

---

# Métodos Numéricos

Um curso para o Mestrado Integrado em Engenharia Informática e  
Computadores da FEUP

---

Carlos Madureira  
Cristina Vila  
Maria de Lurdes Dinis  
José Soeiro de Carvalho

Faculdade de Engenharia da Universidade do Porto  
Departamento de Engenharia de Minas

Copyright ©1995-2019 Carlos M. N. Madureira, Maria Cristina C. Vila, Maria de Lurdes Dinis José Manuel S. Soeiro de Carvalho

Reservados todos os direitos de publicação, tradução e adaptação.

Interdita a reprodução parcial ou integral sem prévia autorização dos autores.

Este documento é uma versão provisória, de utilização exclusiva no âmbito da disciplina de Métodos Numéricos do Mestrado Integrado em Engenharia Informática e Computadores da Faculdade de Engenharia da Universidade do Porto.

Todas as correcções e contribuições são bem-vindas!

Carlos M. N. Madureira (cmad@fe.up.pt)

Maria Cristina C. Vila (mvila@fe.up.pt)

Maria de Lurdes Dinis (mldinis@fe.up.pt)

José M. S. Soeiro de Carvalho (jmsoeiro@fe.up.pt)

Versão 2.1 *Pública*

Impresso em 7 de Outubro de 2019

Baseado num documento obtido do commit GIT 3ef530c, feito por Jose\begin group \let \relax \relax \end group [Please insert \PrerenderUnicode{\IA} into preamble] Soeiro Carvalho em 2019-10-07 16:48:55 +0100

# Conteúdo

<b>Introdução</b>	<b>1</b>
<b>1 O Erro em Análise Numérica</b>	<b>7</b>
1.1 Matemáticas e Análise Numérica	8
1.2 A representação dos números numa máquina	10
1.2.1 Representação, Codificação	12
1.2.2 Representação de quantidades	13
1.2.3 Conversão entre bases	15
1.2.4 Notação	16
1.2.5 Representação IEEE	24
1.3 Erros	28
1.4 Arredondamento e Truncatura	28
1.4.1 Diferentes maneiras de calcular uma expressão	36
1.4.2 Os erros nos dados	41
1.4.3 Cálculo dos erros	45
1.5 Conclusão	48
<b>2 Zeros reais de uma função real</b>	<b>51</b>
2.1 Isolamento das raízes	55
2.2 Método da Bissecção	62
2.3 Método da Corda	66
2.4 Método da tangente	70
2.5 Método de iteração de Picard-Peano	72
2.6 Resolução de Sistemas pelo Método de Iteração	77
<b>3 Sistemas de Equações Lineares</b>	<b>81</b>
3.1 Eliminação Gaussiana	81
3.2 O Erro no Método de Gauss	85
3.3 Técnicas Clássicas para Minimização dos Erros	94
3.4 Ordem e Condição de um Sistema	95
3.5 Método de Khaletsky	102
3.6 Métodos iterativos	103
3.6.1 Método de Gauss-Jacobi	104
3.6.2 Método de Gauss-Seidel	105
<b>4 Quadratura e Cubatura</b>	<b>107</b>
4.1 Introdução	107

## Conteúdo

4.2	Regra dos Trapézios . . . . .	108
4.2.1	A "fórmula" do erro . . . . .	109
4.2.2	Controlo do erro . . . . .	111
4.3	Regra de Simpson . . . . .	113
4.3.1	A "fórmula" de erro . . . . .	114
4.3.2	Controlo do erro . . . . .	115
4.4	Integrais impróprios . . . . .	115
4.5	Integrais singulares . . . . .	117
4.6	Cubatura . . . . .	118
<b>5</b>	<b>Integração de equações diferenciais ordinárias</b>	<b>121</b>
5.1	O significado de uma solução: Método de Euler . . . . .	121
5.1.1	Método de Euler . . . . .	123
5.2	Um melhoramento do Método de Euler . . . . .	129
5.3	Métodos de Runge-Kutta . . . . .	131
5.3.1	Método de Runge-Kuta de Segunda Ordem . . . . .	131
5.3.2	Método de Runge-Kuta de Quarta Ordem . . . . .	132
5.4	Forma geral . . . . .	134
5.5	Sistemas de Equações e Equações de Ordem Superior . . . . .	134
<b>Índice</b>		<b>139</b>

## Lista de Figuras

1.1	Números racionais possíveis para o domínio inteiro $[-3,3]$ . . . . .	12
1.2	A verdadeira <b>MAKINA</b> . . . . .	20
2.1	Isolamento de raízes . . . . .	56
2.2	Isolamento de raízes II . . . . .	58
2.3	Isolamento de raízes III . . . . .	59
2.4	Interpretação geométrica do método da corda . . . . .	66
2.5	Aplicação do método da corda . . . . .	68
2.6	Convergência do método da corda . . . . .	68
2.7	Método da tangente . . . . .	70
2.8	Não convergência no método da tangente . . . . .	72
2.9	Método de Picard-Peano em escada . . . . .	74
2.10	Método de Picard-Peano em teia de aranha . . . . .	74
2.11	Método de Picard-Peano em escada divergente . . . . .	75
2.12	Método de Picard-Peano em teia de aranha divergente . . . . .	75
4.1	Regra de Simpson . . . . .	120



## Lista de Tabelas

1.1	Codificação de números e caracteres . . . . .	13
1.2	Notação posicional do mesmo valor em base 16 e base 10 . . . . .	14
1.3	Bases Numéricas . . . . .	15
1.4	Conversão de $100_{10}$ para base 3 . . . . .	16
1.5	Vírgula fixa e flutuante . . . . .	17
1.6	Comparação de notações em vírgula fixa e flutuante . . . . .	18
2.1	Isolamento de raízes . . . . .	61
2.2	Exemplo de bissecção sucessiva . . . . .	63
2.3	Aplicação do método da corda . . . . .	67
2.4	Aplicação do método da tangente . . . . .	71
2.5	Aplicação do método da tangente, com ponto inicial diferente. . . . .	71
3.1	Aplicação dos Métodos de Gauss-Jacobi e Gauss-Seidel . . . . .	106

# Introdução

Na prática da computação digital científica verificam-se numerosos e diversificados tipos de situações relacionadas com problemas de Análise Numérica. As que se revelam mais simples são aquelas em que se pretende construir um algoritmo para utilizar apenas uma vez, ou, quando muito, um número muito reduzido de vezes, para resolver um dado problema concreto e em que se tem uma ideia razoável do modo como vão desenvolver-se os cálculos. Nestes casos os problemas de *tempo de processamento* não se põem (pelo menos de modo agudo, sobretudo quando dispomos de um computador pessoal que podemos deixar a trabalhar durante a noite só para nós) e as *questões de precisão* são relativamente simples de tratar (até porque se dispõe, o deve dispor, *a priori* de ordens de grandeza para as soluções, o que permite controlar os erros mais óbvios); porém, a tentação de criar soluções extremamente dependentes das características particulares do problema e das peculiaridades da máquina é quase irresistível, de tal modo que os programas desenvolvidos não têm, em geral, qualquer transportabilidade.

Um segundo tipo de situação é o de um algoritmo destinado a correr diversas vezes em diversos tipos de situações todas incluídas em um mesmo contexto mas destinadas em princípio a um único utilizador, e difere da anterior no facto de existir uma potencial diversidade de dados de partida conducente a diferentes tipos de problemas numéricos. A principal dificuldade operacional reside, neste caso, no esforço de imaginação necessário para prever as diversas classes de situações que virão eventualmente a verificar-se, sob pena de se ter que rescrever sucessivas versões do mesmo algoritmo, uma para cada falhanço detectado. A situação torna-se nitidamente mais complicada quando o algoritmo se destina a ser corrido por diversos utilizadores, porque, nesse caso, é virtualmente impossível prever *a priori* o tipo de complicações numéricas que, em função de diferentes dados de partida, podem vir a ocorrer. Este é o caso típico dos *programas de biblioteca de uso geral*. Neste caso torna-se necessário usar dos maiores cuidados para cobrir tantas situações diferentes quantas seja possível, tendo ao mesmo tempo a habilidade de não produzir programas grosseiramente ineficientes ou tão complicados que desencorajem o utilizador comum. A real dificuldade de conciliar objectivos tão contraditórios e a subjectividade inevitável na ponderação das diferentes prioridades levam muitos utilizadores (incluindo os autores deste curso) a olhar com alguma desconfiança a tendência de certos outros para o uso sistemático de rotinas de biblioteca; o preço de uma tal atitude é, inevitavelmente, a necessidade de escrever os seus próprios algoritmos e programas.

Uma terceira classe de situações corresponde aos problemas computacionais tão grandes, tão complexos, tão particulares ou tão novos que se torna necessário, para os resolver, um conhecimento muito profundo da questão concreta que os gerou e o recurso à intuição física do proponente, que se supõe ser o seu melhor conhecedor. Esta classe de problemas dá, normalmente, origem a soluções particulares, que usam técnicas adequadas às peculiaridades de cada problema,



ou de cada formulação particular do problema. O uso dessas soluções por outras pessoas que não o seu próprio autor exige um estudo muito aprofundado que pode, frequentemente, revelar-se infrutífero e/ou impraticável, dada a conhecida tendência dos programadores, sobretudo dos mais criativos, para documentarem mal os seus produtos.

Finalmente, uma outra classe de situações ocorre nas aplicações em tempo real (como os programas de controlo de um processo industrial); nestes casos, o imperativo fundamental, para além de uma precisão (que, ao contrário do que poderia pensar-se, nem sempre será crítica) será o de uma execução extremamente rápida; em casos deste tipo, a rapidez da execução pode mesmo compensar em larga medida uma certa imprecisão dos resultados, desde que não exagerada. Tal como no caso anterior, deve fazer-se um esforço importante para garantir que se dispõe de toda a informação relevante sobre o problema e sobre os seus possíveis métodos de resolução, bem como de todo o tempo necessário para desenvolver uma solução realmente eficaz. Estes dois últimos casos constituem o domínio de eleição da aplicação das habilidades e artimanhas próprias da arte do analista e do programador.

Em resumo, existem três classes fundamentais de necessidades de algoritmos e programas computacionais:

- soluções expeditas, facilmente disponíveis ou implementáveis, embora possivelmente pouco eficientes, mas não imprecisas;
- soluções cuidadosamente pensadas, precisas, rápidas, seguras e de uso geral;
- soluções particulares para problemas particulares, em que as peculiaridades da aplicação em vista podem desempenhar papel central.

Como é óbvio, o presente curso só trata dos dois primeiros tipos de problemas. O nosso ponto de vista central será, portanto, o de que, embora o objectivo primário seja, como sempre, o de sacar resultados da máquina, esses resultados se destinam a fundamentar decisões, por vezes de importância vital, pelo que se torna necessário, antes de mais, compreender claramente e em profundidade o que tais números podem dizer, ou não dizer, o que podem valer, ou não valer. Assim, considerando que se trata de cadeira de mera introdução colocada muito cedo no plano do curso, a selecção de matérias tem em vista aquilo que de mais básico se prende com os objectivos centrais do exercício da profissão de engenheiro a nível superior, e não aquilo que está mais em voga ou mais bem documentado na literatura técnica e científica; secundariamente, a selecção das matérias nem pretende cobrir todas as necessidades que possam surgir no exercício da profissão, nem escolher os "melhores" métodos no sentido do uso eficiente dos recursos das máquinas - que, ainda por cima, mudam quase todos os dias -, mas apenas seleccionar aquelas matérias que melhor possam compensar o tempo e o esforço gastos no seu estudo.

Uma tal selecção é notavelmente dificultada

- pelo facto de *não existir, por trás da análise numérica, um corpo de doutrina coerente*, o que faz com que as soluções numéricas realmente eficientes e adaptadas sejam frequentemente produtos artesanais frutos de processos criativos baseados em toda uma experiência anterior do autor-utilizador;

- *pela enorme diversidade de métodos, técnicas, variantes e variações concebidas até ao presente* e que a maior parte dos tratadistas tende a apresentar a granel, sem qualquer senso crítico;
- pelo facto de muitas dessas técnicas serem *meros artifícios sem interesse geral*, facto que não pode ser revelado senão por um uso intensivo e constantemente crítico que o ensino convencional não encoraja.

A selecção de métodos aqui apresentada representa, portanto, um mero ponto de vista muito pessoal do autor, caucionado apenas pela experiência investigacional e docente da equipa que tem dirigido no Departamento de Engenharia de Minas ao longo das duas últimas décadas.

Por outro lado, o espaço de escolha encontra-se amputado pelo facto de existir no curriculum uma cadeira de Investigação Operacional, no âmbito da qual recaem os métodos cuja fundamentação é de carácter essencialmente algébrico - programação linear, algoritmos de árvores e redes, etc. Neste contexto, o capítulo sobre sistemas de equações lineares aparece como u, tanto estranho no curso, dado que a sua metodologia releva dos métodos algébricos.

A construção do curso no seu conjunto assenta sobre uma opção táctica muito clara, resultante de muitos anos de ensino da matéria ao nível da licenciatura: *total separação entre a Análise Numérica* - entendida como construção de algoritmos e análise do seu comportamento face à inevitável finitude da representação concreta dos números dentro da máquina - e *a programação desses algoritmos* no contexto de uma linguagem e/ou de uma máquina particular. Com efeito, toda a experiência passada nos mostrou para além de qualquer dúvida razoável que a mistura imprudente das duas abordagens e dos respectivos pontos de vista e polarizações faz com que o estudante médio, engodado pela preocupação, não de todo ilegítima mas limitativa, da produtividade imediata, concentre a totalidade do seu esforço no segundo e esqueça por completo o primeiro, aquele que constitui, precisamente, o objectivo primordial da cadeira; acontece que o ponto de vista peculiar da programação valoriza desproporcionadamente os problemas da lógica e da economia do algoritmo e tende a remeter para segundo plano os problemas vitais da precisão, problemas que são de carácter muito menos racionalizável e, por esse facto, de abordagem muito mais árdua para o estudante de formação racionalista.

Um outro ponto de vista didáctico que confere um certo grau de originalidade à estrutura da cadeira é o de se praticar sistematicamente um certo tipo de "reconstrução" da história dos métodos da Análise Numérica com a intenção de mostrar como os métodos se encadeiam logicamente uns com os outros e, nessa perspectiva, podem constantemente ser redesenhados, o que favorece uma das atitudes mentais mais importantes para o candidato a analista numérico.

A necessidade do uso, na cadeira, de algoritmos implementados em computador, indispensável para a aquisição do seu completo domínio pelo estudante, cria um dilema extremamente difícil de ultrapassar: por um lado, os cuidados necessários em matéria de precisão são pouco compatíveis com programação de principiantes e, por outro lado, como se depreende do que ficou dito no início, temos sérias reservas quanto ao uso cegos de programas de biblioteca. Por isso, todos os exemplos e manipulações exibidas e todos os exercícios propostos se destinam a ser trabalhados à mão ou à máquina de calcular ou, na melhor das hipóteses, em folha de calculo; uma tal estratégia, quando bem praticada, permite seguir passo a passo eventuais incidentes resultantes de

problemas de representação; no entanto, uma vez dominados os princípios e os conceitos, deve ser feito um esforço sério para formalizar a aprendizagem sob a forma de construção de programas informáticos que incorporem explicitamente todos os critérios de segurança e precaução que a teoria e a prática "à mão" aconselharam.

Finalmente, na concepção e implementação do curso, um outro problema se levantou: o da recente vulgarização dos *manipuladores simbólicos*, ferramentas informáticas que,

- ao contrário das vulgares linguagens de programação, têm a possibilidade de manipular directamente expressões simbólicas e não apenas números;
- permitem representar os números (dados, resultados intermédios e finais) com precisão "infinita" (dentro, naturalmente, dos limites de memória da máquina), representando os racionais sob a forma de quocientes de inteiros de comprimento adequado e mantendo em suspenso todas as operações irracionais; em opção, um modo de cálculo aproximado permite trabalhar com um número arbitrário de algarismos significativos.

Nestas condições, os manipuladores simbólicos permitem a verificação e o controlo do cálculo programado em linguagens convencionais e é um dos objectivos do curso levar o estudante a encará-los como tal.

Estamos mesmo em crer que, dentro da evolução previsível das capacidades de memória e das velocidades de cálculo dos computadores das próximas gerações, esta é uma das direcções genéricas em que se desenvolverá o cálculo científico nas próximas décadas, o que tornará obsoletas muitas das abordagens tradicionais da análise numérica. Pensamos, com efeito, que a generalização do uso dos manipuladores simbólicos corre o risco, se não for devidamente pensada, estudada e integrada, de vir a desempenhar, em relação ao ensino superior, o mesmo papel ambíguo, e por isso confusionalista e dissolvente, que a generalização do uso das calculadoras de bolso desempenhou, em relação ao secundário, nas décadas de 70 e 80. Por outro lado, há que reconhecer que tais ferramentas informáticas se encontram apenas na fase de utilitários para a realização de cálculos avulsos e não podem ainda ser eficientemente integradas em programas construídos pelo utilizador; isso impede, naturalmente, a completa centragem nelas de uma cadeira que não pode dar-se ao luxo de ignorar as reais necessidades da actual computação científica e técnica. Assim, incapazes, neste momento, de desenvolver um curso completo ao longo desta linha estratégica, os autores limitaram-se a fazer apenas algumas tímidas tentativas de introdução ao uso de manipuladores simbólicos muito acessíveis e que conhece como **derive**, A Mathematical Assistant <sup>TM</sup> [Sof]. ou **Maxima**, a Computer Algebra System [Max19] a propósito de alguns problemas mais delicados levantados pelo desenvolvimento de um curso que é ainda fundamentalmente convencional.

[DB74]

[CB81]

[Go191]

[Ham71]

[Lie68]

[Moo66]

[Mul89]

[Wal90]

