadas pelo envelope em 10% do total de genes. Após todo esse processo, é aplicada a função “ajuste” a todos os cromossomos, que preserva as condições iniciais, como descrito na secção 3.6.2.

É importante ressaltar que o processo é capaz de evoluir mesmo nos casos de matriz de massa singular [Shampine & Reichelt, 1997], quando então os *MN* não conseguem inicializar, lembrando que isto é possível porque as abscissas da *QGL* não incluem os extremos do intervalo.

Além disso, o uso das abscissas da *QGL* acaba se tornando uma precaução contra o “fenômeno de Runge”³⁴ [Berrut & Trefethen, 2004], [Boyd, 1992]. Uma situação em que o uso de abscissas com espaçamento uniforme na interpolação polinomial faz o erro quadrático total ir crescendo à medida que cresce o grau do polinômio de aproximação, para alguns tipos de funções.

Recomenda-se executar o programa para apenas uma geração e avaliar as condições de erro iniciais, como ilustram os gráficos seguintes, nos quais se considera o mesmo *PVI* da secção 5.3.1, para o qual será associada uma solução polinomial de grau 21.

³³ Opcionalmente, pode partir também de um polinômio gerado aleatoriamente.

³⁴ Berrut & Trefethen [2004, p. 509] comentam sobre o uso das abscissas da *QGL* ou dos nós de Chebyshev para evitar esse problema. O fenômeno foi discutido por Karl Runge em 1901.

Gráfico 5. 13. Dispersão externa dos coeficientes na 1ª geração em três tentativas diferentes.

Gráfico 5.14. Erro (resíduo RS) ao substituir o polinômio na EDO. A imposição das CI (poly by SAM) aumentou um pouco o erro quadrático total, mas ele será reduzido no processo evolutivo.

Vale lembrar que o erro apresentado no gráfico acima é o resíduo decorrente da substituição do polinômio que aproxima a função na equação diferencial, ao longo do intervalo de trabalho $[-1, 1]$, transposto para o domínio original.

Com apenas uma geração, mas impondo-se as condições iniciais, diminuíram os erros máximos tanto na função quanto em suas derivadas, embora o erro quadrático total tenha aumentando um pouco. Compare com o resultado ao final da secção 5.2.

Para esta primeira geração, a raiz quadrada do produto do erro máximo pela raiz da integral do erro quadrático, dividida pelo tamanho do intervalo (critério LG), foi $1,04 \cdot 10^{-7}$.

$$\text{Simbolicamente: } (\text{erro})_{LG} = \sqrt{(\text{erro})_{\text{máx}} \cdot \frac{\sqrt{\int_{-1}^1 (\text{erro})^2 \cdot du}}{2}}$$

Merece esclarecimento o fato de o gráfico da dispersão não ser o mesmo obtido na secção 5.2. O motivo é que, embora esse seja o resultado da primeira geração, ele se refere ao melhor cromossomo. Como a população construída, constrita pelo envelope, tem 66 cromossomos, com pequenas variações obtidas pela “roleta viciada”, às vezes, já no primeiro torneio alguns superam o polinômio inicial, pivô dessa geração.

5.4.2 Seleção dos melhores

Cada um dos cromossomos será avaliado ao longo de todo intervalo para obter-se o erro, também ao longo do intervalo. Se o intervalo é dividido em, por exemplo, 126 pontos, cada polinômio será calculado 126 vezes, e os valores de erro encontrados irão para um vetor. Isso demandaria muito tempo de processamento para cada geração, além de que, ao trabalhar simbolicamente com a equação, os processos seriam muito mais lentos.

Entretanto, analisado um candidato, os cálculos do próximo serão muito semelhantes, mudando apenas os valores dos coeficientes. A solução encontrada neste trabalho foi, a partir das ferramentas simbólicas, antes de começar a avaliação dos candidatos e das gerações, construir uma *string* de instruções, não simbólicas, que o processamento deveria executar para avaliar o erro em cada candidato. Dessa forma, dado um candidato, o comando: `eval (string)` fornece os erros desse candidato ao longo do intervalo rapidamente³⁵. Para o exemplo iniciado na secção 5.2, trabalhando com um polinômio de grau 21, com transformação linear, a *string*, que fica armazenada na variável “*rvc*”, contém o texto de 3 páginas. Essa operação é feita com os coeficientes escritos na chamada forma racional³⁶, reduzindo drasticamente os erros de truncamento. Essa parte do processamento independe das ferramentas simbólicas e pode ser feita em qualquer computador que tenha instalado o Matlab, se for para se pensar em processamento distribuído.

Depois que toda a população é avaliada, faz-se a classificação com o critério *LG* e parte-se para uma nova geração, descartando-se os 20% piores. Nessa nova geração, os melhores terão maior probabilidade de acasalamento, conforme detalha a secção 4.1.4 exemplo 5, cujos “filhotes” reporão primeiro os descartados na geração anterior, para manter a população, e, a seguir, serão substituídos por seus descendentes. Então, uma nova classificação é feita para esta nova geração.

Após um determinado pacote de gerações, a dispersão é reavaliada, o envelope se estreita, uma nova geração é montada, e recomeça o processo.

5.4.3 Mutações, *Crossing-over* e Elitismo

Feito o *ranking* de uma geração, os 20% piores são descartados, os cromossomos restantes vão acasalar, com o *cross-over* acontecendo em um *locus* sorteado com a Roleta

³⁵ O ganho em tempo de processamento foi da ordem de centenas de vezes.

Viciada, primeiro repondo os descartados. Depois, os acasalamentos continuam até que se reponha a população inteira. O 4 melhores da geração anterior são repostos nessa nova população para evitar que o processo possa divergir. Com esse recurso, na pior das hipóteses, a evolução estaciona.

Exemplo

Para a equação $y^{(3)} + y^{(1)} + 10y = 0$, com $x \in [0, 4]$, sob as condições iniciais $\vec{y}_0 = [1, 1, -3]$ cuja solução analítica é $y(x) = e^x \cos(2x)$, usando-se uma série de ordem 21, após 1000 gerações, tendo como ponto de partida o método de Adams, obtiveram-se os gráficos seguintes³⁷.

Gráfico 5.15. Dispersão externa dos coeficientes do polinômio de aproximação após 1000 gerações de evolução. A dispersão interna é nula.

³⁶ A melhor razão entre dois inteiros que representa o número. Por exemplo, nesta representação, a fração 1/3 não terá erros de truncamento.

³⁷ Arquivo TRB-543-22-L-N.mat, cuja base de dados é B-L-543-N.mat.

Gráfico 5. 16. Resíduo (RS) obtido ao se substituir o polinômio na *EDO*.

```
=====
Erros máximos no domínio:

Majorante do erro em D0 : 9.01e-010
Majorante do erro em D1 : 2.59e-009
Majorante do erro em D2 : 1.52e-008
Majorante do erro em D3 : 2.56e-007
=====
```

Os gráficos seguintes visam conferir as margens de erro apresentadas pelo programa, comparando-se o polinômio obtido com a solução analítica.

Gráfico 5.17. Erro do polinômio de aproximação em relação aos valores da função e derivadas.

Os gráficos dos erros, comparados aos valores máximos de erro calculados, mostram que de fato o erro ao longo do intervalo é sempre menor do que o majorante apresentado pelo *SAM*; e, além disso, como o cálculo do majorante do erro é extremamente conservador, o erro máximo observado, na função e em cada uma das derivadas, é, pelo menos, cerca de dez vezes menor. Isto é, as margens de erro calculadas, embora pequenas, são de segurança. A secção final do anexo 2 apresenta o polinômio obtido na sua forma racional. Essa forma é importante, pois vai preservar a precisão alcançada pelo *SAM* no processo evolutivo.

5.5 Recorrência na resolução de um *PVI*

Obtido um candidato à solução do *PVI*, a secção 5.2.1, no tópico **dispersão externa**, descreve como compará-lo com o que seria a última derivada, integrada q vezes, onde q é a ordem da equação diferencial.

Os coeficientes da série de Legendre que aproximam a função até grau k podem ser entendidos como um vetor k -dimensional $\vec{t}$, ao qual são ortogonais todos os polinômios de Legendre de ordem maior do que k , pois a base é ortogonal.

Resumidamente, dado um vetor de entrada, a seqüência de operações aqui denominada, arbitrariamente, de mapa $\varphi(\vec{t}_1)$, nos fornece um vetor de saída $\vec{t}_2$.

Se o vetor $\vec{t}$ for a solução exata, então: $\varphi(\vec{t}) = \vec{t}$.

O ponto em que deseja chegar é, então:

$$\text{Eq. \{ 66\}} \quad \begin{cases} \varphi(\vec{t}_1) = \vec{t}_2 \\ \vec{t}_2 = \vec{t}_1 \end{cases}.$$

O sistema sugere um processo de recorrência, de forma que:

$$\text{Eq. \{ 67\}} \quad \begin{cases} \varphi(\vec{t}_n) = \vec{t}_{n+1} \\ \vec{t}_n = \vec{t}_{n+1} \end{cases}$$

Resolver esse sistema por um processo iterativo se assemelha à busca de soluções no mapa logístico.

Conforme descreve a secção 5.2.1, a partir de uma série candidata à resolução da EDO, com seu vetor de coeficientes (vetor de entrada $\vec{t}_n$) obtém-se um vetor de coeficientes (vetor de saída $\vec{t}_{n+1}$), que, se for igual ao vetor candidato, é a solução exata.

5.5.1 Realimentação com a saída

A idéia neste processo é implementar exatamente o que descreve o sistema:

$$\text{Eq. \{ 68\}} \quad \begin{cases} \varphi(\vec{t}_n) = \vec{t}_{n+1} \\ \vec{t}_n \leftarrow \vec{t}_{n+1} \end{cases}.$$

A partir de $\vec{t}_n$ obtém-se $\vec{t}_{n+1}$ e, em seguida, põe-se $\vec{t}_n \leftarrow \vec{t}_{n+1}$ e se refaz $\varphi(\vec{t}_n) = \vec{t}_{n+1}$, sucessivas vezes, até que o processo se estabilize com os vetores de entrada e saída bem próximos, ou, no caso da solução exata, iguais.

Exemplo

EDO: $y^{(3)} + y^{(2)} \cdot y = 0$, no domínio $[0; 5]$ com as condições iniciais $\vec{y}_0 = [0; 0; 0,4699]$, que não tem solução analítica expressa por funções elementares.

O polinômio candidato será de grau 21, ou seja, com 22 coeficientes.

O ponto de partida foi um polinômio nulo. Em cada iteração, o erro (critério LG) avaliado no intervalo foi calculado.

Após 37 iterações, com cerca de 15 s de processamento, o erro (LG) no PVI estabilizou em $1,466 \cdot 10^{-7}$, e os coeficientes obtidos foram usados para construir a função de aproximação, cujos gráficos são apresentados a seguir.

Gráfico 5.18. Dispersão externa dos coeficientes da série que aproxima a função.

Gráfico 5.19. Gráficos da função e derivadas obtidas pelo *SAM*, comparados com o obtido pelo *ode113*, com a opção de precisão máxima. Neste limite de resolução, os gráficos são indistinguíveis, pois o erro é menor que milésimos de pixel.

5.5.2 Realimentação com a média

Outra idéia, tentando aproximar o vetor de entrada e o de saída, foi considerar como novo vetor de entrada a simples média aritmética entre $\vec{t}_n$ e $\vec{t}_{n+1}$, isto é:

$$\text{Eq. \{ 69\}} \quad \begin{cases} \varphi(\vec{t}_n) = \vec{t}_{n+1} \\ \vec{t}_n \leftarrow \frac{\vec{t}_{n+1} + \vec{t}_n}{2} \end{cases}$$

Nos casos analisados, o que se verificou foi que, quando o processo descrito em 5.4.1 converge, este aqui converge também para o mesmo vetor. Entretanto, há casos em o processo de 5.4.1 atinge um estado de alternância entre dois valores, ainda com erro muito alto. Nesses casos, o critério de realimentar com a média entre a entrada e saída faz a recorrência ultrapassar essa estagnação e convergir para valores com erros bem menores.

O quinto exemplo do Capítulo 6 — Resultados e Análise — utiliza essa recorrência.

5.6 Aproximação de uma função

Aproveitando-se o processo evolutivo já codificado e outras funções do *SAM*, tornou-se possível criar uma opção no menu de entrada para o usuário aproximar uma função $g(x)$ em determinado intervalo, usando o critério *LG* em conjunto com os polinômios de Legendre. Como mostra a secção 10 do Apêndice, essa aproximação é melhor que a série de Taylor e garante erros locais menores do que se fosse utilizada simplesmente a expansão de Legendre.

Arfken & Weber [1995, pp. 787-788] apontam que os polinômios de Chebyshev do tipo I fazem, em parte, essa tarefa, uma vez que sua função peso, no intervalo $[-1,1]$, é $\frac{1}{\sqrt{1-x^2}}$. Ou seja, o peso do erro nos extremos do intervalo tende a infinito. Dessa forma, ao se utilizar desses polinômios, o erro é empurrado para os pontos centrais³⁸. Entretanto, se o erro no ponto central for alto, a função está mal aproximada e, supostamente, haveria erros expressivos nos extremos. Embora muito atrativa, a comparação entre essa classe de polinômios e o critério *LG* será deixada para trabalhos futuros³⁹. À primeira vista, para obtenção do primeiro polinômio candidato, Chebyshev tipo I parece ser uma boa escolha; mas, no processo evolutivo-adaptativo, todos os pontos do intervalo precisam ter o mesmo peso, critério este que não é o de Chebyshev tipo I.

O *SAM* monta a equação $D_1 - \frac{dq(x)}{dx} = 0$ no intervalo $[a,b]$, onde $q(x)$ é a função escolhida para ser aproximada e fornecida pelo usuário, como se fosse ela o *PVI*, em que $y_0 = q(a)$.

Em seguida, o *SAM* transpõe linearmente O *PVI* para o intervalo de trabalho $[-1,1]$, faz a expansão de Legendre para a função com o número de coeficientes escolhido pelo usuário, chamando-a de D_0 , e pode começar o processo evolutivo, se essa for a escolha; caso contrário, ele obtém simplesmente a série de Legendre correspondente à função.

³⁸ O peso do erro em $x = 0$ é apenas a unidade, o menor de todo o intervalo.

³⁹ Recentemente, Babolian & Fattahzadeh [2007] utilizaram essa classe de polinômios construindo sua matriz de integração, aprimorando um método proposto por Shen [1994] para resolução de Equações Diferenciais Lineares de alta ordem.

5.7 Soluções periódicas

Eventualmente, mesmo sem conhecer a solução de um *PVI*, é possível mostrar que se $f(x)$ é solução, então $f(x+T)$ também é solução. Se isso acontecer, a solução é periódica, e a menor constante positiva que satisfaça a condição anterior é o período da função.

Por exemplo, considerando a equação do oscilador harmônico:

$y^{(2)}(t) + \omega^2 y(t) = 0$, se $y(t)$ é solução, e existe T tal que $y(z) = y(t+T) = y(t)$, então $y(z)$ é também solução e ela é periódica de período T , se T for o menor⁴⁰ número real positivo que satisfaz a condição descrita.

De fato, como $\frac{dz}{dt} = \frac{dt}{dz} = 1$, tem-se: $\frac{d^2 y}{dz^2} = \frac{d}{dz} \left(\frac{dy}{dz} \right) = \frac{d}{dz} \left(\frac{dy}{dt} \frac{dt}{dz} \right) = \frac{d^2 y}{dt^2}$.

Assim: $y^{(2)}(z) + \omega^2 y(z) = y^{(2)}(t) + \omega^2 y(t)$, o que se queria provar.

Há no *SAM* uma pequena rotina, razoavelmente simples, que permite estender o resultado para domínios ilimitados se a solução for periódica, preservando a forma simbólica das funções, ou seja, sem perda de precisão.

Sendo k o natural inferior mais próximo da razão $\frac{x}{T}$, onde T é o período da função, a variável x da função e das derivadas é substituída por $z = x - kT$.

No *SAM* a obtenção de k foi feita com o comando `floor(x/T)` do Matlab.

A opção para estender o domínio da solução é apresentada na saída do processamento, quando o *SAM* apresenta o polinômio de aproximação e os gráficos.

⁴⁰ Isso porque se T satisfaz a condição, todos os múltiplos inteiros também satisfazem.

Capítulo 6. Resultados e Análises

O algoritmo foi implementado para vários *PVIs*, principalmente não lineares, até a ordem 4.

Os resultados são apresentados nos 11 exemplos a seguir. Um exemplo sobre análise de qualidade de um método numérico de integração já foi apresentado na secção 5.3.1 e um exemplo sobre aproximação de funções é apresentado no Anexo, secção 7.

Em cada caso, é explicitado o *PVI* e as margens de erro do polinômio de aproximação e derivadas. Conforme o problema, diferentes graus foram usados para os polinômios.

6.1 Exemplos

A subsecção 2.2.1 expõe porque uma boa aproximação para a função pode não ser adequada à equação diferencial, já que a aproximação para a solução deve levar em conta a relação que guardam entre si a função, as derivadas e a variável independente conforme impõe o *PVI*.

Com isso em mente, em cada exemplo apresentam-se, além do gráfico do erro na função, também os gráficos de erro nas derivadas, no domínio original do *PVI*.

O erro em cada ponto do intervalo é o resíduo (*RS*) que se obtém ao substituir o polinômio na equação diferencial sob a forma $E(y^q, y^{q-1}, \dots, y, x) = 0$

Nos casos em que não há solução analítica expressável por uma combinação finita de funções elementares, a solução obtida pelo *SAM* é comparada com os pontos obtidos pelo método de Adams com a precisão extrema permitida pelo Matlab, quando ele consegue inicializar.

Exemplo 1.

A equação diferencial de Bessel de ordem zero, que corresponde ao *PVI* [Babolian & Fattahzadeh, 2007]:

$$xy^{(2)} + y^{(1)} + yx = 0 \quad x \in [0, 1], \text{ com } \vec{y}_0 = [1, 0], \text{ cuja solução analítica é}$$

$$J_0(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k!)^2} \left(\frac{x}{2}\right)^{2k}, \text{ foi proposta para o SAM aproximar com 26 coeficientes,}$$

partindo do polinômio nulo.

Como a matriz de massa nessa equação é singular⁴¹ [Shampine & Reichelt, 1997], o método de Runge-Kutta não consegue inicializar, e tampouco o de Adams, que depende do primeiro.

Antes de entrar no processo evolutivo, de alto custo computacional, é conveniente repetir o processo de aprendizado várias vezes, tomando-se como nova entrada do processo a média entre os vetores de entrada e de saída, buscando candidatos melhores entre eles. No SAM esta é a 2ª opção das recorrências, secção 5.5.2, que levará, se convergir, ao melhor candidato conforme o critério de erro *LG*.

Neste exemplo, esse processo iterativo já foi suficiente para obter uma aproximação polinomial com margens de erro muito baixas, mesmo assim, o processo evolutivo ainda conseguiu reduzi-las, com 200 gerações⁴².

Gráfico 6. 1. Resíduo que se obtém ao substituir o polinômio de aproximação na ED.

⁴¹ Ao tentar isolar a derivada de maior ordem, no extremo inferior do intervalo ter-se-á uma divisão por zero. Como o *PVI* é linear, há métodos matriciais, baseados em polinômios ortogonais para obter funções aproximadas, reduzindo a equação diferencial a um problema de álgebra linear. Nesses métodos, o coeficiente que multiplica a derivada de maior ordem, assim como os outros, são matrizes.

⁴² Com aproximadamente 5 min de processamento.

Gráfico 6.2. Comparativo entre a função e suas derivadas com as do polinômio de aproximação. Para esta resolução gráfica, eles são indistinguíveis.

Gráfico 6.3. Dispersão relativa (externa) dos coeficientes.

Gráfico 6. 4. Erros na função e nas derivadas ao longo do domínio após 200 gerações.

Majorantes dos erros calculados pelo *SAM*:

Majorante do erro em D0 : 3.86e-014
 Majorante do erro em D1 : 5.76e-013
 Majorante do erro em D2 : 1.36e-010

Para este mesmo *PVI*, com polinômios de Chebyshev até ordem 5, Babolian [Babolian & Fattahzadeh, 2007] obteve erros máximos da ordem de 10^{-3} , com resultados um pouco melhores que a expansão pura de Legendre para pontos próximos aos extremos do domínio, mas ele não compara os pontos mais centrais, que têm menos peso no critério de erro de Chebyshev. Com esse grau de polinômio, o obtido⁴³ pelo *SAM* tem erro máximo da ordem de 10^{-6} , com erro nulo na abscissa inicial, isto é, em termos de erro máximo (L^∞), a série evoluída é bem melhor que a de Legendre ou a de Chebyshev.

Exemplo 2.

Considere o *PVI*:

$$xy^{(1)} - y - x^2 \cdot \cos(x) = 0, \quad x \in [0, 8], \quad \text{com } y(0) = 0.$$

⁴³ Com 1000 gerações e cerca de 3 min de processamento.

Como no exemplo anterior, este *PVI* também tem matriz de massa singular.

Por substituição direta, pode-se verificar que $y = x \cdot \sin(x)$ é solução deste *PVI*.

No entanto, partindo de um polinômio aleatório, usando também o processo iterativo por 120 vezes (~ 30 s de processamento) e depois a evolução com 200 gerações (~ 10 min), o *SAM* encontrou outra solução, cuja derivada na origem não é nula, conforme mostram os gráficos seguintes.

Gráfico 6. 5. Como essa série passou por um processo evolucionário mais longo, o critério *LG* fez com que o erro ficasse mais distribuído ao longo do intervalo.

Gráfico 6. 6. Solução para o *PVI* obtida pelo *SAM* e a solução analítica após as considerações sobre o valor de k (vide a discussão sobre a multiplicidade de soluções ao final do exemplo).

Gráfico 6. 7. Dispersão relativa dos coeficientes para o *PVI* proposto.

Supondo que a solução encontrada seja uma função $h(x) = x \cdot \sin(x) + g(x)$, tal que $g(0) = 0$ e $g^{(1)}(0) \neq 0$, pode-se tentar determinar essa função g .

Fazendo-se a substituição na *ED* obtém-se:

$$x \sin(x) + x^2 \cos(x) + xg^{(1)}(x) - x \sin(x) - g(x) - x^2 \cos(x) = 0, \text{ ou seja,}$$

$xg^{(1)}(x) = g(x)$, que pode ser resolvida por separação de variáveis, considerando-a válida em um intervalo $[a, x] \in \mathbb{R}$.

$$\frac{dg}{g} = \frac{dx}{x}, \quad \frac{g(x)}{g(a)} = \frac{x}{a} \Rightarrow g(x) = \frac{g(a)}{a} x.$$

Pondo-se $\frac{g(a)}{a} = k$, então, simplesmente, $g(x) = kx$.

No entanto, se $a = 0$ e $g(a) = 0$, o problema se defronta com uma indeterminação e qualquer k real satisfaz a equação. Em outras palavras, não é válido o teorema de Lipschitz (secção 2.3) e a solução não é única. Aqui, este é o caso, já que $y(0) = 0$.

Como $g^{(1)}(x) = k$, pode-se, para verificar a margem de erro da particular solução encontrada pelo *SAM*, determinar-se k por $k = y^{(1)}(0)$.

Neste processamento obteve-se, com seis significativos: $k = -16,8052$.

Pode-se agora verificar as margens de erro, tendo como solução exata a função:

$$f(x) = x \sin(x) + kx.$$

Gráfico 6.8. Margens de erro, em relação à solução exata, do polinômio de aproximação e sua derivada.

Majorante do erro em D0 : 1.09e-010
Majorante do erro em D1 : 1.03e-010

Exemplo 3.

O PVI: $\frac{1}{216} \left(\frac{d^2 y}{dx^2} \right)^3 + 2 \frac{dy}{dx} - y - 6x^2 + 1 = 0$, no intervalo $[0; 1]$

com as condições iniciais: $\vec{y}_0 = [1; 0]$, cuja solução exata é $y = x^3 + 1$, é outro caso em que os *MN* consagrados não conseguem dar a partida, aparentemente, pelas mensagens de erro do Matlab, pelo fato de o sistema de equações na condição inicial envolver soluções com parte imaginária.

Neste caso, como número de coeficientes é muito pequeno, após a recorrência 2, em 1 min foram processadas 1000 gerações e o *SAM* encontrou a solução exata.

Gráfico 6. 9. Resíduo resultante da substituição do polinômio de aproximação na ED.

Gráfico 6. 10. Concordância entre a função e derivadas exatas e as obtidas pelo SAM. Neste caso ele determinou a solução exata.

Gráfico 6. 11. Dispersão relativa dos coeficientes. Nesse caso, essa dispersão é na verdade fruto de um “ruído” da QGL, que terá reflexos nos majorantes de erro previstos.

Gráfico 6. 12. Erro entre o polinômio de aproximação e derivadas comparados com a solução exata.

Majorante do erro em D0 : 1.42e-014
 Majorante do erro em D1 : 2.85e-014
 Majorante do erro em D2 : 4.46e-014

Exemplo 4.

O PVI de 4ª ordem, não linear

$$D_4 + D_3 \cdot D_1 - \frac{D_0}{2} + 3 \cdot \sin^2 x + (4D_0 + 3) \cos x + (D_1 - \sin x)^2 + \frac{D_2}{2} = 0, \text{ no domínio}$$

$[0, \pi]$, sob condições iniciais: $[0, 0, 2, 0]$ tem a solução analítica: $f(x) = x \cdot \sin x$.

Para por a prova a capacidade do SAM em resolvê-la, usou-se uma série com 22 coeficientes com a transposição linear do domínio original para o de trabalho.

As recorrências não se mostraram funcionais para esta equação no estágio inicial. Dessa forma, primeiramente o polinômio foi posto no processo de evolução, de forma a inicializar a recorrência com erros mais baixos. Após 1000 gerações, a recorrência 2 foi aplicada e as margens de erro, tanto na função quanto nas derivadas se colocaram na ordem de 10^{-11} . Com o envelope constrito dessa forma, retomou-se a evolução por mais 200 gerações.

Os gráficos e margens de erros obtidos foram os seguintes.

Gráfico 6.13. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.

Gráfico 6.14. Comparativo entre a função e suas derivadas com as do polinômio de aproximação.

Gráfico 6.15. Dispersão relativa (externa) dos coeficientes.

Gráfico 6.16. Erros em relação à função e às derivadas.

Majorante do erro em D0 : 2.89e-012
 Majorante do erro em D1 : 6.38e-012
 Majorante do erro em D2 : 4.76e-011
 Majorante do erro em D3 : 9.57e-010
 Majorante do erro em D4 : 2.38e-008

Gráfico 6.17. Comparativo dos resíduos entre o melhor polinômio — obtido por polinômios ortogonais — com o polinômio evoluído pelo SAM.

Exemplo 5.

Aqui, vai-se considerar o uso da transformação trigonométrica (não-linear), apenas para mostrar a aplicabilidade em alguns casos.

O PVI é

$$D_4 + \frac{4D_1(3D_1x(x^4 - 1) - 9x^2 + 2)}{(1 + x^2)^3 \tan(x)} = 0, \text{ com as condições iniciais}$$

$$\vec{y}_0 = [0 \ 0 \ 2 \ 0], \text{ cuja solução analítica é: } y = \tan^2(x)$$

Partindo de uma série nula, com sete coeficientes, e utilizando a recorrência, já se obteve alto grau de precisão com 200 iterações (cerca de 30 s). Em seguida, a série entrou no processo evolutivo, por 600 gerações (~ 5 min).

Gráfico 6.18. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.

Gráfico 6.19. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação.

Gráfico 6. 20. Dispersão relativa (externa) dos coeficientes.

Gráfico 6. 21. Erros na função e derivadas. Neste caso os erros obtidos são um pouco maiores que os majorantes devido à limitação de dígitos do processador.

```

Majorante do erro em D0 : 1.26e-018
Majorante do erro em D1 : 2.43e-017
Majorante do erro em D2 : 2.34e-016
Majorante do erro em D3 : 1.91e-016
Majorante do erro em D4 : 1.96e-014

```

Exemplo 6.

Este exemplo, construído de forma reversa, ou seja, montando o *PVI* a partir da solução⁴⁴, consta somente para mostrar como, em alguns casos, pode ser aplicável a transformação irracional — uma das opções do *SAM*.

$$\text{O PVI} \quad D_3 D_1 - D_2 - \frac{3x(x^2 + 1)^{\frac{5}{2}} + 12x^2 - 3}{(x^2 + 1)^5} = 0, \text{ sob as condições iniciais}$$

$\vec{y}_0 = [0 \ 1 \ 0]$, cuja solução analítica é $y(x) = \frac{x}{\sqrt{1+x^2}}$, foi implementado.

A partida com o método de Adams na precisão extrema, após a transposição de domínios, não se mostrou funcional. Usou-se então a opção de menor precisão:

```
options = odeset('RelTol',1e-2,'AbsTol',1e-3);
```

Trabalhando com 4 coeficientes, em 200 gerações o *SAM* convergiu para a solução exata.

Os gráficos seguintes de análise de resultados visam apenas mostrar que margens de erro muito baixas (abaixo de 10^{-14}) podem ser fruto da limitação dos processadores ou da própria *QGL* que, embora muito precisa, como todo processo numérico, tem limitações.

⁴⁴ De certa forma, é um exemplo *ad hoc*, mas a intenção é mostrar a funcionalidade da opção.

Gráfico 6.22. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.

Gráfico 6.23. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação.

Gráfico 6. 24. Dispersão relativa (externa) dos coeficientes.

Gráfico 6. 25. Erros em relação à função e às derivadas.

Majorante do erro em D0 : 2.17e-018
 Majorante do erro em D1 : 3.64e-017
 Majorante do erro em D2 : 2.62e-016
 Majorante do erro em D3 : 7.95e-016
 Majorante do erro em D4 : 4.18e-014

Exemplo 7.

O *PVI* $D_3 + D_2 \cdot D_0 = 0$, no intervalo $[0, 5]$ com as condições iniciais

$y(0) = 0$, $y'(0) = 0$ e $y''(0) = 0.4699$, não tem solução analítica.

Como ponto de partida utilizou-se uma série nula com com 22 coeficientes e, em seguida, aplicou-se a primeira alternativa de recorrência, que estacionou em margem de erro muito baixa ($LG \sim 10^{-9}$) após 43 iterações (~ 30 s). Depois disso, foi aplicada a evolução por 400 gerações (~ 15 min). Como este *PVI* não tem solução analítica, os resultados (erros) foram comparados aos obtidos pelo método de Adams com a máxima precisão disponível no Matlab 7.

Gráfico 6. 26. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na *ED*.

Gráfico 6. 27. Comparativo entre a solução de Adams (máxima precisão) e suas derivadas com as do polinômio de aproximação.

Gráfico 6. 28. Dispersão externa relativa dos coeficientes.

Gráfico 6. 29. Erros em relação à função e às derivadas.

Majorante do erro em D0 : 9.29e-009
 Majorante do erro em D1 : 3.93e-008
 Majorante do erro em D2 : 2.87e-007
 Majorante do erro em D3 : 3.59e-006

Exemplo 8.

Esta equação diferencial foi proposta e resolvida por Cheb-Terrab [2002] como um desafio nas equações diferenciais. Este exemplo visa destacar que, mesmo para equações diferenciais lineares, há problemas de difícil solução, mas que podem ser abordados pelo *SAM*.

A equação diferencial de 3ª ordem:

$$y''' = \frac{(6 - 4x + x^2)y'}{(x-1)^2} - \frac{(1 - 6x + 15x^2 - 20x^3 + 15x^4 - 6x^5 + x^6 + 4(e^x)^3 - (e^x)^3 x)y}{(e^x)^3(x-1)^3}$$

cuja solução analítica é:

$$y = \frac{e^{x(e^{-x}+1)}}{x-1} C_1 + \frac{e^{-1/2 x(e^{-x}-2)} \sin(1/2 \sqrt{3} x e^{-x})}{x-1} C_2 + \frac{e^{-1/2 x(e^{-x}-2)} \cos(1/2 \sqrt{3} x e^{-x})}{x-1} C_3$$

Sob as condições iniciais⁴⁵ $\vec{y}_0 = [-2 \ -5.366 \ -12.366]$, as constantes C_1 , C_2 e C_3 todas são iguais a 1. Aqui, o algoritmo usou um polinômio de 21º grau, face às possíveis particularidades da função. Uma vez que a função tem uma singularidade em $x = 1$, o domínio $[0; 0.6]$ foi estipulado para a análise. Após o uso da 2ª opção de recorrência, com 27 iterações o erro LG atingiu a ordem de 10^{-7} . Em seguida *SAM* processou a evolução por 200 gerações (~ 8 min).

Gráfico 6.30. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na *ED*.

⁴⁵ Há mais casas decimais nas *CI*, considerando a sensibilidade desse *PVI* a esses parâmetros. Explicitamente, as *CI* são: $[-2 \ -5.366025403784438 \ -12.366025403784437]$.

Gráfico 6.31. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação.

Gráfico 6.32. Dispersão relativa (externa) dos coeficientes.

Gráfico 6. 33. Erros em relação à função e às derivadas.

Majorante do erro em D0 : $5.17e-012$
 Majorante do erro em D1 : $2.25e-010$
 Majorante do erro em D2 : $3.67e-008$
 Majorante do erro em D3 : $6.71e-006$

Os majorantes de erro ainda não prevêm corretamente o erro máximo, somente a ordem de grandeza, pois o processo evolutivo trabalhou com poucas gerações. Isso significa que há dispersão externa ainda significativa nos coeficientes de ordem maior do que a do polinômio de aproximação utilizado.

De qualquer forma, as margens de erro são tão baixas que parte da imprecisão pode ser atribuída aos limites da *QGL* e do próprio processador. Deve ser observado também que, em termos relativos, as margens de erro são extremamente baixas, considerando os valores que assume a terceira derivada no trecho final do domínio.

Exemplo 9.

Para este exemplo, o algoritmo processou a equação diferencial que se obtém para um pêndulo simples com ângulo inicial qualquer, usando-se um polinômio de grau 40. A condição inicial escolhida foi tal que inicialmente o pêndulo estava quase na posição crítica,

caso que não permite aproximações para pequenos ângulos. A partida foi dada com o método de Adams e, depois da recorrência 2, o polinômio foi evoluído com 20 gerações.

ED: $y^{(2)} + \sin(y) = 0$. Condições iniciais: $y(0) = 3.1$ e $y^{(1)}(0) = 0$.

Domínio: $[0; 9]$, cerca de $\frac{1}{4}$ de período.

Gráfico 6.34. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED. O fato de o erro estar regularmente distribuído no domínio mostra a proximidade da estagnação evolutiva com o critério LG.

Gráfico 6.35. Comparativo entre a solução do MN com as do polinômio de aproximação.

Gráfico 6. 36. Dispersão relativa (externa) dos coeficientes.

Gráfico 6. 37. Erros em relação à função e às derivadas, em relação ao método de Adams.

Majorante do erro em D0 : 1.91e-005
 Majorante do erro em D1 : 2.81e-004
 Majorante do erro em D2 : 1.25e-002

Exemplo 10.

Este exemplo vai comparar o polinômio obtido pelo *SAM* com a série de Taylor relativa à simples equação $y' = y$, no intervalo $[-0,5; 0,5]$, cuja solução é $y = \exp(x)$, sob a condição inicial $y(-0.5) = \exp(-0.5)$, com um polinômio de 10º grau.

À primeira vista, o polinômio procurado seria a série de Taylor para $y = \exp(x)$, em torno de $x = 0$, mas há polinômios que produzem erro menor do que ele, conforme a norma.

Aplicando-se a 2ª recorrência e depois de 200 gerações de processo evolutivo (~ 2 min), produziu nos coeficientes de menor grau pequenas diferenças, de tal forma que compensasse a falta de coeficientes de ordens maiores. Assim, o polinômio obtido é “melhor”, pelo menos no intervalo considerado, que o da série de Taylor. O mesmo comportamento foi verificado para outras funções, como $\tan(x)$, por exemplo.

Gráfico 6.38. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na ED.

Gráfico 6. 39. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação.

Gráfico 6. 40. Dispersão externa relativa dos coeficientes.

Gráfico 6. 41. Erros em relação à função e às derivadas.

Majorante do erro em D0 : $4.41e-014$
 Majorante do erro em D1 : $1.22e-012$

Este exemplo merece algumas comparações a mais. Uma comparação entre os erros nos intervalos quando se usa a série de Taylor, a de Legendre e o polinômio evoluído, cujos resultados podem ensejar trabalhos futuros (secção 6.3).

Gráfico 6. 42. Comparativo de erros em relação ao resíduo no *PVI* entre a série de Legendre e o polinômio evoluído.

Gráfico 6. 43. Comparativo de erros em relação ao resíduo no *PVI* entre o polinômio evoluído e a série de Taylor.

Exemplo 11.

Considere-se a equação diferencial:

$(1-x^2)y^{(2)} - 2xy^{(1)} + 20y - 80x^3 - 63x^2 + 12x + 1 = 0$ $x \in [-1, 1]$, sob as CI $\vec{y}_0 = [3, -33]$, cuja

solução exata, na base canônica, é $y(x) = \frac{175}{8}x^4 + 10x^3 - \frac{57}{4}x^2 - 4x + \frac{11}{8}$, que é o exemplo da

secção 3.5.2

Na base de Legendre, os coeficientes da expansão são $C = [1 \ 2 \ 3 \ 4 \ 5]$.

Esse *PVI* foi implementado no *SAM* considerando um polinômio de grau 5. Em 227 gerações, com pacotes de 20 gerações entre as secções de aprendizado, o processo evolutivo encontrou a solução exata. As recorrências divergiram, mesmo quando aplicadas a partir da solução exata.

Os resultados obtidos são descritos pelos gráficos seguintes.

Gráfico 6. 44. Resíduo ao longo do domínio ao substituir o polinômio de aproximação na *ED*.

Gráfico 6.45. Comparativo entre a solução analítica e suas derivadas com as do polinômio de aproximação.

Gráfico 6.46. Dispersão externa relativa dos coeficientes.

Gráfico 6. 47. Erros em relação à função e às derivadas.

Majorante do erro em D0 : 0.00e+000
 Majorante do erro em D1 : 0.00e+000
 Majorante do erro em D2 : 0.00e+000

Neste caso, como as potenciações são de graus relativamente baixos, a dispersão nula foi obtida sem ruídos. Em decorrência, as margens de erro são nulas.

6.2 Conclusão

Considerando os casos apresentados e vários outros analisados, esta pesquisa considera que as seguintes conclusões são de particular importância.

A descrição das derivadas e das integrais⁴⁶ da base Legendre em função da própria base permite que os *PVIs* sejam abordados no espaço de Hilbert.

O teorema da matriz operacional centraliza a questão dos operadores lineares na construção da matriz Z .

⁴⁶ Razzaghi & Yousefi [2001] construíram uma matriz de integração de Legendre no intervalo $[0, 1]$ para polinômios normalizados de outra forma, mas não a matriz de diferenciação.

Com os dois resultados descritos nos parágrafos anteriores se torna possível delimitar uma faixa de variação dos coeficientes de uma série, que se estreita cada vez mais em um processo evolutivo, na resolução dos *PVIs*. Esse processo evolutivo usa a série de Legendre como ponto de partida, mas o resultado final não é uma expansão que leve em conta os mínimos quadrados, ou seja, não é a série de Legendre pura, que permite erros mais expressivos ao fim do intervalo. Talvez por essa razão, o uso dos polinômios de Chebyshev, nos quais os pesos nos extremos do intervalo são maiores, resulte em L^∞ menores. Em versões futuras, isso será testado no processo adaptativo quando da integração da derivada de maior ordem, mas depois deve-se voltar para a base de Legendre de forma que todos os pontos do intervalo tenham o mesmo peso.

Os processos de recorrência (secção 5.5) propostos neste trabalho podem abreviar significativamente a busca de soluções em série dos *PVIs*;

Embora com razoável aumento de custo computacional, a evolução da expansão em série ortogonais pode aumentar em milhares de vezes o passo (h) dos processos numéricos de integração tradicionais, aumentando o grau do polinômio de aproximação, e, eventualmente, encontrar soluções exatas.

A transposição genérica de domínios nos *PVIs*, juntamente com as das *CI*, permite a translação do qualquer *PVI* para o espaço de Hilbert.

Durante o processamento, a faixa de limitação onde as mutações poderiam acontecer foi imperativa, uma vez que as variações eram contínuas. A integração dos coeficientes de Legendre pela matriz apresentada em 3.5.2 permitiu o refinamento da limitação e o incremento da rapidez do processamento.

O uso do *SAM* em aperfeiçoamentos de soluções numéricas pode substituir funções de controle por polinômios, supondo que as *CI* permaneçam as mesmas;

O critério de seleção tendo em vista o produto do erro máximo pela raiz quadrada da integral do erro quadrático impediu a seleção de funções bem aproximadas em boa parte do intervalo mas com erros expressivos no começo ou no fim, combinando a otimização global com a local.

A dispersão dos coeficientes de Legendre, que resulta na estimativa do majorante do erro feita pela matriz de diferenciação, pode ser usada para verificar a precisão de métodos numéricos de integração, a despeito do conhecimento do algoritmo do método.

6.3 Trabalhos futuros

A pesquisa nos periódicos relacionados ao tema deste trabalho permaneceu até a redação dessas páginas finais. A publicação de Doha & Bhrawy, em agosto deste ano, [Doha & Bhrawy, 2008], generalizando as matrizes operacionais de integração nos polinômios de Jacobi, acena com a possibilidade de escolha de várias classes de polinômios ortogonais nos métodos numéricos de integração, especialmente do Galerkin [Babolian & Fattahzadeh, 2007], [Elbarbary, 2002], [Shen, 1995], [Shen, 1994].

Para abordagens com polinômios de alto grau, o problema pode recair em uma matriz mal-condicionada, cuja diagonalização será afetada por erros inerentes a esse processo numérico. Talvez o processo evolutivo-adaptativo possa aprimorar esses coeficientes.

Considerando os polinômios ortogonais, talvez seja possível aumentar o passo do processo numérico de integração, sem perda de qualidade, como sugere o exemplo 10 da secção anterior. De fato, isso já foi feito com relativo sucesso por Higham [1996] com polinômios ortogonais para métodos de baixa ordem (2-3), e, aparentemente, merece pesquisas mais aprofundadas, uma vez que os principais métodos numéricos de integração, quais sejam, o de Runge-Kutta/Dormand & Prince [Campos, 2007, p. 333] e o de Adams-Bahsforth-Moulton [Campos, *ib.* p. 339], são baseados na série de Taylor e Lagrange, respectivamente, cuja adequação a um intervalo, como visto, não é tão boa quanto a de Legendre.

A secção 3.5.3 mostrou a impossibilidade da construção da matriz operacional de integração de Fourier⁴⁷, mas essa base parece ser um caminho para a abordagem de *PVIs* quase-periódicos em longos intervalos, como equação de Mathieu, por exemplo [Lara & Gadella, 2008].

As recorrências elaboradas neste trabalho na secção 5.5 funcionaram para a maioria dos casos, mesmo os não lineares, mas não em todos. O tema suscita uma investigação futura sobre quais devem ser as condições necessárias para que esse processo convirja.

Problemas do valor inicial em mais de uma dimensão já foram objeto de estudos com polinômios ortogonais [Shen, 1994], [Shen, 1995], [Doha & Bhrawy, 2008], considerando que as soluções sejam de variáveis separáveis.

⁴⁷ Escrita em função dos elementos da própria base.

Pode-se pensar, como um desafio, em um estudo futuro usando o processo evolutivo-adaptativo aqui desenvolvido, utilizando-se uma malha, como se faz em elementos finitos [Lee *et al.*, 2007], [Lim, Im & Cho, 2007], em uma tentativa de abordar problemas multidimensionais sem a imposição de que a solução seja de variáveis separáveis.

Finalmente, como meta de curto prazo, considera-se elaborar uma interface gráfica amigável, que já dê a partida com alguns valores *default*⁴⁸, e possa ser utilizada em outras áreas de pesquisa, a despeito do conhecimento das ferramentas internas do *SAM*, seja para aproximação de funções, resoluções de *PVIs*, análise de processos numéricos de integração ou transposições não-lineares de domínios, para *PVIs* de até 4ª ordem, com pequenas adaptações para abordar também os problemas com condições de **contorno**.

6.3.1 Matriz Sanduíche (Ω)

Como apontado na secção anterior, a pesquisa neste trabalho perdurou até a redação das páginas finais. Durante a escrita desta secção, obteve-se uma forma matricial para todas as operações lineares para qualquer base polinomial do espaço de Hilbert, pelo Teorema da Matriz Sanduíche. Sem dúvida, a aplicação desse teorema poderia abreviar vários cálculos do capítulo 3 — Fundamentação Analítica e, talvez, lá o teorema devesse estar, o que, praticamente, corresponderia a uma nova redação da tese, isto é, a exibição deste teorema nesta secção, embora obedeça à cronologia do trabalho, acarreta determinada vulnerabilidade em relação à sequência lógica. Por outro lado, omiti-lo não seria aceitável.

6.4 Teorema da Matriz Sanduíche

Sendo B_G uma base genérica de polinômios no intervalo $[a, b]$, a matriz Z_G do teorema da matriz operacional é obtida por $G\Omega G^{-1}$, onde $\Omega = L^{-1}Z_L L$, sendo $G_{n+1,n+1}$ e $L_{n+1,n+1}$ as matrizes que descrevem os polinômios genéricos e os de referência (L) na base canônica, respectivamente, no mesmo intervalo.

⁴⁸ Por exemplo, já escolhendo a transformação linear, identificando a ordem da *ED* e solicitando um vetor de *CI* com a dimensão adequada.

Prova

Como α é uma operação linear: $\alpha B_G = \alpha[G \cdot B_C] = G\alpha[B_C]$, onde B_C é a base canônica $B_C = [1, x, x^2, \dots, x^n]^T$.

Como a base canônica pode ser escrita em função da base de referência, por $B_C = L^{-1}B_L$, pode-se escrever:

$$\alpha B_G = \alpha[G \cdot B_C] = GL^{-1}\alpha[B_L].$$

Como $\alpha[B_L] = Z_L B_L$, obtém-se: $\alpha B_G = GL^{-1}Z_L B_L$.

Resta agora transformar B_L para a base genérica B_G , a fim de que o resultado seja escrito em função da base genérica de entrada.

Pelas matrizes definidas, têm-se: $B_L = LB_C$ e $B_C = G^{-1}B_G$, logo

$$\alpha B_G = G \underbrace{L^{-1}Z_L L}_{\Omega} G^{-1} B_G.$$

Como $\alpha B_G = Z_G B_G$, a matriz genérica Z_G é obtida por:

$$Z_G = G\Omega G^{-1}, \text{ onde } \Omega = L^{-1}Z_L L.$$

Corolários

1. A matriz de integração de uma base genérica dos polinômios de Jacobi é dada

$$\text{por: } M_{Gi} = \left(G^{-1}\right)^T \Omega_i^T G^T.$$

De fato, se a operação linear for uma integração, tem-se:

$$Z_{Gi} = GL^{-1}Z_{Li}LG^{-1} \text{ e, como } M_{Gi} = Z_{Gi}^T, \text{ conclui-se:}$$

$$M_{Gi} = \left(G^{-1}\right)^T L^T Z_{Li}^T \left(L^{-1}\right)^T G^T = \left(G^{-1}\right)^T \Omega_i^T G^T.$$

2. A matriz de diferenciação de uma base genérica dos polinômios de Jacobi é dada por:

$$M_{Gd} = \left(G^{-1}\right)^T \Omega_d^T G^T$$

Comentários

A demonstração usou a matriz de referência Z_L , considerando as matrizes de Legendre, pelo fato de elas serem facilmente obteníveis e terem todos seus elementos

expressos de forma simples por números racionais, mas qualquer outra base que tivesse a matriz Z já determinada poderia ter sido usada, inclusive a própria base canônica onde $B_C = IB_C$, com várias simplificações, já que a condição de ortogonalidade não foi exigida na demonstração.

É comum, nos processos espectrais de Galerkin, usarem-se as assim chamadas bases *shifted*-Lengendre ou *shifted*-Chebyshev, no intervalo real $[0,1]$ para soluções de *PVIs*. Nesses casos, a base de referência para construção das matrizes de integração e diferenciação (ou do sanduíche) pode ser a própria base canônica no intervalo $[0, 1]$, cuja expressão de integração ou diferenciação é simples e quase imediata.

Considerando-se a unicidade do resultado da operação linear de diferenciação nas funções contínuas, conclui-se que a matriz Ω_{Gd} é um invariante para todas as bases polinomiais.

6.4.1 Exemplos

Matriz de Integração e Diferenciação de Chebyshev I (T)

Considere-se a função $f(u) = -2 - 10x + 6x^2 + 16x^3$ $x \in [-1,1]$, que será integrada de -1 até u .

Escrita na forma de uma série de Chebyshev, ter-se-ia:

$$f(u) = [1, 2, 3, 4] \cdot [T_0, T_1, T_2, T_3]^T$$

Pelo teorema do sanduíche, conhecendo-se a matriz de integração de Legendre (Z_{Li}), pode-se obter a matriz de integração de Chebyshev, pelo corolário 1.

$$M_{Ti} = (T^{-1})^T \Omega_i^T T^T$$

Explicitamente, têm-se:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 \\ 0 & -3 & 0 & 4 & 0 \\ 1 & 0 & -8 & 0 & 8 \end{bmatrix}, \quad \Omega_i^T = \begin{bmatrix} 1 & -1/2 & 1/3 & -1/4 & 1/5 \\ 1 & 0 & 0 & 0 & -1/21 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 2/9 \\ 0 & 0 & 0 & 1/4 & 0 \end{bmatrix} \text{ e}$$

$$M_{Ti} = \begin{bmatrix} 1 & -1/4 & -1/3 & 1/8 & -1/5 \\ 1 & 0 & -1/2 & 0 & -1/21 \\ 0 & 1/4 & 0 & -1/4 & 0 \\ 0 & 0 & 1/6 & 0 & -2/9 \\ 0 & 0 & 0 & 1/8 & 0 \end{bmatrix}$$

Efetuada-se a operação $V = M_{Ti} \cdot [1, 2, 3, 4, 0]^T$, obtêm-se os coeficientes da série que

representa a integral da função, ou seja, $F(u) = \int_{-1}^u f(u) du = V \cdot B_T$, de forma que

$$V = [0, -1/2, -1/2, 1/2, 1/2].$$

Na forma canônica: $F(u) = 1 - 2u - 5u^2 + 2u^3 + 4u^4$.

Comentário: os elementos da última coluna da matriz de integração são, por assim dizer, arbitrários, pois vão multiplicar um elemento nulo. Portanto, se em vez da base de Legendre usar-se outra base para a construção da matriz sanduíche, esses elementos poderão ser diferentes, mas o resultado da integração permanece o mesmo.

Por exemplo, se na construção do sanduíche for usada a base canônica, os elementos da última coluna serão todos nulos, com exceção do primeiro que será unitário. Nesse sentido, a unicidade da matriz de integração só é garantida até a penúltima coluna.

Considerando a matriz de integração na base canônica, os termos laterais do sanduíche correspondem à matriz identidade, logo, Ω_i^T é uma das formas da matriz de integração na base canônica, que difere das outras apenas pelos elementos da última coluna.

Para obter a matriz de diferenciação de Chebyshev o procedimento é análogo, obtendo-se:

$$\Omega_d^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ e } M_{Td} = \begin{bmatrix} 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 4 & 0 & 8 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Como esperado, $M_{Td} \cdot V^T = [1, 2, 3, 4, 0]^T$.

Os termos não nulos do invariante $\Omega_{d(n+1, n+1)}^T$ são: $\Omega_{d(i, i+1)}^T = i$, com $i = 1, n$, sendo essa matriz a própria matriz canônica de diferenciação, já que, para essa base, os termos laterais do sanduíche são a matriz identidade.

Matriz de diferenciação e integração de Laguerre

Considerando polinômios até a ordem 4, a matriz que descreve os coeficientes dos polinômios de Laguerre (L_a) base canônica, na ordem crescente dos expoentes, é:

$$L_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 1/2 & 0 & 0 \\ 1 & -3 & 3/2 & -1/6 & 0 \\ 1 & -4 & 3 & -2/3 & 1/24 \end{bmatrix}$$

Considerando o intervalo para integração de 0 até x , a matriz sanduíche de integração para a base canônica é:

$$\Omega_i^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 0 \end{bmatrix}$$

Como, pelo corolário 1, $M_{L_a i} = (L_a^{-1})^T \Omega_i^T L_a^T$, obtém-se:

$$M_{L_a}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & -4/5 \\ -1 & 1 & 0 & 0 & 21/5 \\ 0 & -1 & 1 & 0 & -44/5 \\ 0 & 0 & -1 & 1 & -46/5 \\ 0 & 0 & 0 & -1 & -19/5 \end{bmatrix}$$

Para a diferenciação, considerando ainda a base canônica de 0 até x , a matriz é a mesma de -1 até x , já que a diferença é uma constante:

$$\Omega_d^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Pelo corolário 2, $M_{L_a d} = (L_a^{-1})^T \Omega_d^T L_a^T$, obtém-se: $M_{L_a d} =$

$$\begin{bmatrix} 0 & -1 & -1 & -1 & -1 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Considerando, como exemplo, os coeficientes de uma série de Laguerre tais que:

$C_{La} = [1, 7, 5, 2, 0]$, que leva explicitamente a: $f(x) = 15 - 23x + \frac{11}{2}x^2 - \frac{1}{3}x^3$, $x \in [0, \infty)$, obtêm-

se os coeficientes da série da integral da função por:

$$V^T = M_{La} \cdot C_{La}^T, \text{ ou seja, } V = [1, 6, -2, -3, -2].$$

Naturalmente, $M_{La} \cdot V^T = C_{La}^T$, como pode ser verificado.

Matriz de diferenciação e integração de Hermite

Aqui se considera a integração também de 0 até x e podem-se usar as mesmas matrizes sanduíche da subsecção anterior.

A matriz que descreve os coeficientes dos polinômios de Hermite em função da base canônica é:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ -2 & 0 & 4 & 0 & 0 \\ 0 & -12 & 0 & 8 & 0 \\ 12 & 0 & -48 & 0 & 16 \end{bmatrix}, \text{ tem-se:}$$

$$f(x) = -9 - 10x + 20x^2 + 16x^3$$

Pelo corolário 1, $M_{Hi} = (H^{-1})^T \Omega_i^T H^T$

$$V^T = M_{Hi} \cdot C_H^T \Rightarrow V_H = [1/2, 1/2, 7/4, 5/6, 1/4], \text{ ou seja,}$$

$$F(x) = -9x - 5x^2 + \frac{20}{3}x^3 + 4x^4.$$

A obtenção da matriz de diferenciação é análoga e, como nos outros exemplos, tem-se, naturalmente:

$$M_{Hd} \cdot V^T = [1, 7, 5, 2, 0]^T.$$

Apêndice – Descrição do Algoritmo e Comentários

Este apêndice visa descrever a construção, funcionalidades, pseudocódigos (quando um detalhamento se fizer necessário) e outras características do algoritmo utilizado neste trabalho para a resolução de equações diferenciais. A plataforma foi o software Matlab 7, com o *tollbox* de matemática simbólica. Um exemplo simples vai acompanhando, paulatinamente, nas várias etapas da resolução, cada uma das secções.

Aberto o Matlab, coloca-se o *path* da pasta em que estão os arquivos de funções e de comando, por exemplo, *path = c:\ED_AG*, se foi para esse diretório que você copiou os arquivos. Depois disso, crie um diretório para seu uso. Lá ficarão os arquivos com as bases de dados e os resultados do processamento.

O acesso ao programa se faz digitando-se:

ED

Esse arquivo é o controle primeiro de fluxo (ED.m). A tela vai apresentar 8 opções:

```
=====
1. Criar uma BASE de dados ... Equação, intervalo etc..
2. Criar um novo polinômio a partir de uma BASE existente.
3. Retomar um processamento.
4. Fazer os gráficos de algum processamento anterior.
5. Analisar a qualidade de um método numérico de integração.
6. Solução por recorrência.
7. Aproximar uma função (critério LG).
8. Problemas do Contorno (BVP)

Escolha a tarefa.
```

As secções seguintes descrevem o que é processado em cada uma das opções.

1.1 Base de dados

Ao escolher a opção 1, o programa vai chamar o arquivo de comandos DATA.m. Na tela, aparecerá a mensagem:

```
=====
Digite um nome para a base de dados.
=====
```

Escolha um nome apropriado, do qual você possa se lembrar facilmente depois. Neste exemplo, o nome escolhido foi 'pêndulo'. O programa vai salvar esse arquivo

colocando um B- na frente do nome que você escolheu para ficar claro que é um arquivo de base de dados, e não de processamento. A tela apresentará:

```
=====
Este programa é baseado em processos numéricos de integração e
algoritmo genético.
As funções para aproximação são os polinômios de Legendre.
A saída será um polinômio de aproximação até o grau escolhido.
Insira o domínio ( [a, b] ) da função.
=====
```

Insira o domínio da função procurada, exatamente da forma que a mensagem estipula, com os colchetes. Nesse exemplo, o domínio escolhido e inserido foi: `[0, 9]` .

O passo seguinte é a entrada da equação, conforme explica a tela.

```
=====
Entre com a equação na variável x.
D0 é a função D1 a primeira derivada
e assim por diante.
O programa já impõe que a eq. digitada seja = 0
=====
```

A entrada desse exemplo foi a equação do pêndulo simples $\frac{d^2y}{dt^2} + \sin(y) = 0$

```
D2 + sin(D0) = 0
```

A colocação do “ = 0 ” não é necessária, o programa já faz isso. Entretanto, se for colocada, não altera em nada o processamento. Se a variável da equação não for x , a troca deve ser efetuada, caso contrário o processo simbólico não reconhece a variável.

Agora, o programa apresenta a equação que você digitou e identifica a ordem da equação. A função que identifica a ordem da equação é `find_ordem.m`. Ela utiliza recursos da matemática simbólica do Matlab.

A apresentação da equação e identificação da ordem é importante para que o usuário não entre em um longo processamento com algum dígito ou sinal errado na equação, desperdiçando tempo.

```
=====
D2+sin(D0) = 0

A equação é de 2ª ordem.
Se souber, escreva a solução analítica, ou apenas tecle enter.
=====
```

Sem dúvida, a opção de solução analítica parece estranha, pois, se o usuário a conhecesse, por que usaria o programa? Entretanto, essa opção existe para que se possa verificar a confiabilidade dos resultados com soluções conhecidas e tê-las no arquivo de base de dados para eventuais futuras consultas. Nesse exemplo, apenas foi teclado enter.

A próxima tela pergunta sobre a abordagem que será feita: se um método numérico será usado como referência (ponto de partida) ou não. A escolha deve ser feita neste ponto, pois os métodos numéricos precisam das condições iniciais, enquanto que o AG pode procurar uma solução sem que lhe sejam dadas as condições iniciais. Além disso, há equações em que os métodos numéricos emperram e, portanto, manter o método numérico em todos os processamentos não se revelou possível.

```
=====
Método numérico ou aleatório? (N/A)
=====
```

Nesse exemplo, a escolha foi N (método numérico), para iniciar.

1.1.1 Método numérico

Com a escolha acima, o programa chama então o arquivo de comandos `DATAnum.m`.

Essa opção obriga a entrada de condições iniciais, pois sem elas os métodos numéricos não trabalham. Tela:

```
=====
Cond.Inic: [y(xmin) y'(xmin) y''(xmin) y''' . . .]
[3.1 0]
=====
```

Entre com o vetor de condições iniciais. O número de elementos desse vetor deve ser igual à ordem da equação diferencial. No exemplo, a entrada se refere ao pêndulo perto da posição crítica, o ângulo é próximo de π rad quando a velocidade é nula.

Em seguida, vem a escolha para transposição do domínio. Se o domínio for finito, a mais simples é a linear. A trigonométrica e a irracional são para domínios infinitos. No exemplo, a escolha foi linear.

```
=====
1 . Linear:          tx2u = mx + n
2 . Trigonométrica: tx2u = atan(x)
3 . Irracional:      tx2u = x/(1+x^2)^(1/2)

Qual transformação vc escolhe?
1
=====
```

O programa apresenta então qual equação ele vai resolver no intervalo de trabalho $([c, d])$, a equação transformada, e as condições iniciais também transformadas.

```
=====
```

Equação no intervalo de trabalho [c; d].

Equação transformada:

$$.494e-1 \cdot D^2 + \sin(D0) = 0$$

Vetor de cond inics p/ o Método Numérico: $y(0)$; $y'(0)$...

3.1000 0

=====

O próximo passo é a integração numérica no próprio intervalo em que a equação está definida com métodos numéricos consagrados. Deve ser feita uma escolha, conforme a precisão e o tipo de ED. A maioria dos casos pode ser resolvida com a opção 1 – Método de Adams, com alta precisão. Se o processamento estiver muito demorado ou acusar algum erro, por exemplo, divisão por zero, rode o programa novamente e, ao chegar a este ponto, faça outra opção.

=====

1. ode113 - RelTol: 2e-014 -AbsTol:1e-21

2. ode113 - RelTol: 1e-2 -AbsTol:1e-3

3. ode45 - RelTol: 1e-10 -AbsTol:1e-12

4. ode23 - RelTol: 1e-2 -AbsTol:1e-3

5. ode15s - RelTol: 1e-2 -AbsTol:1e-3

6. ode23t - RelTol: 1e-2 -AbsTol:1e-3

7. ode23s - RelTol: 1e-10 -AbsTol:1e-12

8. Pular esta integração para fazer só a no domínio de trabalho.

Escolha o método.

1

Ode113 usado com sucesso!

=====

As próximas telas serão de apresentação de resultados, considerando que a opção escolhida tenha sido a 1.

A janela de gráficos deve se abrir e apresentar o seguinte gráfico.

Gráfico I. 1. Gráfico da função e derivadas no intervalo próprio.

O próximo gráfico representa a resolução numérica da ED transformada para o intervalo $[-1, 1]$, onde os polinômios de Legendre formam uma base completa.

Gráfico I. 2. Gráfico da função e derivadas no intervalo de trabalho.

Esses gráficos visam permitir ao usuário o acompanhamento do processamento dos dados para verificar se os resultados são consistentes. Em uma versão mais econômica, tudo isso pode ser omitido, deixando os gráficos somente para a parte final.

Após a apresentação dos gráficos, o programa comunica que processou todas as entradas e que a base de dados está definida.

```
=====
F I M   D A S   E N T R A D A S !
A BASE de dados está definida.
Vc. quer tentar uma série (s/n)?
n
=====
```

A opção foi não. A série será processada depois.

1.1.2 Método aleatório

Considerando o menu:

```
=====
```

Método numérico ou aleatório? (N/A)

=====

Se o usuário escolher A (aleatório), nenhum método numérico será processado para dar a “partida” no programa, e os gráficos discutidos na seção anterior não serão apresentados. O programa vai partir, então, de um polinômio nulo.

Definidas as *CI*, os menus se sucedem da mesma forma que o item anterior até a base de dados estar montada.

1.2 Criar um polinômio

Abaixo, tem-se a tela de entrada do programa, com a escolha 2.

=====

1. Criar uma BASE de dados ... Equação, intervalo etc..
2. Criar um novo polinômio a partir de uma BASE existente.
3. Retomar um processamento.
4. Fazer os gráficos de algum processamento anterior.
5. Analisar a qualidade de um método numérico de integração.
6. Solução por recorrência.
7. Aproximar uma função (critério LG).

Escolha a tarefa.

2

=====

A próxima tela depois da escolha será:

=====

Quantos coefs. p/ o poly de trabalho(1 a 40)?

15

=====

Aqui, a escolha foi 15, ou seja, um polinômio de 14º grau.

A próxima tela pede ao usuário que nomeie o arquivo de trabalho. Isso é importante para que ele possa salvar os resultados do processamento e, eventualmente, retomar a evolução mais tarde. O nome escolhido foi T1.

=====

Hello!

Atenção: o arquivo a ser salvo

não é o arquivo da base de dados! É o de TRB.

Nome p/ o arq. de trabalho?

T1

=====

Em seguida, a pergunta é: *quantas gerações você quer processar?*. No exemplo, foram pedidas 2001 gerações.

=====

Quantas gerações vc quer?

2001

É preciso, então, esperar um pouco. O programa processa lotes de 200 gerações antes de exibir a margem de erro. A cada lote, é exibido o produto dos erros LG e a diferença em relação ao lote anterior. Essa diferença aparece porque, às vezes, o número de casas apresentada em LG pode não ser suficiente para mostrar a evolução.

A secção 2.2.1 detalha que o produto LG é a raiz do erro quadrático médio multiplicado pelo erro máximo, para o melhor da população.

A sequência de resultados é apresentada abaixo. Ao final, o programa pergunta se usuário deseja os gráficos comparativos. A opção escolhida foi *sim*. Eventualmente, se a margem de erro ainda estiver muito grande, pode-se encerrar o processamento sem a apresentação dos gráficos.

```
=====
Ger. nº 200 /2001 - 26 coefs. - LG: 2.2649e-003 dif.4.41e-005
Ger. nº 400 /2001 - 26 coefs. - LG: 2.2149e-003 dif.5.00e-005
Ger. nº 600 /2001 - 26 coefs. - LG: 2.1975e-003 dif.1.74e-005
Ger. nº 800 /2001 - 26 coefs. - LG: 2.1386e-003 dif.5.89e-005
Ger. nº 1000 /2001 - 26 coefs. - LG: 2.1293e-003 dif.9.27e-006
Ger. nº 1200 /2001 - 26 coefs. - LG: 2.1277e-003 dif.1.54e-006
Ger. nº 1400 /2001 - 26 coefs. - LG: 2.1277e-003 dif.1.51e-015
Ger. nº 1600 /2001 - 26 coefs. - LG: 2.1243e-003 dif.3.42e-006
Ger. nº 1800 /2001 - 26 coefs. - LG: 2.1199e-003 dif.4.43e-006
Ger. nº 2000 /2001 - 26 coefs. - LG: 2.1188e-003 dif.1.05e-006
Ger. nº 2001 /2001 - 26 coefs. - LG: 2.1188e-003 dif.0.00e+000
Vc quer os graficos comparativos e a função? (s/n)?
s
=====
```

O primeiro gráfico apresenta a dispersão relativa (σ_{rel}) dos coeficientes, como descrito na secção 5.3.1.

Gráfico I. 3. Dispersão relativa dos coeficientes nesse estágio da evolução.

O segundo gráfico apresenta o erro (L^∞) ao longo do domínio. Melhor dizendo, o resultado que se obtém ao substituir o polinômio no PVI , ou seja, o resíduo. Se o polinômio estiver bem evoluído, o erro estará bem distribuído ao longo do domínio, isto é, sem picos de erro nos extremos, como exemplifica o gráfico seguinte.

Gráfico I. 4. Erro ao longo do domínio e a raiz do erro quadrático médio (RMS).

O próximo gráfico apresenta a concordância de pontos entre o polinômio evoluído e os pontos obtidos pelo método numérico escolhido.

Gráfico I. 5. O polinômio e sua derivada se superpõem aos pontos obtidos pelo método numérico de precisão.

O último gráfico compara o erro ao longo do domínio do polinômio evoluído em relação ao melhor polinômio que se obteria diretamente dos pontos do método numérico, assim como o *RMS* de cada um deles.

Gráfico I. 6. Além de o polinômio evoluído ter *RMS* menor, ele é bem comportado nos extremos do domínio.

O *prompt* final apresenta a função, na forma polinomial, no domínio da ED com três decimais para os coeficientes, o erro quadrático médio, tendo como critério a ED, e o tempo de processamento. A imposição de três casas decimais é só para apresentação na tela, pois os coeficientes são racionais, ou seja, no Matlab eles são tratados e salvos como elementos simbólicos, de precisão ilimitada.

=====

A solução é: $f(x) =$

$$\begin{aligned}
 & -0.562 \cdot 10^{-11} x + 0.108 \cdot 10^{-9} x^{23} + 0.841 \cdot 10^{-7} x^{21} + 0.176 \cdot 10^{-13} x^{25} \\
 & - 0.361 \cdot 10^{-8} x^{22} - 0.0230 x^2 - 10.8 x^{10} - 16.0 x^8 - 6.93 x^6 \\
 & - 0.572 x^4 - 0.202 \cdot 10^{-11} x^{24} - 0.263 x^{14} - 2.64 x^{12} - 0.000200 x^{18} \\
 & - 0.0111 x^{16} + 12.5 x^7 + 2.56 x^5 + 0.0643 x^3 + 0.929 x^{13} + 6.01 x^{11} \\
 & + 15.0 x^9 + 0.0000192 x^{19} + 0.00166 x^{17} + 0.0600 x^{15} - 0.145 \cdot 10^{-5} x^{20} \\
 & + 3.10
 \end{aligned}$$

Raiz do erro médio quadrático médio na ED: 5.90e-004

Erros máximos no domínio:

Majorante do erro em D0 : 1.73e-003

Majorante do erro em D1 : 1.11e-002

Majorante do erro em D2 : 1.68e-001

Se a função for periódica, digite 'Período': a função e as derivadas serão ajustadas para qualquer intervalo.

Figura I. 1. Tela que apresenta a série que aproxima a função com três algarismos significativos.

Se o usuário quiser os coeficientes na forma racional, ao final do processamento a variável F0 conterá o polinômio simbólico. Assim, é só digitar F0 para observar os coeficientes simbólicos, como a razão entre dois inteiros.

1.3 Retomada de um processamento

Eventualmente, o usuário pode querer tentar evoluir ainda mais um polinômio que já tenha passado por várias gerações. Essa opção atende a essa necessidade.

A tela vai perguntar o nome do arquivo TRB-*.mat cujo processamento será retomado. A melhor forma de inserir é copiar o nome do arquivo examinando o diretório em que ele se encontra e colar no *prompt*, evitando assim erros de digitação.

```

=====
Digite o nome do arquivo TRB-*.mat.
TRB-pend-26-L-N.mat
=====

```

A próxima pergunta se refere ao número de coeficientes. O usuário pode querer aumentar a precisão, aumentando um coeficiente, ou, às vezes, reduzir um coeficiente com a intenção de obter um polinômio menor. No exemplo, a escolha foi *manter*.

```
=====
enter: Manter; 1: aumentar ; 2: reduzir
E os coeficientes?
```

Após a retomada, os menus e gráficos são idênticos aos da secção anterior.

Essa tarefa pode ser executada até que se verifique a estagnação evolutiva, quando o *SAM* não consegue mais evoluir, em virtude da limitação do número de coeficientes.

1.4 Gráficos de processamentos

Com essa opção é possível, para qualquer processamento efetuado, rever os gráficos obtidos.

Outra forma seria o usuário pedir a retomada de um processamento, solicitando apenas uma geração, mas isso afetaria a matriz de evolução.

Além disso, nesta opção é possível avaliar como foi a evolução com as sucessivas gerações, observando-se a margem de erro ao longo do domínio.

O comando *grafAG* executa um *script* de montagem para um gráfico 3D que vai mostrando o erro quadrático ao longo do intervalo para as sucessivas gerações, conforme ilustra o gráfico seguinte.

Gráfico I. 7. Erro absoluto ao longo do intervalo para as sucessivas gerações.

1.5 Qualidade de um processo de integração numérica

Esta opção se presta a uma análise, usando os coeficientes de Legendre, considerando que o usuário tenha um conjunto de pontos que são a solução numérica de determinada equação.

Como apresenta a secção 5.1, a dispersão nos coeficientes de Legendre é um índice de quanto os resultados são confiáveis. Há duas dispersões a serem consideradas, uma que verifica a dispersão dos coeficientes entre os pontos obtidos pelo método (consistência interna) e outra que verifica a dispersão em relação à última derivada (consistência com a equação).

A secção 5.3.1 da tese apresenta um exemplo completo desse recurso do *SAM*.

1.6 Recorrência

Conforme descreve a secção 5.1, o processo computacional de busca dos coeficientes da série de Legendre para a função pode ter como entrada até um polinômio aleatório. A partir do primeiro, o processo computa os coeficientes da série analisando quanto valeria a última derivada ao longo do domínio, ou seja, obtém um novo vetor. Esse novo vetor pode ser usado como nova entrada no processo iterativo até que o vetor de saída se iguale ao de entrada, quando, então, ter-se-á obtido a solução exata. Nos casos analisados neste trabalho, nem sempre esse processo converge; outra alternativa é pôr-se como nova entrada a média entre o vetor de saída e o de entrada, cogitando-se um processo de convergência em que a diferença de módulos entre os vetores iria diminuindo, como descreve a secção 5.4. Para os casos analisados, esse processo se revelou mais abrangente, embora um pouco mais lento, convergindo na maioria deles, mas não em todos.

A intenção é aprimorar o polinômio de forma a produzir uma solução melhor que a apresentada pelo conjunto de pontos, e com erro máximo menor.

A opção “Reduzir dispersão” fez parte da pesquisa, mas não produz séries melhores. O que se verificou é que essa opção vai convergindo para dispersões menores dentro do número de coeficientes estipulado para a série, mas vai empurrando dispersões grandes para os coeficientes de ordens maiores que a série e aumentando o erro *LG*.

Como exemplo, vai-se considerar a base de dados da secção I.1 (B-L-Pend-N.mat), mas poderia ser também um arquivo de trabalho. Nesse último caso o número de

coeficientes já estaria definido. Como a opção foi por uma base de dados, o *SAM* vai perguntar com quantos coeficientes o usuário quer trabalhar. Nesse exemplo, escolheu-se 27 e a opção 2, reentrada com a média entre o vetor de entrada e o de saída, como descreve a secção 5.5. da tese.

Em seguida, o *SAM* pergunta quantas iterações o usuário deseja. Escolheu-se 10 (~ 30 s de processamento).

O erro *LG* constatado após a primeira iteração foi de $1,07e-2$. O menor erro *LG* da série veio após na iteração 6: $2,47e-3$. Se o usuário escolher “consertar” a origem dos dados, os coeficientes da série com menor *LG*, dentre as iterações feitas, serão atribuídos ao arquivo de entrada.

O processo de recorrência é particularmente importante nos casos em que os *MN* não conseguem dar a partida, seja pelo fato da matriz de massa ser singular, ou pelo fato de algum valor imaginário emergir nos cálculos iniciais do método, como ocorreu no exemplo 3 do capítulo 6. Usar a recorrência nesses casos abrevia muito o processo evolutivo.

1.7 Aproximação de uma função

Este é um subproduto das funcionalidades do *SAM*. Ao usuário fornecer a função $q(x)$ e o intervalo em que deseja a aproximação, o *SAM* vai procurar por uma aproximação polinomial que seja boa para a função e também para a derivada, com o critério *LG*, montando o *PVI*:

$$\frac{d}{dx}(q - f^*) = 0, \text{ onde } f^* \text{ é a série que aproxima a função.}$$

$$\text{A imposição acima equivale a: } y^{(1)} - \frac{dq(x)}{dx} = 0, \text{ com } y_0 = q(a).$$

Se no intervalo fornecido a função e a derivada não forem limitadas, a série não é uma boa aproximação.

Como exemplo, vai-se obter uma aproximação polinomial para $q(x) = \sqrt{x}$ no intervalo $[1, 2]$, com 10 coeficientes que evoluiu por 200 gerações.

Majorante do erro em D0 : 2.74e-009
Majorante do erro em D1 : 1.15e-007

Os erros ao longo do domínio são apresentados nos gráficos seguintes.

Gráfico I. 8. Erros no intervalo em que a função foi aproximada.

1.8 Comentários

Esses comentários finais têm a intenção de esclarecer algumas particularidades do *SAM* e apresentar alguns resultados dos processamentos cuja colocação no corpo da tese seria de, certa forma, desmedida.

1.8.1 Artíficos do código

Na secção 5.4.2, esta tese apontou o uso do comando *eval* para reduzir o tempo de processamento centenas de vezes, evitando o trabalho com cálculos simbólicos a cada geração. Há dois usos particulares desse artifício, uma *string* que determina a seqüência de procedimentos de ajuste dos coeficientes para que sejam mantidas as condições iniciais, armazenada na variável “ajuste”, e outra *string* que determina a seqüência de procedimentos no torneio que fará o *ranking* dos polinômios candidatos, armazenada na variável “rvc”. Os conteúdos dessas variáveis são apresentados a seguir considerando o exemplo número 3, pois nele o número de coeficientes é pequeno, mas é possível ter-se uma idéia de quão extensa seria a expressão ao se trabalhar com 26 coeficientes, por exemplo.

ajuste =

```

C(1:end,1) =2*C(1:end,3)-5*C(1:end,4)+1;C(1:end,2) =3*C(1:end,3)-
6*C(1:end,4);
rvc =
8./27.*(3.*C(i_indiv,3)+15.*C(i_indiv,4).*tspan).^3+4.*C(i_indiv,2)+12.*C(i
_indiv,3).*tspan+4.*C(i_indiv,4).*(15./2.*tspan.^2-3./2)-C(i_indiv,1)-
C(i_indiv,2).*tspan-C(i_indiv,3).*(3./2.*tspan.^2-1./2)-
C(i_indiv,4).*(5./2.*tspan.^3-3./2.*tspan)-1./2-3.*tspan-3./2.*tspan.^2

```

1.8.2 Resultados simbólicos

As expressões a seguir constituem as séries obtidas com o processo evolutivo, expressas com coeficientes dados pela razão entre dois inteiros em cada um dos exemplos da secção 6.1.

Exemplo 1.

```

f(x) =
309485009821345012844019175/309485009821345068724781056+13115837053/7737125
2455336267181195264*x-305897301000461925/154742504910672534362390528*x^2+
77371252676752188072692075/77371252455336267181195264*x^3-
171792659027495275/154742504910672534362390528*x^4

```

Exemplo 2.

```

f(x) =
-10401915355315197692271389683/618970019642690137449562112*x-
6729742933600774843374840375/5575186299632655785383929568162090376495104*x^
19+9903520316050265432010753975/9903520314283042199192993792*x^2-
97498571811028740585/39614081257132168796771975168*x^3-
105637542388896890087430959405/633825300114114700748351602688*x^4-
5139928281590642407581/158456325028528675187087900672*x^5+42255304748161784
208445016193/5070602400912917605986812821504*x^6-
1373096974367108405669397/20282409603651670423947251286016*x^7-
514953758905122796966077298785/2596148429267413814265248164610048*x^8-
206929074818023460962407205/5192296858534827628530496329220096*x^9+23064664
4122923814966491659205/83076749736557242056487941267521536*x^10-
2774537455974620607773538855/332306998946228968225951765070086144*x^11-
29724263631459304834891474237/1329227995784915872903807060280344576*x^12-
919884961305742791470248725/1329227995784915872903807060280344576*x^13+6442
542722749565757750793425/21267647932558653966460912964485513216*x^14-
1984214737001253118526090145/85070591730234615865843651857942052864*x^15+98
096571622539121903793082405/43556142965880123323311949751266331066368*x^16-
26356764898114384767872724765/87112285931760246646623899502532662132736*x^1
7+35660221805411513546835701025/1393796574908163946345982392040522594123776
*x^18+2703506264391549017821414995/8920298079412249256614287309059344602392
1664*x^20-
14314141842945734609457495/44601490397061246283071436545296723011960832*x^2
1-140156026065/309485009821345068724781056

```

Exemplo 3.

```
f(x) = x^3+1
```

Exemplo 4.

```

f(x) =
-1/649037107316853453566312041152512*(18374710096798829252*x*pi^20+
10537019111747972505907232797780390*pi^17*x^4-

```

$$\begin{aligned}
&14919768517025608671133781567100\pi^3x^{18}+4436849912172685772643566688\pi^{14}x^7+5333223535058583621747360\pi^{16}x^5- \\
&5199810751040762016886583602634736\pi^{15}x^6- \\
&44706208489793571011544856731510\pi^5x^{16}+30344699011378741668541009639800 \\
&\pi^4x^{17}-40544054649230449\pi^{21}- \\
&6405739490845634853909730222847250\pi^{19}x^2+19769053451969620057980\pi^{18}x^3+310234595920138496471047755000\pi^{12}x^9+122186052585981526442612813460 \\
&5310\pi^{13}x^8+49421999531682954357340986887040\pi^6x^{15}- \\
&168967718527547078046077111054828\pi^{11}x^{10}+512294309984512628701680513780 \\
&0\pi^{10}x^{11}+1589938472299992413739161270704\pi^9x^{12}+26956918899830665144 \\
&663341548800\pi^8x^{13}+98538601360159953988970008320x^{21}- \\
&1035024412292792454385967374260\pi x^{20}- \\
&42571723370574474324058326436800\pi^7x^{14}+5021333182268931516965366497800\pi^2x^{19})/\pi^{21}
\end{aligned}$$

Exemplo 5.

$$\begin{aligned}
f(x) = &1/41538374868278621028243970633760768*(2073315532879845470720\pi^2\operatorname{atan}(x)^4- \\
&516991409841833329\pi^6+3507311423558942568\operatorname{atan}(x)\pi^5+124029799075826855 \\
&60832\operatorname{atan}(x)^6+409967327414062155390611148731338560\pi^4\operatorname{atan}(x)^2- \\
&35840\pi^3\operatorname{atan}(x)^3-10648250230785822240768\pi\operatorname{atan}(x)^5)/\pi^6
\end{aligned}$$

Exemplo 6.

$$f(x) = \frac{x}{(x^2+1)^{(1/2)}}$$

Exemplo 7.

$$\begin{aligned}
f(x) = &-565364007/1511157274518286468382720x+ \\
&226630693706398467897831633/18446744073709551616000000000000x^{14}- \\
&975657116641420459439355981/5902958103587056517120000000000x^{11}+5347312813 \\
&89308735106136333/2361183241434822606848000000000x^{10}+38332688314667334190 \\
&884567/1888946593147858085478400000000x^8+152452613591116377845229/47223664 \\
&82869645213696000000x^6+207574063027033441207/302231454903657293676544000x^4-1088671948155970389502863/11805916207174113034240000000x^7- \\
&3488533244891638110925569/1888946593147858085478400000x^5- \\
&467000456388507659/30223145490365729367654400x^3- \\
&27470958947028154581222199/11805916207174113034240000000x^9- \\
&141227834041176021265273119/3689348814741910323200000000000x^{13}+279071859/ \\
&4835703278458516698824704+1257029722394615496575704533/18446744073709551616 \\
&0000000000000000x^{18}- \\
&20136614544777927033369953/461168601842738790400000000000000000000x^{19}+39095 \\
&6714728548369562944783/2305843009213693952000000000000000000000x^{20}- \\
&8694672769877257556849313/288230376151711744000000000000000000000x^{21}- \\
&687040640713870164387194031/2305843009213693952000000000000000000000x^{15}- \\
&414077822717534410754857923/576460752303423488000000000000000000000x^{17}+3983380 \\
&4223606503137698538173/7378697629483820646400000000000000000000x^{16}+56807424263 \\
&6917288530339/2417851639229258349412352x^2+3355503391704550628053300061/36 \\
&8934881474191032320000000000000000x^{12}
\end{aligned}$$

Exemplo 8.

$$\begin{aligned}
f(x) = &-19928453097501786578368935605/3713820117856140824697372672x- \\
&7654198987082319733701918725/1237940039285380274899124224x^2- \\
&3264769207466845916513860625/464227514732017603087171584x^3- \\
&214409033957591382496968434375/33424381060705267422276354048x^4-
\end{aligned}$$

```

116254594840816373995406871875/16712190530352633711138177024*x^5-
252772480618943856600602890625/37602428693293425850060898304*x^6-
223701441132498925767745234375/28201821519970069387545673728*x^7+1049285255
9856704239444708984375/902458288639042220401461559296*x^8-
2495381643939985906088575947265625/12183186896627069975419731050496*x^9+631
4693015539127799649409462890625/4061062298875689991806577016832*x^10-
21240569969775744159469131787109375/2284347543117575620391199571968*x^11+34
85865156205142054965131986083984375/82236511552232722334083184590848*x^12-
18471485493573334381619950897216796875/123354767328349083501124776886272*x^
13+2814690681898933734975040130615234375/6853042629352726861173598715904*x^
14-
4485305219409364033455381011962890625/5139781972014545145880199036928*x^15+
2807630625378674667371510951995849609375/1973676277253585336017996430180352
*x^16-
5170528496382970982279496662139892578125/2960514415880378004026994645270528
*x^17+41660090191071294567938242816925048828125/266446297429234020362429518
07434752*x^18-
28906908312584047340427289485931396484375/299752084607888272907733207833640
96*x^19+87985704218280012174830188274383544921875/2398016676863106183261865
66266912768*x^20-
23412339644890638055971419811248779296875/359702501529465927489279849400369
152*x^21-4951760157141526692292779025/2475880078570760549798248448

```

Exemplo 9.

```

f(x) =
-1120637074713220969891663977565781/730166745731460135262101046296576*x-
4521835291032956781344853393842739/10141204801825835211973625643008*(2/9*x-
1)^7+520804486734160024021166004918037/316912650057057350374175801344*(2/9*
x-1)^9-
29244438962459877335134675098037/39614081257132168796771975168*(2/9*x-
1)^11-
1280450009290606478681379707409207/39614081257132168796771975168*(2/9*x-
1)^13+4742426055396479082453470011092617/19807040628566084398385987584*(2/9
*x-
1)^15+674956749252140021782796339414725/162259276829213363391578010288128*(
2/9*x-1)^3-
131505144418813602032711735099885/309485009821345068724781056*(2/9*x-
1)^37+1232573724206249669407073605097085/309485009821345068724781056*(2/9*x
-1)^30-
1955958774203459633824315341011337/309485009821345068724781056*(2/9*x-
1)^28-
1111886484003393774665743084340839/154742504910672534362390528*(2/9*x-
1)^24+2363551365458760459417363989269065/309485009821345068724781056*(2/9*x
-1)^26+412975873429088715201760389733919/77371252455336267181195264*(2/9*x-
1)^22+24740113856851206335857832206215/618970019642690137449562112*(2/9*x-
1)^39-
1953243998026402431823675853004047/618970019642690137449562112*(2/9*x-
1)^20-346610208074118320287199523254039/618970019642690137449562112*(2/9*x-
1)^16+921893639101360047595340530162711/618970019642690137449562112*(2/9*x-
1)^18+411104495799797847483573446649193/2475880078570760549798248448*(2/9*x
-
1)^14+13819045701823131140056329831495455/324518553658426726783156020576256
*(2/9*x-1)^4-
6622726808156633542139229550669691/81129638414606681695789005144064*(2/9*x-
1)^6+968514096716052416628332070277215/158456325028528675187087900672*(2/9*
x-1)^10-
152664772761330800813254327730531/316912650057057350374175801344*(2/9*x-
1)^8-
748816105859949313919410169715761/19807040628566084398385987584*(2/9*x-
1)^12-

```


Referências Bibliográficas

- ARFKEN, G. B.; WEBER, H.J. (1995). *Mathematical methods for physicists*. 4 ed. San Diego: Academic Press, xxii, 1030 p.
- ARMENTANO, M. G. (2001). Error estimates in Sobolev spaces for moving least square approximations. **Siam Journal on Numerical Analysis**, v.39, n.1, p.38-51.
- BABOLIAN, E.; FATTAHZADEH, F. (2007). Numerical solution of differential equations by using Chebyshev wavelet operational matrix of integration. **Applied Mathematics and Computation**, v.188, n.1, p.417-426.
- BARAGLIA, R. et al. (2001). A hybrid heuristic for the traveling salesman problem. **Ieee Transactions on Evolutionary Computation**, v.5, n.6, p.613-622.
- BATTIN, R. H. (1976). Resolution of Runge-Kutta-Nystrom condition equations through 8th order. **Aiaa Journal**, v.14, n.8, p.1012-1021.
- BATTIN, R. H. (1977). Resolution of Runge-Kutta-Nystrom condition equations through 8th order - reply. **Aiaa Journal**, v.15, n.5, p.763-763.
- BERGEY, P. K.; RAGSDALE, C. (2005). Modified differential evolution: a greedy random strategy for genetic recombination. **Omega-International Journal of Management Science**, v.33, n.3, p.255-265.
- BERRUT, J. P.; TREFETHEN, L. N. (2004). Barycentric Lagrange Interpolation. **SIAM Rev.** 46, 501, [OI:10.1137/S0036144502417715](https://doi.org/10.1137/S0036144502417715).
- BOAS, M. L. (1983). *Mathematical Methods in the Physical Sciences*. 2ed, New York: John Wiley & Sons, 793p.
- BOYD, J. P. (1992). Defeating the Runge phenomenon for equispaced polynomial interpolation via Tikhonov regularization. **Applied Mathematics Letters**, v. 5, n. 6, pp. 57-59.
- BRYDEN, K. M. et al. (2006). Graph-based evolutionary algorithms. **Ieee Transactions on Evolutionary Computation**, v.10, n.5, p.550-567.
- BYRON, F. W.; FULLER, R. W. (1992). *Mathematics of classical and quantum physics*. New York: Dover Publications. 2 v, 661 p.
- CALVO, M. et al. (2008). Sixth-order symmetric and symplectic exponentially fitted modified Runge-Kutta methods of Gauss type. **Computer Physics Communications**, v.178, n.10, p.732-744.
- CAMPOS, F. F. F. (2007). *Algoritmos Numéricos*. Rio de Janeiro: Livros Técnicos e Científicos. 428 p.
- CAO, H. Q. et al. (2000). A two-level hybrid evolutionary algorithm for modeling one-dimensional dynamic systems by higher-order ODE models. **Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics**, v.30, n.2, p.351-357.

- CHAUDHURY, P.; BHATTACHARYYA, S. P. (1998). Numerical solutions of the Schrödinger equation directly or perturbatively by a genetic algorithm: test cases. **Chemical Physics Letters**, v.296, n.1-2, p.51-60.
- CHEB-TERRAB, E. S. (2002). ODE trends in computer algebra: four linear and nonlinear challenges. In **The Maple Summer Workshop**, Waterloo, Canada, 9p.
- CHEN, C. T. et al. (2008). Optimal design and control of CPU heat sink processes. **Ieee Transactions on Components and Packaging Technologies**, v.31, p.184-195.
- COELHO, L. S. (2008). Reliability-redundancy optimization by means of a chaotic differential evolution approach. **Chaos, solitons & fractals**.
- CUVAJ, C. (1968). Henri Poincaré's mathematical contributions to relativity and Poincaré stresses. **American Journal of Physics**, v.36, n.12, p.1102-&.
- DARWIN, C. (1952). *The origin of species*. Col. Great Books. Londres: Britannica, 660 p.
- DOHA, E. H.; BHRAWY, A. H. (2008). Efficient spectral-Galerkin algorithms for direct solution of fourth-order differential equations using Jacobi polynomials. **Applied Numerical Mathematics**, v.58, n.8, Aug, p.1224-1244.
- DORMAND, J. R.; PRINCE, P. J. (1980). A family of embedded Runge-Kutta formulae. **J. Comp. Applied Mathematics**, vol. 6, 1980, p. 19-26.
- DUAN, D. W. (1997). An efficient method for computing nodes and weights of Gaussian-Zernike quadrature. **Ieee Antennas and Propagation Society International Symposium 1997**, Vols 1-4, p.1626-1629.
- ELBARBARY, E. M. E. (2002). Legendre expansion method for the solution of the second- and fourth-order elliptic equations. **Mathematics and Computers in Simulation**, v.59, n.5, Jun, p.389-399.
- GARCIA, D. (2008). Function orthofit.m, código em Matlab para ajuste de pontos por polinômios ortogonais de alta ordem. (www.mathworks.com/matlabcentral/files/16574/orthofit) Acessado em 20/10/2008.
- GIBBS M. S.; DANDY, G.C.; MAIER H.R. (2008). A genetic algorithm calibration method based on convergence due to genetic drift. **Information Sciences**, v. 178, n.14, p.2857-2869.
- GRANEK, G. (2000). Poincaré's contributions to relativistic dynamics. **Studies in History and Philosophy of Moderns Physics**, v.31B, n.1, p.15-48.
- HAHN, R. (2005). *Pierre Simon Laplace, 1749-1827: a determined scientist*. Cambridge, Mass.: Harvard University Press., x, 310 p.
- HARRÉ, R. (1969). *Some nineteenth century British scientists*. Oxford, New York: Pergamon Press, viii, 259 p. (The Commonwealth and International Library. Liberal studies division. Science and society, v. 5.)
- HIGHAM, D. J. (1996). Runge-Kutta type methods for orthogonal integration. **Applied Numerical Mathematics**, v.22, n.1-3, p.217-223.

- HOLLAND, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Harbor, 130 p.
- HRSTKA, O. et al. (2003). A competitive comparison of different types of evolutionary algorithms. **Computers & Structures**, v.81, n.18-19, p.1979-1990.
- HRSTKA, O.; KUCEROVA, A. (2004). Improvements of real coded genetic algorithms based on differential operators preventing premature convergence. **Advances in Engineering Software**, v.35, n.3-4, p.237-246.
- HULL, T. E. et al. (1972). Comparing Numerical Methods For Ordinary Differential Equations. **Siam Journal on Numerical Analysis**, v.9, n.4, p.603-637.
- KAELO, P; ALI, M.M. (2007). Differential evolution algorithms using hybrid mutation. **Computational Optimization and Applications**, v.37, n.2, Jun, p. 231-246.
- KARR, C. L. et al. (2000). Solving inverse initial-value, boundary-value problems. **Engineering Applications of Artificial Intelligence**, v.13, n.6, p. 625-633.
- KEINER, J.; POTTS, D. (2008). Fast evaluation of quadrature formulae on the sphere. **Mathematics of Computation**, v.77, p.397-419.
- KESAN, C. (2003). Taylor polynomial solutions of linear differential equations. **Applied Mathematics and Computations**, v.142, n.1, Sep, p.155-165.
- KHELLAT, F.; YOUSEFI, S. A. (2006). The linear Legendre mother wavelets operational matrix of integration and its application. **Journal of the Franklin Institute-Engineering and Applied Mathematics**, v.343, p.181-190.
- KRINK; THIEMO; PATERLINI, S. (2006). Differential Evolution and Particle Swarm Optimization for Partitional Clustering. **Computational Statistics & Data Analysis**, 2006, p. 1220-1247.
- LARA, L. P.; GADELLA, M. (2008). An approximation to solutions of linear ODE by cubic interpolation. **Computers & Mathematics with Applications**, v.56, n.6, Sep, p.1488-1495.
- LEE, C. et al. (2007). A Posteriori Error Estimation and Adaptive Node Refinement for Fast Moving Least Square Reproducing Kernel (FMLSRLK) Method. **Computer Modeling In Engineering & Sciences**, v.20, n.1, p.35-41.
- LIM, H. J; IM, S.; CHO, Y-S. (2007). Variable-node elements for non-matching meshes by means of MLS (moving least-square) scheme. **International Journal for Numerical Methods and Engineering**, v. 72, p. 835-837.
- LIU, B.; LU, Jing; WANG, Yan; TANG, Yang. (2008). An effective parameter extraction method based on memetic differential evolution algorithm. **Microelectronics Journal**.
- LIU, F. B. (2008). A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source. **International Journal of Heat and Mass Transfer**, v.51, n.15-16, p. 3745-3752.

- LUTZ, E. (1992). Exact gaussian quadrature methods for near-singular integrals in the boundary element method. **Engineering Analysis with Boundary Elements**, v.9, n.3, p. 233-245.
- MEHRA, J.; RECHENBERG, H. (1982). *The historical development of quantum theory*. New York: Springer-Verlag, 366 p.
- MICHALEWICZ, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer Verlag, 330 p.
- MITCHELL, M. (1997). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.
- MOSCOVITCH, D. et al. (2004). Determination of a unique solution to parallel proton transfer reactions using the genetic algorithm. **Biophysical Journal**, v.87, n.1, p.47-57.
- MOURA, C. A. D. (2002). *Análise Funcional para aplicações - Posologia*. Rio de Janeiro: Ciência Moderna, 217 p.
- MUGHAL, Asif M.; YE, Xiu; IQBAL, Kamran. (2006). Computational algorithm for higher order Legendre polynomial and gaussian quadrature method. **2006 International Conference on Scientific Computing**. Las Vegas, p. 73-77, June 26-29.
- NAKANISHI, H.; SUGAWARA, M. (2000). Numerical solution of the Schrödinger equation by a microgenetic algorithm. **Chemical Physics Letters**, v.327, n.5-6, p.429-438.
- NOUGIER, J. P. (2001). Méthodes de calcul numérique 2. **Hermes Science Publications**. Section 4.7 p. 116-121.
- OBITKO, M. (1998). Genetic Algorithms. (<http://www.obitko.com/tutorials/genetic-algorithms>) Acessada 23/09/2008.
- O'CONNOR, J. J. E. F. R. (2008). The MacTutor of Mathematics archive. (<http://www-history.mcs.st-andrews.ac.uk/history/index.html>) Acessada em 20/09/2008.
- PSIHOYIOS, G.; SIMOS, T. E. (2003). Exponentially and trigonometrically fitted explicit advanced step-point (EAS) methods for initial value problems with oscillating solutions. **International Journal of Modern Physics C**, v.14, n.2, p.175-184.
- RASTEGAR, R.; HARIRI, A. (2006). A step forward in studying the compact Genetic Algorithm. **Evolutionary Computation**, v.14, n.3, p. 277-289.
- RAZZAGHI, M.; YOUSEFI, S. (2001). Legendre wavelets operational matrix of integration. **Int. J. Syst. Sci.** 32 (4), p. 495-502.
- ROUT, B. K.; MITTAL, R. K. (2008). Optimal manipulator parameter tolerance selection using evolutionary optimization technique. **Engineering Applications of Artificial Intelligence**, v.21, n.4, p. 509-524.
- SAYEED, M.; MAHINTHAKUMAR, G. K. (2005). Efficient parallel implementation of hybrid optimization approaches for solving groundwater inverse problems. **Journal of Computing in Civil Engineering**, v.19, n.4, p. 329-340.

- SENGUPTA, T. K. et al. (2007). Control of flow using genetic algorithm for a circular cylinder executing rotary oscillation. **Computers & Fluids**, v.36, n.3, p. 578-600.
- SEZER, M.; KESAN, C. (2000). Polynomial solutions of certain differential equations. **International Journal of Computer Mathematics**, v.76, n.1, Nov, p. 93-104.
- SHAMPINE, L. F.; REICHEL, M. W. (1997). The MATLAB ODE suite. **Siam Journal on Scientific Computing**, v.18, n.1, p. 1-22.
- SHEN, J. (1994). Efficient spectral-Galerkin method. 1. Direct solvers of 2nd-order and 4th-order equations using Legendre polynomials. **Siam Journal on Scientific Computing**, v.15, n.6, p. 1489-1505.
- _____ (1995). Efficient spectral-galerkin method. 2. Direct solvers of 2nd-order and 4th-order equations using Chebyshev polynomials. **Siam Journal on Scientific Computing**, v.16, n.1, p. 74-87.
- STORN, R. (1996). On the usage of differential evolution for function optimization, **NAPHIS**.
- STORN, R.; PRICE K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, 11: p. 341-359.
- THE MATHWORKS, INC., Matlab 7.2, 2006.
- TIETJEN, B.; HUTH, A. (2006). Modelling dynamics of managed tropical rainforests - An aggregated approach. **Ecological Modelling**, v.199, n.4, pp.421-432.
- WILSON, H.; TURCOTTE, L. H.; HALPERN, D. (2003). *Advanced Mathematics and Mechanics Applications using MATLAB*. 3ed, New York: Chapman & Hall/CRC.
- ZHU, J. et al. (2008). Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes. **Journal of Computational Physics**, v.227, n.9, p. 4330-4353.
- ZUPPA, C. (2003). Good quality point sets and error estimates for moving least square approximations. **Applied Numerical Mathematics**, v. 47, p. 575-585.