Apontamentos para a cadeira de

Optimização e Algoritmos Numéricos

Prof. Paulo Correia

December 8, 2005

Contents

Chapter 1

Introdução

A Análise Numérica é uma área especializada da Matemática directamente relacionada com a resolução de problemas matemáticos tendo como ferramenta principal o computador.
Os dois aspectos principais a considerar no cálculo computacional de soluções de problemas matemáticos são:

1. o desenvolvimento e a análise de métodos computacionais para a resolução dos problemas
2. a implementação destes métodos com o objectivo de executar cálculos científicos.

A estes métodos chamaremos Métodos Numéricos. Envolvem procedimentos e fórmulas que nos levam à aproximação de um problema matemático por um problema numérico ou à resolução de um problema numérico. A Análise Numérica é pois o campo que estuda o comportamento dos Métodos Numéricos. Tal estudo engloba a existência e unicidade de solução, a convergência, a estabilidade. etc.
O desenvolvimento de algoritmos numéricos a partir dos métodos numéricos é feito tendo em atenção o tipo e as características do problema e o contexto em que surge. O algoritmo deve conter todos os detalhes computacionais, nomeadamente em termos das condições iniciais e de paragem do procedimento. O conceito de método numérico é mais geral que o de algoritmo numérico.
Alguns dos algoritmos desenvolvidos para resolver problemas matemáticos padrão encontram-se implementados e incorporados em bibliotecas de programas de computação numérica. No entanto, e porque para certo tipo de problemas podem existir vários algoritmos, torna-se necessário seleccionar o algoritmo mais adequado para a resolução do problema entre mãos tendo em conta as suas especificações. Nos casos em que o problema matemático tem certas particularidades que o distinguem dos outros, pode não existir, nas bibliotecas numéricas, um programa que calcule a solução desse problema, sendo necessário ao utilizador desenvolver o seu próprio algoritmo/software. Nesta situação, o utilizador necessita de fazer um estudo mais pormenorizado sobre a aplicabilidade e estabilidade das técnicas numéricas.
Define-se algoritmo numérico como sendo um conjunto de directivas para executar operações matemáticas, criadas para obter a solução e um determinado problema matemático. Um algoritmo não fica bem definido se surgirem dúvidas quanto à operação que deve ser executada a seguir. Um algoritmo é pois uma técnica numérica para resolver um problema matemático.
Quando implementado como uma rotina ou num package, o algoritmo torna-se uma peça de software.
Quando se comparam algoritmos, para determinar o melhor, é habitual usar uma medida do custo do algoritmo. Essa medida pode ser baseada no número total de operações aritméticas executadas, como caracteriza a complexidade. A complexidade de um algoritmo está relacionada com o número

de operações elementares necessárias para resolver um tipo de problemas. No entanto, no estudo do desempenho dos algoritmos outras propriedades devem ser incluídas tais como: a estabilidade, o domínio de convergência e a eficiência.
Para a produção de rotinas eficientes, como elementos essenciais de software numérico, é necessário preencher um conjunto de requisitos:

1. A primeira etapa diz respeito à formulação do problema e consiste na definição do modelo matemático, bem como na especificação dos dados do problema, do tipo e da quantidade de resultados que se pretendem obter.

2. A segunda etapa envolve a selecção do método a usar. Define-se método como sendo o conjunto de fórmulas matemáticas que devem ser usadas par encontrar a solução do modelo matemático. Esta selecção compreende uma procura entre os métodos existentes ou mesmo a criação de algum método, caso se torne necessário. Nesta etapa temos de decidir por um método numérico aceitável isto é, por um método que nos resolva o problema de uma maneira tão económica e precisa quanto possível.

3. Uma vez decidido qual o método a usar, deve passar-se à terceira etapa que consiste em fazer a descrição detalhada do processo computacional, gerando o algoritmo numérico.

4. A última etapa consiste em converter o algoritmo num elemento de software numérico. É pois a realização física do algoritmo. Consiste num programa de computador e devem ser utilizadas, da melhor maneira possível, todas as capacidades computacionais do sistema.

   O utilizador pode usar uma linguagem máquina ou uma linguagem orientada para o problema, de alto nível, mais simples, cujas instruções estão mais perto da linguagem corrente falada.

De uma forma informal, uma linguagem de programação é um vocabulário especial que um computador compreende e que permite ao utilizador comunicar com o computador. Usando este vocabulário, o utilizador constrói uma sequência de instruções ou comandos os quais constituem um programa que pode ser submetido ao computador para execução.
As linguagens de programação estão tradicionalmente divididas em duas classes: linguagens de baixo nível e linguagens de alto nível. As linguagens de baixo nível, referidas muitas vezes como código máquina ou linguagem assembly, são primárias e requerem um conhecimento detalhado do funcionamento do computador como uma máquina. Para uma tarefa elementar o programa pode requerer numerosas linhas de instruções em código máquina. Por este motivo, foram desenvolvidas as linguagens de alto nível de forma a permitir ao utilizador escrever escrever instruções de forma mais simples sem conhecer os detalhes exactos de como o computador funciona realmente.
As instruções escritas em linguagens de programação de alto nível são traduzidas ou compiladas em código máquina antes de serem submetidas ao computador para execução. Algumas dessas linguagens de programação de alto nível são o FORTRAN, Pascal ou C.
Iremos utilizar um sistema algébrico computacional chamado Maple como linguagem de programação. Outros sistemas de cálculo de distribuição comercial e comparáveis ao Maple são, por exemplo, o Matlab, Mathematica e Mathcad. No caso de distribuição livre existem o Scilab (www.scilab.org/), Octave (www.octave.org) e Maxima (maxima.sourceforge.net) entre outros.
Um sistema algébrico computacional é diferente de uma linguagem de alto nível no sentido em que é na verdade um programa escrito numa linguagem de alto nível (como o C no caso do Maple) que está em execução contínua em background aguardando a introdução de dados (input) por parte do utilizador.

Assim, quando submetemos uma sequência de instruções ao Maple, as instruções são traduzidas no correspondente código C o qual é compilado e executado. Por este motivo, os programas em Maple não são eficientes quando se trata de utilização científitica, industrial ou comercial em larga escala. Contudo, o Maple é muito útil para a aprendizagem dos conceitos básicos de programação e para a auxiliar na compreensão de como construir algoritmos complexos.

Chapter 2

Aritmética computacional

2.1 Valor aproximado e erro

Exemplo 2.1 *Quando uma calculadora representa o valor de* π *por* 3.141592654 *ou o número de Neper, e, por* 2.718281828 *está a efectuar um arredondamento.*

Para a exigência do problema em causa o valor aproximado de π dado por 3.14 poderá ser suficiente. Isto quer dizer que o erro que cometemos ao considerar este valor é negligenciável.
Vários conceitos estão envolvidos no exemplo que demos: valor exacto, valor aproximado, erro.
Os dados e parâmetros de um problema são muitas vezes resultado de medições experimentais e, portanto, são afectados de alguma incerteza, provocando os chamados *erros inerentes*.

2.2 Representação de números no computador

> On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after lift-off. The rocket was on its first voyage, after a decade of development costing $7 billion. The destroyed rocket and its cargo were valued at $500 million. A board of inquiry investigated the causes of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system. Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer. The number was larger than 32,768, the largest integer storeable in a 16 bit signed integer, and thus the conversion failed.
> http://www.ima.umn.edu/ arnold/455.f97/notes.html
> ver James Gleick, http://www.around.com/ariane.html

A maioria dos computadores utiliza o sistema numérico binário ou algumas das suas variantes, tais como a base 8 ou a base 16. No entanto, o processo de efectuar as operações elementares é exactamente o mesmo, qualquer que seja a base que se considere. Como exemplo da representação binária e da sua conversão à base decimal, consideremos

$$\begin{aligned}(1101.101)_2 &= \mathbf{1}\times 2^3+\mathbf{1}\times 2^2+\mathbf{0}\times 2^1+\mathbf{1}\times 2^0+\mathbf{1}\times 2^{-1}+\mathbf{0}\times 2^{-2}+\mathbf{1}\times 2^{-3}\\ &= 13.625\end{aligned}$$

O primeiro número está escrito em base binária e o último em decimal. Visto que vamos trabalhar sempre na base decimal eliminaremos a indicação da base.

Assim, a representação de um número x

$$x = \pm\, d_n d_{n-1} \ldots d_1 d_0 \,.\, d_{-1} d_{-2} \ldots d_{-p}$$

significa que a parte inteira corresponde a

$$d_n \times 10^n + d_{n-1} \times 10^{n-1} + \cdots + d_1 \times 10^1 + d_0 \times 10^0$$

com $d_i \in \{0, 1, \ldots, 9\}$, $i = 0, \ldots, n$, e a parte fraccionária com p casas decimais é dada por

$$d_{-1} \times 10^{-1} + d_{-2} \times 10^{-2} + \cdots + d_{-p} \times 10^{-p}.$$

Normalmente representamos um número real em notação científica, do seguinte modo,

$$x = \pm \underbrace{0.\, d_1 d_2 \cdots d_n \cdots}_{\text{mantissa}} \times 10^t.$$

Esta representação de um número não é única. Por exemplo,

$$62.3 \times 10^{-6} = 623 \times 10^{-7} = 0.00623 \times 10^{-2}.$$

Diremos que se trata de *mantissa normalizada* quando $0.1 \leq m < 1$ e $d_1 \neq 0$. Quando nos referimos aos *dígitos da mantissa* referimo-nos à mantissa normalizada. No exemplo anterior, diríamos que a mantissa tem três dígitos e a representação normalizada seria 0.623×10^{-4}.
Quando um número na forma (?) é armazenado no computador, apenas um número finito de dígitos pode ser usado para representar a mantissa. Isto é, a sequência de dígitos que representa a mantissa $m = 0.\, d_1 d_2 \ldots d_n d_{n+1} \ldots$ deve ser finita dando origem à representação no formato de ponto flutuante do número.

DEFINIÇÃO 2.1 *Define-se formato de ponto flutuante de um número x a representação*

$$fl(x) = \pm\, 0.\, d_1 d_2 \ldots d_p \times b^t$$

onde $0.\, d_1 d_2 \ldots d_n$ define a mantissa do número, com p dígitos, t o expoente. O sistema constituído por todos os números neste formato e pelo 0 é identificado por FP(p,q) onde p indica o número de dígitos da mantissa e q o número máximo de algarismos do expoente.

Um sistema de ponto flutuante $\mathrm{FP}(\beta, p, t_1, t_2)$ é um subconjunto finito de números racionais, que se caracteriza pela base β (usualmente trabalhamos com base decimal, $\beta = 10$, mas internamente as máquinas usam a base binária, $\beta = 2$) pelo números de dígitos na mantissa e por uma limitação nos expoentes que pode tomar valores entre t_1 e t_2.

DEFINIÇÃO 2.2 *O sistema de ponto flutuante $FP(10, p, t_1, t_2)$ é constituído por todos os números racionais x tais que:*

$$x = 0$$
$$ou$$
$$x = \pm\, 0.d_1 d_2 \cdots d_p \times 10^t, \quad d_i \in \{0, \ldots, 9\}, \quad d_1 \neq 0, \quad t \in \{t_1, \ldots, t_2\}$$

e designa-se por sistema de ponto flutuante em base decimal, com p dígitos na mantissa e expoentes variando entre t_1 e t_2.

Exemplo 2.2 *O sistema* $FP(10, 3, -2, 2)$ *compreende o 0 e todos os números racionais da forma*

$$x = \pm\, 0.d_1d_2d_3 \times 10^t, \quad d_i \in \{0, \ldots, 9\}, \quad d_1 \neq 0, \quad t \in \{-2, \ldots, 2\}.$$

O maior número representável neste conjunto é ???? e o menor é ????.

Coloca-se agora a questão de saber, dado um número real

$$x = \pm\, 0.\, d_1d_2 \cdots d_n \cdots \times 10^t$$

qual o número aproximado, que denotaremos por $\mathrm{fl}(x)$, que o representa num sistema de ponto flutuante $\mathrm{FP}(10, p, t_1, t_2)$?
O formato de ponto flutuante obtém-se por dois processos. Um deles corta todos os dígitos que aparecem na mantissa à direita de d_k, da posição k, originando

$$\mathrm{fl}_t(x) = \pm\, 0.\, d_1d_2 \ldots d_k \times b^t$$

em que k depende do computador utilizado. Este processo é conhecido por *corte* (ou *truncatura*). No processo de corte a representação do número x, no formato de ponto flutuante normalizado, em que o primeiro dígito da mantissa d_1 é diferente de zero, é obtida como sendo a quantidade mais próxima de x que se encontra entre x e 0. Ou seja,

$$\widetilde{x} = 0.d_1d_2 \cdots d_p \times 10^t.$$

O outro processo conhecido por arredondamento simétrico, adiciona 5 unidades ao dígito $k+1$, aplicando em seguida o processo de truncatura a x. Isto é equivalente a truncar, depois do dígito d_k se o d_{k+1} é menor do que 5, ou adicionar uma unidade a d_k, se d_{k+1} for for maior ou igual a 5, truncando de seguida. Ou seja,

$$\widetilde{x} = \begin{cases} \pm\, 0.d_1d_2 \cdots d_p \times 10^t & \text{se } 0 \leq d_{p+1} < 5 \\ \pm\, (0.d_1d_2 \cdots d_p + 0.00 \cdots 01) \times 10^t & \text{se } 5 \leq d_{p+1} < 10. \end{cases}$$

O *Maple* tem três estruturas básicas para a representação de números, dependendo se se trata de números inteiros ou racionais. Estes três tipos são: `integer`, `fraction` e `float`. O tipo `integer` tem vários subtipos como por exemplo, `nonnegint` e `posint`. Um inteiro não-negativo é representado por uma sequência de algarismos na base decimal e um inteiro negativo é representado por uma sequência de algarismos na base decimal precedida de um sinal 'menos'. O comprimento máximo de um `integer` depende do sistema utilizado. Um sistema de 32 bits limita o comprimento a 524 280 algarismos enquanto que num sistema de 64 bits o limite são 38 654 705 646 algarismos.
Um número racional é representado, no tipo `fraction`, como um par de inteiros (numerador e denominador) com os factores comuns removidos e com denominador positivo. O tipo `rational` contém os tipos `integer` e `rational`.
Os números inteiros e racionais podem ser introduzidos no Maple nas suas representações na base decimal ou sob a forma de fracção e são então armazenados internamente com os tipos `integer` e `fraction`, respectivamente.
Um número racional pode ser também introduzido usando a notação de ponto decimal ou a notação científica e, neste caso, é armazenado com o tipo `float`. Por exemplo, `a:=-63741.8973, b:=637418973E-4 e c:=6.37418973e4` dão o mesmo valor numérico a `a`, `b` e `c`.

2.3 ERROS NA ARITMÉTICA EM PONTO FLUTUANTE

Um outro tipo de erro é o que decorre da utilização do computador para representar um número.

EXEMPLO 2.3 *Quando uma calculadora representa o valor de π por 3.141592654 ou o número de Neper, e, por 2.718281828 está a efectuar um arredondamento.*

DEFINIÇÃO 2.3 (ARREDONDAMENTO E ERRO DE ARREDONDAMENTO) *O processo de substituição de um número x pelo formato de ponto flutuante $\mathrm{fl}(x)$ chama-se arredondamento. À diferença $\mathrm{fl}(x) - x$ chama-se erro de arredondamento.*

DEFINIÇÃO 2.4 (ERRO ABSOLUTO E ERRO RELATIVO) *Seja x o valor exacto duma grandeza real e seja $\widetilde{x}$ um valor aproximado de x.*
Ao valor

$$\Delta := |x - \widetilde{x}|$$

chama-se erro absoluto de $\widetilde{x}$ em relação a x. Se $x \neq 0$, define-se o erro relativo de $\widetilde{x}$ em relação a x por

$$\delta := \frac{|x - \widetilde{x}|}{|x|}$$

Ao produto $100\,\delta$ expresso em percentagem, chama-se percentagem de erro.

EXEMPLO 2.4 *Calcule o erro absoluto e o erro relativo*

EXEMPLO 2.5 *Consideremos os números $x = 1/3$ e $y = 1/3000$, e as aproximações $\widetilde{x} = 0.3333$ e $y = 0.0003$. Verificamos que $\widetilde{x}$ e $\widetilde{y}$ são valores aproximados de x e y, respectivamente, com o mesmo erro absoluto*

$$\left|0.333333\cdots - 0.3333\right| = \left|0.0003333333\cdots - 0.0003\right| = 0.0000333\cdots$$

contudo, a percentagem de erro no primeiro caso, é

$$\frac{0.0000333\cdots}{0.3333} \times 100 = 0.01\% \quad e$$

$$\frac{0.0000333\cdots}{0.0003} \times 100 = 0.10\% \quad \textit{no segundo caso.}$$

(unidade de arredondamento)

DEFINIÇÃO 2.5 (ALGARISMOS SIGNIFICATIVOS) *Uma aproximação $\widetilde{x}$ diz-se ter m algarismos significativos em relação ao valor exacto x se m é o maior inteiro não-negativo para o qual o error relativo no número é 5×10^{-m}.*

EXEMPLO 2.6 (ASAITHAMBI,20) *Considere $x = 0.02136$ e $\widetilde{x} = 0.02147$. Então*

$$\frac{|x - \widetilde{x}|}{|x|} = \frac{0.00011}{0.02136} \approx 0.00515 \leq 5 \times 10^{-2}.$$

Portanto, $\widetilde{x}$ tem dois algarismos significativos em relação a x.

> Na aritmética em ponto flutuante do *Maple*, o resultado de cada operação aritmética básica, se não for demasiado grande ou demasiado pequeno, é arredondado correctamente de acordo com o valor definido pela variável `Digits`. O número de algarismos usado para representar um `float` é definido usando a variável `Digits`.
> Podemos tentar minimizar os erros de arredondamento nos cálculos aumentando a precisão dos números em ponto flutuante usando a variável `Digits`. Por defeito, o seu valor é `Digits:=10`. [Betounes & Redfern,162]

Definição 2.6 (ε da máquina) *O mais pequeno número ε em ponto flutuante que quando adicionado ao número em ponto flutuante* 1.0 *produz um número diferente de* 1.0 *diz-se número epsilon da máquina.*

As propriedades associativa e distributiva não são verificadas na aritmética de ponto flutuante.

Exemplo 2.7 (não associatividade da adição) *Consideremos*

> O *Maple* pode efectuar dois tipos de aritmética: exacta e em ponto flutuante. A aritmética exacta tem a vantagem de não apresentar erro de arredondamento. Contudo, pode ser longa e complicada ou mesmo, não elucidativa. [Betounes& Redfern,162]

2.4 Algoritmos e propagação de erros

Enquanto que a representação de números em ponto flutuante resulta num erro de arredondamento efectuar operações aritméticas num computador resulta na propagação do erro de arredondamento. Isto passa-se principalmente porque, em geral, o resultado de uma operação aritmética efectuada entre dois números em ponto flutuante do mesmo tamanho não é um número desse tamanho. um erro introduzido num passo durante um cálculo pode ser ampliado ou reduzido em operações posteriores. A isto se chama propagação do erro.

Exemplo 2.8 *Se* $x = 0.300 \times 10$, $y = 0.852 \times 10^{-6}$ *e* $z = 0.400 \times 10$ *então*

$$x + y = 0.3000000852 \times 10 \quad e \quad \frac{z}{x} = 0.133\cdots \times 10.$$

Sejam x e y os valores exactos de dois números com correspondentes representações em ponto flutuante $\widetilde{x} := \mathrm{fl}(x)$ e $\widetilde{y} := \mathrm{fl}(y)$. Representemos pelo símbolo $\circledast$ uma das operações $+, -, \times, \div$. Então $\mathrm{fl}(\widetilde{x} \circledast \widetilde{y})$ será o resultado aproximado enquanto que o valor exacto será $x \circledast y$. Por conseguinte, o erro no cálculo será

$$x \circledast y - \mathrm{fl}(\widetilde{x} \circledast \widetilde{y}) = \underbrace{(x \circledast y - \widetilde{x} \circledast \widetilde{y})}_{\text{erro propagado}} + \underbrace{(\widetilde{x} \circledast \widetilde{y} - \mathrm{fl}(\widetilde{x} \circledast \widetilde{y}))}_{\text{erro de arredondamento}}$$

O erro de arredondamento é facilmente estimado usando a relação (??). Para a estimação dos erros propagados existem diversos métodos. Um desses métodos é a aritmética de intervalos. Cada número é representado por um par de números em ponto flutuante, um minorante e um majorante. O resultado de cada operação básica fica assim compreendido no intervalo calculado.

Exemplo 2.9 (análise de intervalos) *Suponhamos que* $x = 0.234 \times 10$ *e* $y = 0.171 \times 10$

Um outro método recorre ao teorema do valor médio para as derivadas
Desta análise, podemos concluir que os erros absolutos podem propagar-se rapidamente quando multiplicamos por um número muito grande ou dividimos por um número muito pequeno. Por outro lado, para a adição e subtracção os erros absolutos não se propagam rapidamente. Os erros relativos podem propagar-se rapidamente especialmente quando calculamos a diferença entre duas quantidades muito próximas, resulatndo na perda de muitos algarismos significativos.

EXEMPLO 2.10 (CANCELAMENTO SUBTRACTIVO) *Consideremos o cálculo da seguinte expressão*

2.5 ESTABILIDADE

Dois dos conceitos mais importantes em Análise Numérica são o de estabilidade matemática e o de estabilidade numérica. A primeira é uma propriedade do problema matemático que vamos resolver e a segunda diz respeito ao algoritmo.
Um problema matemático diz-se bem-condicionado se pequenas perturbações introduzidas nos dados originarem apenas pequenas alterações na solução do problema. Neste caso, também se diz que o problema é matematicamente estável.
Por sua vez, se os resultados se apresentarem muito sensíveis a pequenas perturbações introduzidas nos dados, o problema diz-se mal-condicionado ou inerentemente instável.
Para problemas bem definidos em que os resultados dependem dos dados de uma forma contínua, é possível introduzir e definir uma medida para determinar o grau de condicionamento. Essa medida chama-se número de condição.
A estabilidade numérica é uma propriedade dos algoritmos. Podem existir várias maneiras de organizar os cálculos que levam à resolução de um problema matemático. Processos numéricos distintos para resolver o mesmo problema matemático podem comportar-se de maneira diferente em relação à propagação dos erros de arredondamento. Esta sensibilidade maior ou menor relativa às operações de arredondamento chama-se estabilidade numérica. Os erros de arredondamento propagam-se de operação para operação indo influenciar o resultado final. Se estes erros influenciam fortemente o resultado final o algoritmo diz-se numericamente instável.
Para a resolução de um problema bem-condicionado e dependendo da maneira como se encontram organizadas as operações, um algoritmo pode ser estável se os erros inerentes e de arredondamento não se acumularem exageradamente, ou instáveis se esses mesmos erros se acumularem e se se propagarem exageradamente por forma a afectarem significativamente os resultados finais. Por outro lado, se o problema for mal-condicionado nenhum algoritmo conseguirá atenuar a acumulação dos erros inerentes e de arredondamento.

Chapter 3

Resolução de equações não-lineares

3.1 Processo iterativo e ordem de convergência

3.2 Localização de raízes reais

3.3 Método da bissecção

DADOS: f, a, b, ε

INICIALIZAÇÃO: $a^{(0)} := a$, $b^{(0)} := b$

CICLO: Para $k = 0, 1, \ldots$ fazer

$x^{(k)} := \dfrac{a^{(k)} + b^{(k)}}{2}$

Se $\left|a^{(k)} - x^{(k)}\right| \leq \varepsilon$ ou $f(x^{(k)}) = 0$

Então zerodef $:= x^{(k)}$. STOP

Senão

Se $f(a^{(k)})\, f(x^{(k)}) < 0$

Então $a^{(k+1)} := a^{(k)}$, $b^{(k+1)} := x^{(k)}$

Senão $a^{(k+1)} := x^{(k)}$, $b^{(k+1)} := b^{(k)}$

FIM DO CICLO

SAÍDA: k, zerodef

Algoritmo 3.1: *Método da bissecção*

Este método tem a vantagem de convergir sempre para uma solução e, por outro lado, é fácil determinar

um majorante do número de iterações neessárias para garantir uma certa precisão. Por estes motivos, o método da bissecção é frequentemente usado como uma primeira aproximação por outros métodos mais eficientes.

EXERCÍCIO 3.1 *Implemente o algoritmo* 3.1 *através de um procedimento em Maple.*

Ao avaliar a condição $f(a^{(k)})\,f(x^{(k)}) < 0$ no *Maple* é preferível usar a *função sinal* definida por

$$\operatorname{sgn}(x) = \begin{cases} -1, & \text{se } x < 0 \\ 0, & \text{se } x = 0 \\ 1, & \text{se } x > 0, \end{cases}$$

a fim de evitar a ocorrência de *overflow* ou *underflow* ao ser efectuado o produto de $f(a^{(k)})$ por $f(x^{(k)})$, isto é,

$$\operatorname{sgn}(f(a^{(k)}))\operatorname{sgn}(f(x^{(k)})) < 0.$$

EXERCÍCIO 3.2 (F-B,34) *Justifique porque é que, para determinar o ponto médio do intervalo* $[a^{(k)}, b^{(k)}]$, *é aconselhável usar a expressão*

$$x^{(k)} = a^{(k)} + \frac{b^{(k)} - a^{(k)}}{2},$$

em vez da equação algebricamente equivalente,

$$x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}.$$

3.4 MÉTODO DA FALSA POSIÇÃO

$$x^{(k+1)} = b^{(k)} - f(b^{(k)})\frac{b^{(k)} - a^{(k)}}{f(b^{(k)}) - f(a^{(k)})} \tag{3.4.1}$$

Fazendo $a^{(k+1)} := x^{(k+1)}$ ou $b^{(k+1)} := x^{(k+1)}$ e mantendo o outro extremo do intervalo inalterado conforme $f(x^{(k+1)})\,f(b^{(k)}) < 0$ ou $f(a^{(k)})\,f(x^{(k+1)}) < 0$, e procedendo de igual modo, podemos obter uma nova aproximação do zero x^*.

EXERCÍCIO 3.3 *Implemente o algoritmo* 3.2 *através de um procedimento em Maple.*

EXERCÍCIO 3.4 *O método da falsa posição tende a desenvolver uma convergência lenta quando um dos extremos do intervalo se imobiliza. Modifique o método da falsa posição de modo a evitar que um dos extremos se imobilize sucessivamente. (Sugestão: divida o valor da função por 2 no extremo que se imobiliza.)*

DADOS: f, a, b, ε

INICIALIZAÇÃO: $a^{(0)} := a$, $b^{(0)} := b$, $x^{(0)} := b^{(0)}$

CICLO: Para $k = 0, 1, \ldots$ fazer

$$x^{(k+1)} := b^{(k)} - f(b^{(k)}) \frac{b^{(k)} - a^{(k)}}{f(b^{(k)}) - f(a^{(k)})}$$

Se $\left|x^{(k+1)} - x^{(k)}\right| \leq \varepsilon \left|x^{(k+1)}\right|$

Então zerodef $:= x^{(k+1)}$. STOP

Senão

Se $f(a^{(k)})\, f(x^{(k+1)}) < 0$

Então $a^{(k+1)} := a^{(k)}$, $b^{(k+1)} := x^{(k+1)}$

Senão $a^{(k+1)} := x^{(k+1)}$, $b^{(k+1)} := b^{(k)}$

FIM DO CICLO

SAÍDA: k, zerodef

Algoritmo 3.2: *Método da falsa posição*

3.5 Método de Newton-Raphson

Newton explicou o seu método em 1669 através do uso de exemplos numéricos. Não usou a ilustração geométrica de uma curva aproximada pela sua tangente, nem no seu trabalho é apresentada a fórmula de recorrência correntemente utilizada hoje me dia. Esta última, que evita a necessidade de meticulosas substituições, foi desenvolvida por Raphson em 1690, o qual partilha com Newton a designação do método.
A questão fundamental de saber qual deve ser a condição inicial de forma a garantir a convergência das aproximações, só foi colocada explicitamente em 1768 por Mourraille[1] e, posteriormente, por Lagrange. Neste caso, uma abordagem geométrica foi já utilizada para explorar a questão. [Chabert et al., 170]
No seu Method of Fluxions, Newton dá uma indicação do número de casas decimais obtidas em cada passo, mas o seu raciocínio é empírico e não analítico.[Chabert et al., 183]

Mourraille, J.-R., *Traité de la résolution des équations en général*, London & Marseille, 1768.

Exercício 3.5 *Implemente o algoritmo* 3.3 *através de um procedimento em Maple.*

DADOS: f, x_{inicial}, ε

INICIALIZAÇÃO: $x^{(0)} := x_{\text{inicial}}$

CICLO: **Para** $k = 0, 1, \ldots$ **fazer**

$$x^{(k+1)} := x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

Se $\left|x^{(k+1)} - x^{(k)}\right| \leq \varepsilon \left|x^{(k+1)}\right|$

Então zerodef $:= x^{(k+1)}$. **STOP**

FIM DO CICLO

SAÍDA: k, zerodef

Algoritmo 3.3: *Método de Newton*

3.5.1 Zeros múltiplos

DEFINIÇÃO 3.1 *Uma função f diz-se ter um zero de multiplicidade m num ponto a se f pode ser escrita na forma*

$$f(x) = (x-a)^m h(x)$$

onde h verifica $\lim_{x\to a} h(x) \neq 0$.

Um zero de multiplicidade $m = 1$ diz-se um zero simples, enquanto que se $m > 1$ se diz um zero múltiplo. Temos a seguinte caracterização dos zeros múltiplos.

TEOREMA 3.1 *Um número a é um zero de multiplicidade m de f se e só se*

$$f(a) = f'(a) = \cdots = f^{m-1}(a) = 0.$$

EXEMPLO 3.1 (ASAITHAMBI,95) *Mostre que $f(x) = e^{2x} - 1 - 2x - 2x^2$ tem um zero de multiplicidade 3 em $a = 0$ (ver figura ??).*

O método de Newton é um método linear no caso de um zero múltiplo.

EXERCÍCIO 3.6 (MC,51) *Determinar, aplicando o método de Newton, o zero duplo da função polinomial $f(x) = x^3 - 3.08x^2 + 3.1236x - 1.03968$ com um erro inferior a $\varepsilon = 5 \times 10^{-6}$. Considere uma aproximação inicial $x^{(0)} = 1$.*

O método de Newton pode ser modificado para o caso de um zero de multiplicidade m recuperando a convergência quadrática, através de

$$x^{(k+1)} = x^{(k)} - m\frac{f(x^{(k)})}{f'(x^{(k)})} \tag{3.5.2}$$

Exercício 3.7 *Resolva o exercício* 3.6 *aplicando o método de Newton modificado com a fórmula de iteração* (3.5.2).

Exercício 3.8 (F-B,47) *O método numérico definido por*

$$x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})\, f'(x^{(k-1)})}{\left(f'(x^{(k-1)})\right)^2 - f(x^{(k-1)})\, f''(x^{(k-1)})}$$

para $k = 1, 2, \ldots$, *pode ser aplicado em vez do método de Newton para equações com raízes múltiplas. Resolva o exercício* 3.6 *usando este método. Compare os resultados com os obtidos nos exercícios anteriores.*

Ver [Mathews,90] método de Halley: convergência cúbica!

3.6 Método da secante

No método da secante o zero de f é aproximado pelo zero da secante que passa por dois pontos de coordenadas $(x^{(k)}, f(x^{(k)}))$ e $(x^{(k-1)}, f(x^{(k-1)}))$. Estes dois pontos são duas aproximações do zero de f e os seus valores estão suficientemente próximos um do outro (Figura ??).

DADOS: f, x_{inicial}, ε

INICIALIZAÇÃO: $x^{(0)} := x_{\text{inicial}}$, $x^{(1)} := x_{\text{inicial}} + \epsilon_0$

CICLO: Para $k = 1, \ldots$ fazer

$$x^{(k+1)} := x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}$$

Se $\left|x^{(k+1)} - x^{(k)}\right| \leq \varepsilon \left|x^{(k+1)}\right|$

Então zerodef $:= x^{(k+1)}$. STOP

FIM DO CICLO

SAÍDA: k, zerodef

Algoritmo 3.4: *Método da secante*

3.7 Método do ponto fixo

Exercício 3.9 *Implemente o algoritmo* 3.5 *através de um procedimento em Maple.*

Exercício 3.10 *A convergência do método iterativo do ponto fixo é, em geral, linear. No entanto, é possível acelerar a convergência por um processo devido a Aitken. Para aplicar a aceleração de Aitken,*

DADOS: g, x_{inicial}, ε

INICIALIZAÇÃO: $x^{(0)} := x_{\text{inicial}}$

CICLO: Para $k = 1, \ldots$ **fazer**

$x^{(k+1)} := g(x^k)$

Se $\left|x^{(k+1)} - x^{(k)}\right| \leq \varepsilon \left|x^{(k+1)}\right|$

Então zerodeg $:= x^{(k+1)}$. **STOP**

FIM DO CICLO

SAÍDA: k, zerodeg

Algoritmo 3.5: *Método do ponto fixo*

necessitamos de três iteradas $x^{(k-1)}$, $x^{(k)}$ *e* $x^{(k+1)}$. *Assim, partindo de uma iterada* $x^{(k-1)}$, *calculamos duas novas iteradas* $x^{(k)} = g(x^{(k-1)})$ *e* $x^{(k+1)} = g(x^{(k)})$ *e depois podemos utilizar a expressão*

$$\widehat{x}^{(k+1)} = x^{(k-1)} - \frac{\left(x^{(k)} - x^{(k-1)}\right)^2}{\left(x^{(k+1)} - x^{(k)}\right) - \left(x^{(k)} - x^{(k-1)}\right)} \tag{3.7.3}$$

para determinar $\widehat{x}^{(k+1)}$.
Adapte o algoritmo 3.5 *aplicando a aceleração de Aitken.*

3.8 Zeros de polinómios

No início do século XIX, aparentemente de forma independente, três matemáticos desenvolveram técnicas engenhosas para a transformação de polinómios. Foram eles Ruffini (1804), Budan (1807 e 1813) e Horner (1819). Estas técnicas combinadas com os resultados de localização dos zeros de um polinómio, permitiam determinar aproximações da raíz pretendida com consideráveis reduções nos cálculos.
Apesar de ser possível encontrar traços deste método num texto de Newton, será apenas no século XIX que aparece um método sistemático de encontrar todos os coeficientes de um polinómio transformado $Q(x) = P(x + u)$.
Parece ter sido Paolo Ruffini o primeiro, pelo menos na Europa, a formular um algoritmo para a transformação dos coeficientes de uma equação.
O algoritmo definido por Ruffini permite-nos usar os $m + 1$ coeficientes de um polinómio P para obter os $m + 1$ coeficientes do polinómio $Q(x) = P(x + u)$.
O algoritmo descrito por Budan, aparecido pela primeira vez em 1807, mostra como calcular os coeficientes de um polinómio $Q(x) = P(x + 1)$, a partir dos $m + 1$ coeficientes de um dado polinómio P, usando apenas adições. Budam generalizou o método num trabalho apresentado em 1813.
Horner, que apresentou o seu algoritmo em 1819, desconhecia o trabalho de Ruffini mas não o de Budan de 1807. O processo de cálculo que desenvolveu, determinando simultâneamente os valores de $P(a)$ e $P'(a)$ com significativa redução no número de operações a efectuar, facilita consideràvelmente a aplicação do método de Newton quando aplicado a equações algébricas. [Chabert et al., 230-231]

3.8.1 Localização dos zeros

O Teorema Fundadamental da Álgebra [fazer referência exterior] diz-nos que um polinómio de grau n tem exactamente n zeros, reais e complexos, contando a multiplicidade.
Verifica-se que os zeros complexos de um polinómio p de coeficientes reais ocorrem em pares conjugados. Isto é, se $x^* = a + bi$, $b \neq 0$, é um zero de p então $\overline{x}^* = a - bi$ é também um zero de p.
Podemos estimar o número de zeros reais positivos e negativos de um polinómio p recorrendo à regra de Descartes.
Regra dos sinais Descartes. O número N_+ de zeros reais positivos de um polinómio real p não excede o número V de variações de sinal dos seus coeficientes não-nulos e o valor $V - N_+$ é par.
Consideremos o polinómio $q(x) = p(-x)$. Se x^* for um zero de p, então

$$0 = p(x^*) = p(-(-x^*)) = q(-x^*)$$

pelo que concluímos que $-x^*$ é um zero de q. Aplicando a regra de Descartes ao polinómio q obtemos uma estimativa dos zeros negativos do polinómio p.

Exemplo 3.2 (Asaithambi,105) *Consideremos*

$$p(x) = 2x^4 + 3x^3 - x^2 - 5x - 1.$$

A sequência de sinais dos coeficientes não-nulos é $\{+, +, -, -, -\}$. *Existe apenas uma mudança de sinal pelo que* $V := 1$. *Para que* $V - N_+$ *seja par, terá de ser* $N_+ = 1$. *Ou seja,* f *tem exactamente um zero positivo.*
Escrevendo agora,

$$q(x) := p(-x) = 2x^4 - 3x^3 - x^2 + 5x - 1,$$

verificamos que a sequência de sinais dos coeficientes não-nulos é $\{+, -, -, +, -\}$. *Temos assim 3 mudanças de sinal,* $V := 3$. *Sabendo que o número* $V - N_-$ *tem de ser par, concluímos que* N_- *pode tomar os valores* $N_- = 1$ *ou* $N_- = 3$.
Concluímos que p *tem um zero real positivo e três zeros reais negativos ou um zero real positivo, um zero real negativo e dois zeros complexos conjugados.*

3.8.2 Método de Newton para equações algébricas

Consideremos a utilização do método de Newton na resolução da equação algébrica

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0, \quad a_n \neq 0. \tag{3.8.4}$$

Com essa finalidade vamos apresentar uma forma eficiente de calcular $p(\hat{x})$ e $p'(\hat{x})$.
O polinómio p pode escrever-se na seguinte forma

$$p(x) = a_0 + x\,(a_1 + x\,(a_2 + \cdots + x\,(a_{n-2} + x\,(a_n + a_{n-1})))) \tag{3.8.5}$$

conhecida por *forma de Horner*.
Com esta forma são necessárias n adições e n multiplicações, o que constitui uma nítida melhoria se compararmos com as n adições e $2n - 1$ multiplicações que são necessárias na forma (3.8.4).
Definamos agora os seguintes coeficientes:

$$\begin{cases} b_n = a_n \\ b_i = b_{i+1} x + a_i, \quad i = n-1, n-2, \ldots, 0 \end{cases}$$

Da forma de Horner (3.8.5), vemos que,

$$p(x) = b_0.$$

Introduzindo o polinómio

$$q(t) = b_n t^{n-1} + b_{n-1} t^{n-2} + \cdots + b_2 t + b_1$$

temos que,

$$\begin{aligned}(t-x)q(t) + b_0 &= (t-x)(b_n t^{n-1} + b_{n-1} t^{n-2} + \cdots + b_2 t + b_1) + b_0 \\ &= b_n t^n + (b_{n-1} - b_n x)t^{n-1} + \cdots + (b_1 - b_2 x)t + (b_0 - b_1 x) \\ &= a_n t^n + a_{n-1} t^{n-1} + \cdots + a_1 t + a_0.\end{aligned}$$

Logo,

$$p(t) = (t-x)q(t) + b_0 \tag{3.8.6}$$

onde $q(t)$ é o quociente e b_0 é o resto da divisão de $p(t)$ por $t - x$.
Se x é um zero de p, então $b_0 = 0$ e $p(t) = (t-x)q(t)$.
A relação (3.8.6) permite-nos obter p'. Com efeito, derivando em ordem a t, temos

$$p'(t) = (t-x)q'(t) + q(t) \tag{3.8.7a}$$

$$p'(x) = q(x) \tag{3.8.7b}$$

Para determinarmos p' basta assim aplicar a forma de Horner a q:

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_2	a_1	a_0
$\hat{x}$		$b_n\hat{x}$	$b_{n-1}\hat{x}$	$\cdots$	$b_3\hat{x}$	$b_2\hat{x}$	$b_1\hat{x}$
	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_2	b_1	$\boxed{b_0}$
$\hat{x}$		$c_n\hat{x}$	$c_{n-1}\hat{x}$	$\cdots$	$c_3\hat{x}$	$c_2\hat{x}$	
	c_n	c_{n-1}	c_{n-2}	$\cdots$	c_2	$\boxed{c_1}$	

3.9 Exercícios

3.1 (Neide,94) *Mostre que as seguintes equações possuem exactamente uma raíz. Determine, em cada caso, um intervalo que a contenha.*

(a) $x^2 + \ln x = 0$.

(b) $x\,e^x - 1 = 0$.

Determine essas raízes com dois algarismos exactos, usando o método da bissecção.

Chapter 4

Sistemas de equações lineares e não-lineares

4.1 Introdução

– do sistema de duas equações em duas incógnitas no secundário a sistemas de milhões de equações em milhões de incógnitas ($n \simeq 10^8$).

4.2 Sistemas triangulares

4.3 Método de Gauss

4.3.1 Algoritmo clássico

[Kress,11]
Vamos descrever o método de eliminação de Gauss para um sistema de equações lineares

$$Ax = b,$$

onde $A = (a_{ij})$ é uma dada matriz $n \times n$ com elementos reais, $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$ o vector dos termos independentes e $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ é o vector solução. Mais especificamente, o sistema de equações pode ser escrito na forma

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i, \quad j = 1, \ldots, n$$

ou seja,

$$\begin{aligned} a_{11}\, x_1 + a_{12}\, x_2 + \cdots + a_{1n}\, x_n &= b_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \cdots + a_{2n}\, x_n &= b_2 \\ \vdots \qquad & \quad \vdots \\ a_{n1}\, x_1 + a_{n2}\, x_2 + \cdots + a_{nn}\, x_n &= b_n \end{aligned}$$

A ideia em que assenta o método de eliminação de Gauss é usar a primeira equação para eliminar a primeira incógnita das $n-1$ equações seguintes, de seguida usar a nova segunda equação para eliminar

a segunda incógnita das restantes $n-2$ equações e assim sucessivamente. Deste modo, através de $n-1$ destas eliminações o sistema linear dado é transformado num sistema linear equivalente de forma triangular

$$\begin{aligned} a_{11}\,x_1 + a_{12}\,x_2 + \cdots + a_{1n}\,x_n &= b_1 \\ a_{22}\,x_2 + \cdots + a_{2n}\,x_n &= b_2 \\ \vdots \qquad & \vdots \\ a_{nn}\,x_n &= b_n \end{aligned}$$

onde $a_{ii} \neq 0$, $i = 1, 2, \ldots, n$.
Este sistema triangular pode ser resolvido recursivamente de forma a obtermos x_n a partir da última equação, depois obtermos x_{n-1} da penúltima equação e assim sucessivamente. Tal processo é designado por substituições ascendentes. Explicitamente, é definido por

$$x_n = \frac{b_n}{a_{nn}}, \qquad x_i = \frac{1}{a_{ii}}\Big(b_i - \sum_{j=i+1}^{n} a_{ij}\,x_j\Big), \qquad i = n-1,\ n-2, \ldots, 1$$

Número de operações aritméticas de Gauss

4.3.2 Escolha de pivot

Influência dos erros de arredondamento

Pivot parcial

No passo k do algoritmo de Gauss, os coeficientes da coluna k, $a_{ik}^{(k-1)}$, $i = k, \ldots, n$, são candidatos a pivot. Seja p_k um índice tal que verifique

$$|a_{p_k,k}| = \max_{k \leq i \leq n} \left|a_{ik}^{(k-1)}\right|.$$

Se $p_k \neq k$ procedemos à troca das linhas p_k e k.

Pivot global

Tomamos como candidatos a pivot no passo k do algoritmo de Gauss, todos os elementos abaixo e à direita de $a_{kk}^{(k-1)}$, ou seja, $a_{ij}^{(k-1)}$, $i, j = k, \ldots, n$. Determinam-se dois índices p_k e q_k, verificando,

$$|a_{p_k,q_k}| = \max_{k \leq i,j \leq n} \left|a_{ij}^{(k-1)}\right|.$$

Se $p_k \neq k$ trocam-se as linhas k e p_k como no caso anterior; se, além disso, $q_k \neq k$ efectuamos também a troca das colunas k e q_k.

4.4 Factorizações triangulares

4.4.1 Factorizações LU

Factorização	Características	Nome
LU	$l_{ii} = 1,\ i = 1, \ldots, n$	Doolittle
LU	$u_{ii} = 1,\ i = 1, \ldots, n$	Crout
LL^T	$U = L^T,\ l_{ii} \geq 0$	Cholesky

DADOS: A, b

CICLO: Para $k = 1, \ldots, n-1$ fazer

  CICLO: Para $i = k+1, \ldots, n$ fazer

    $l_{ik} := \dfrac{a_{ik}}{a_{kk}}$

    CICLO: Para $j = k+1, \ldots, n$ fazer

      $a_{ij} := a_{ij} - l_{ik}\, a_{kj}$

      $b_i := b_i - l_{ik} b_k$

    FIM DO CICLO j

  FIM DO CICLO i

FIM DO CICLO k

CICLO: Para $k = n, n-1, \ldots, 1$ fazer

  $x_k := b_k$

  CICLO: Para $j = k+1, \ldots, n$ fazer

    $x_k := x_k - a_{kj}\, x_j$

  FIM DO CICLO j

  $x_k := \dfrac{x_k}{a_{kk}}$

FIM DO CICLO k

SAÍDA: x

Algoritmo 4.1: *Método de eliminação de Gauss*

Factorização de Doolittle

Factorização de Crout

Factorização de Cholesky

4.4.2 Factorização de Doolittle e método de Gauss

Resolução de sistemas de equaçõe baseada na factorização LU

Resolução de sistemas com a mesma matriz

Cálculo do determinante de A

Cálculo da matriz inversa

4.5 Refinamento iterativo da solução

Refinamento iterativo é uma técnica para melhorar a precisão de uma solução fornecida por um método directo. [Quarteroni et al., 111]

4.6 Métodos iterativos

Os métodos iterativos podem tornar-se competitivos com os métodos directos desde que o número de iterações necessárias para convergir (de acordo com uma tolerância dada) ou é independente de n ou 'scales' sublinearmente em ordem a n. [Quarteroni et al., 123]

4.6.1 Método de Jacobi

4.6.2 Método de Gauss-Seidel

Convergência dos dois métodos iterativos cf. EF

4.7 Sistemas de equações não-lineares

4.7.1 Método de Newton

4.7.2 Métodos lineares generalizados

Método de Jacobi generalizado

Método de Gauss-Seidel generalizado

Chapter 5

Interpolação polinomial e aproximação de funções

Interpolação refere-se a determinar uma função que represente exactamente um conjunto de valores. O tipo mais elementar de interpolação consiste em ajustar um polinómio a um conjunto de pontos representando os dados.

Teorema 5.1 (Teorema da aproximação de Weierstrass,1886) *Seja f uma função definida e contínua em $[a,b]$. Dado $\varepsilon > 0$, existe um polinómio p definido em $[a,b]$, com a propriedade*

$$|f(x) - p(x)| < \varepsilon, \quad \textit{para todo o } x \in [a,b].$$

Basicamente, o que o teorema de Weierstrass nos diz é que uma função contínua pode ser aproximada tanto quanto se queira num intervalo fechado por um polinómio.

5.1 Polinómio interpolador

Definição 5.1 *Sejam $x_0, x_1, \ldots, x_n$ elementos pertencentes ao intervalo $[a,b]$ e $f(x_0), f(x_1), \ldots, f(x_n)$ os valores correspondentes de uma função real f definida em $[a,b]$. Um polinómio de grau n*

$$p_n(x) = a_n x^n + \cdots + a_2 x^2 + a_1 x + a_0$$

verificando as condições

$$p_n(x_i) = f(x_i), \quad i = 0, 1, \ldots, n \tag{5.1.1}$$

chama-se polinómio interpolador de f nos nós de interpolação $x_0, x_1, \ldots, x_n$.

Podemos usar directamente esta definição para determinar o polinómio interpolador de uma função f cujos valores são conhecidos nos $n+1$ nós $x_0, x_1, \ldots, x_n$ num intervalo $[a,b]$, aplicando o método dos coeficientes indeterminados: avaliando $p_n(x)$ em cada um dos nós, virá

Obtemos, assim, um sistema de $n+1$ equações lineares em $n+1$ incógnitas, representado matricialmente por,

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) \end{pmatrix} \tag{5.1.2}$$

A matriz dos coeficientes do sistema de equações (5.1.2) é denominada *matriz de Vandermonde* e é regular caso os x_i sejam distintos.
Contudo, este processo não é eficiente para sistemas de grande dimensão.

5.2 Fórmula de Lagrange

Seja f uma função definida num intervalo $[a,b]$ e $x_0, x_1, \ldots, x_n$ um conjunto de $n+1$ pontos distintos em $[a,b]$.
Pretendemos determinar um polinómio p_n de grau n interpolador de f nos nós $x_0, x_1, \ldots, x_n$.

Definição 5.2 *Os polinómios*

$$\begin{aligned} L_i(x) &= \frac{(x-x_0)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{j-1})(x_i-x_{j+1})\cdots(x_i-x_n)} \\ &= \prod_{j=0, j\neq i}^{n} \frac{x-x_j}{x_i-x_j}, \quad i = 0, 1, \ldots, n \end{aligned}$$

chamam-se polinómios de Lagrange.

Verificam a seguinte propriedade:

$$L_i(x_k) = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{se } i \neq k \end{cases} \qquad \text{para } i = 0, 1, \ldots, n, \quad k = 0, 1, \ldots, n.$$

Deste modo, o polinómio de grau n

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\, L_i(x), \tag{5.2.3}$$

ao verificar as condições (5.1.1), é um polinómio interpolador de f nos nós $x_0, x_1, \ldots, x_n$. A expressão no segundo membro de (5.2.3) é denominada *fórmula de Lagrange*.

5.3 Erro de interpolação

O polinómio interpolador p_n coincide com f nos nós de interpolação. Em qualquer outro ponto, é cometido um erro quando pretendemos aproximar f po p_n.
O erro de interpolação da função f pelo polinómio p_n é a função R_n definida por

$$R_n(x) = f(x) - p_n(x), \quad x \in [a,b].$$

Teorema 5.2 *Seja p_n o polinómio de grau n interpolador de uma função f nos nós distintos $x_0, x_1, \ldots, x_n$. Se f tem derivadas contínuas até à ordem $n+1$ em $[a,b]$, então para qualquer $x \in [a,b]$ existe um número $\xi \in [\min\{x_0, x_1, \ldots, x_n, x\}, \max\{x_0, x_1, \ldots, x_n\}]$ tal que,*

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i). \tag{5.3.4}$$

5.4 Diferenças divididas

Definição 5.3 *A diferença dividida de primeirra ordem de f relativa aos argumentos x_i e x_{i+1}, $i = 0, 1, \ldots, n-1$, representa-se por $f[x_i, x_{i+1}]$, ou $f_{i,i+1}$, e calcula-se como*

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

As diferenças divididas de f de ordem superior são definidas por recorrência. Assim, a *diferença dividida* de segunda ordem de f relativa aos argumentos x_i, x_{i+1} e x_{i+2} define-se por,

$$f[x_i, x_{i+1}, x_{i+2}] = f_{i,i+2} = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+1} - x_i}.$$

A *diferença dividida* de ordem m de f relativa aos argumentos $x_i, x_{i+1}, \ldots, x_{i+m}$ define-se por,

$$f[x_i, x_{i+1}, \ldots, x_{i+m}] = f_{i,i+m} = \frac{f[x_{i+1}, x_{i+m}] - f[x_i, x_{i+m-1}]}{x_{i+1} - x_i}.$$

Verifica-se a seguinte propriedade,

$$f[x_i, x_{i+1}, \ldots, x_{i+m}] = f_{i,i+m} = \sum_{j=0}^{m} \frac{f(x_{i+j})}{\prod\limits_{\substack{k=0 \\ k \neq j}}^{m} (x_{i+j} - x_{i+k})} \tag{5.4.5}$$

Exercício 5.1 *Verifique a propriedade (5.4.5) para o caso $i = 0$, $m = 1$.*

Outra propriedade das diferenças finitas é que o seu valor não depende da ordem pela qual os seus argumentos são escritos em $f[x_i, x_{i+1}, \ldots, x_{i+m}]$.

x	$f(x)$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$	$f[\cdot,\cdot,\cdot,\cdot]$	$\ldots$	$f[\cdot,\ldots,\cdot]$
x_0	f_0					
		$f_{0,1}$				
x_1	f_1		$f_{0,2}$			
		$f_{1,2}$		$f_{0,3}$		
x_2	f_2		$f_{1,3}$		$\ldots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$f_{0,n}$
x_{n-2}	f_{n-2}		$f_{n-3,n}$		$\ldots$	
		$f_{n-2,n-1}$		$f_{n-3,n}$		
x_{n-1}	f_{n-1}		$f_{n-2,n}$			
		$f_{n-1,n}$				
x_n	f_n					

5.5 FORMA DE NEWTON PARA O POLINÓMIO INTERPOLADOR

A adição de um novo ponto ao método de Lagrange requer um acréscimo substancial de cálculo: cada polinómio de Lagrange necessita de ser multiplicado, no numerador e no denominador por um factor e um novo polinómio de Lagrange, correspondente ao ponto adicionado, deverá ser acrescentado.
A fórmula das diferenças divididas evita a necessidade de efectuar novamente os cálculos: basta introduzir um termo para obtermos um polinómio interpolador de grau superior.
Deste modo, obtemos a denominada forma de Newton para um polinómio p_n de grau n,

$$\begin{aligned} p_n(x) =& f(x_0) + f[x_0, x_1]\,(x - x_0) + f[x_0, x_1, x_2]\,(x - x_0)(x - x_1) \\ &+ \cdots + f[x_0, x_1, \ldots, x_n]\,(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}$$

nos pontos $x_0, x_1, \ldots, x_{n-1}$.

5.6 ERRO DE INTERPOLAÇÃO E DIFERENÇAS DIVIDIDAS

A expressão (5.3.4) requer o conhecimento da derivada de ordem $n+1$ de f e tem, por esse motivo, interesse limitado na prática. Podemos contudo determinar o erro de interpolação em termos da diferença dividida de ordem $n+1$ da função f.
Suponhamos que o ponto ξ no qual desejamos calcular o erro é adicionado à tabela das diferenças finitas para $x_0, x_1, \ldots, x_n$. Então o polinómio p_{n+1} que coincide com f em $x_0, x_1, \ldots, x_n$ e ξ é dado por

$$p_{n+1}(x) = p_n(x) + f[x_0, x_1, \ldots, x_n, \xi]\,(x - x_0)(x - x_1)\ldots(x - x_n)$$

Contudo, como $p_{n+1}(\xi) = f(\xi)$, obtemos

$$f(\xi) - p_n(\xi) = f[x_0, x_1, \ldots, x_n, \xi]\,(\xi - x_0)(\xi - x_1)\ldots(\xi - x_n).$$

Como ξ é um ponto arbitrário no intervalo $[a, b]$ provámos o resultado seguinte.

TEOREMA 5.3 *Seja p_n o polinómio de grau n interpolador de uma função f definida em $[a, b]$ nos nós distintos $x_0, x_1, \ldots, x_n \in [a, b]$. Então para qualquer $x \in [a, b]$, tem-se*

$$R_n(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i).$$

EXERCÍCIO 5.2 *(a) Construa a tabela das diferenças divididas para os seguintes valores tabelados*

x	0.1	0.5	1.0	0.2	0.6
$f(x)$	0.100167	0.521095	1.175201	0.201336	0.036654

(b) Determine o polinómio interpolador.

(c) Use o polinómio da alínea anterior para obter uma aproximação do valor de $f(0.3)$. Obtenha uma majoração para o erro desta aproximação.

(d) Acrescente o ponto $f(0.8) = 0.888106$ ao final da tabela e determine o polinómio interpolador através de diferenças divididas. Determine novamente uma aproximação de $f(0.3)$.

$$
\begin{array}{cc|ccccc}
x_0 & f_0 & & & & & \\
 & & f_{0,1} & & & & \\
x_1 & f_1 & & f_{0,2} & & & \\
 & & f_{1,2} & & f_{0,3} & & \\
x_2 & f_2 & & f_{1,3} & & f_{1,n} & \\
 & & f_{n-2,n-1} & & f_{n-3,n} & & \\
x_4 & f_{n-1} & & f_{n-2,n} & & & \\
 & & f_{n-1,n} & & & & \\
x_n & f_n & & & & &
\end{array}
$$

5.7 Comportamento do erro de interpolação. Polinómios de Chebyshev

Até este momento assumimos sempre que os valores da função a interpolar eram conhecidos em certos pontos. Suponhamos agora que tínhamos a liberdade de escolher $n+1$ pontos num certo intervalo. Como deveríamos efectuar tal escolha?
Como estamos a interpolar uma função por um polinómio de grau n podemos escolher esses $n+1$ pontos de forma que o $\max_{a\leq x\leq b}|f(x)-p_n(x)|$ seja mínimo. Visto que desconhecemos $f^{n+1}(\xi)$ em (5.3.4) o melhor que podemos fazer é escolher $\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_n$ de modo que minimize $\max_{a\leq x\leq b}|\prod_{i=0}^{n}(x-\hat{x}_i)|$. Este problema pode ser solucionado recorrendo aos denominados *polinómios de Chebyshev*.

5.7.1 Polinómios de Chebyshev

Os polinómios de Chebyshev de grau $m = 0, 1, 2, \ldots$ definem-se por recorrência do seguinte modo:

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_m(x) &= 2x\,T_{m-1}(x) - T_{m-2}(x), \quad \text{para } x \geq 2
\end{aligned}
$$

5.7.2 Propriedades dos polinómios de Chebyshev

5.8 Splines

5.8.1 Splines de grau um (polinómios lineares segmentados)

5.8.2 Splines cúbicos

5.9 Método dos mínimos quadrados

5.10 Aproximação trigonométrica

5.10.1 Interpolação de funções por polinómios trigonométricos

5.10.2 Aproximação de funções por polinómios trigonométricos

Chapter 6

Diferenciação e integração numérica

6.1 Exercícios

1. [M-F,374] O comprimento de arco de uma curva $y = f(x)$ num intervalo $a \leq x \leq b$ é dado por

$$L := \int_a^b \sqrt{1 + (f'(x))^2}\, dx.$$

   (i) Aproxime o comprimento de arco de cada uma das funções usando a regra do trapézio composta com $n = 10$.

   (ii) Aproxime o comprimento de arco de cada uma das funções usando a regra de Simpson composta com $n = 5$.

   (a) $f(x) = x^3$, para $0 \leq x \leq 1$.
   (b) $f(x) = \sin(x)$, para $0 \leq x \leq \pi/4$.
   (c) $f(x) = e^{-x}$, para $0 \leq x \leq 1$.

2. [M-F,374] O comprimento de arco de uma curva $y = f(x)$ num intervalo $a \leq x \leq b$ é dado por

$$L := \int_a^b \sqrt{1 + (f'(x))^2}\, dx.$$

   (i) Aproxime o comprimento de arco de cada uma das funções usando a regra do trapézio composta com $n = 10$.

   (ii) Aproxime o comprimento de arco de cada uma das funções usando a regra de Simpson composta com $n = 5$.

   (a) $f(x) = x^3$, para $0 \leq x \leq 1$.
   (b) $f(x) = \sin(x)$, para $0 \leq x \leq \pi/4$.
   (c) $f(x) = e^{-x}$, para $0 \leq x \leq 1$.

Chapter 7

Optimização Numérica

Vamos abordar neste capítulo a minimização de funções de uma ou mais variáveis.

A formulação básica de um problema deste tipo escreve-se da seguinte maneira: dada $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, denominada *função objectivo*,

$$\text{minimizar } f(\mathbf{x}) \text{ em } \mathbb{R}^n \tag{7.0.1}$$

e é chamado um problema de *optimização sem restrições*. O ponto $\mathbf{x}^*$, solução de (7.0.1), diz-se um *minimizador global* de f.

Um exemplo típico consiste em determinar a localização optimal de n recursos, $x_1, x_2, \ldots, x_n$, em competição entre si e regidos por uma lei específica. Geralmente, estes recursos não são ilimitados; esta circunstância, dum ponto de vista matemático, consiste em exigir que o minimizador da função objectivo pertença a um subconjunto $\Omega \subset \mathbb{R}^n$ e, eventualmente, que alguma restrições traduzidas através de igualdades ou desigualdades devam ser satisfeitas.

Quando tais restrições existem o problema de optimização diz-se *condicionado* e pode ser formulado do seguinte modo: dada uma função objectivo f,

$$\text{minimizar } f(\mathbf{x}) \text{ em } \Omega \subset \mathbb{R}^n.$$

7.1 Optimização sem restrições de funções de uma variável

Para funções diferenciáveis um método elementar para determinar os extremos de uma função é localizar os pontos onde a derivada se anula. Os método numéricos para localização dos zeros podem ser adaptados para esta situação. Comecemos por recordar algumas definições e teoremas.

Definição 7.1 *Uma função f tem um mínimo local no ponto x^* se existir um intervalo aberto I contendo x^* tal que $f(x^*) \leq f(x)$ para todo o x pertencente a I. O ponto x^* diz-se um minimizador de f. Analogamente, diz-se que f tem um máximo local em x^* se $f(x) \leq f(x^*)$ para todo o $x \in I$. Caso f tenha um mínimo local ou um máximo local em x^* diremos que f tem um extremo local em x^*.*

Suponhamos que f está definida no intervalo I.

(i) Se $x_1 < x_2$ implica que $f(x_1) < f(x_2)$ para todo x_1, $x_2 \in I$, então f é uma *função crescente* em I.

(ii) Se $x_1 < x_2$ implica que $f(x_1) > f(x_2)$ para todo x_1, $x_2 \in I$, então f é uma *função decrescente* em I.

DEFINIÇÃO 7.2 *Uma função f diz-se unimodal em $I = [a, b]$ se existe um único número $x^* \in I$ tal que*

$$f \text{ é decrescente em } [a, x^*]$$
$$f \text{ é crescente em } [x^*, b].$$

TEOREMA 7.1 *Seja f uma função contínua em $I = [a, b]$ e diferenciável em (a, b).*

(i) Se $f'(x) > 0$ para todo o $x \in (a, b)$, então f é crescente em I.

(ii) Se $f'(x) < 0$ para todo o $x \in (a, b)$, então f é decrescente em I.

TEOREMA 7.2 *Suponhamos que f está definida no intervalo $I = [a, b]$ e tem um extremo local num ponto interior $x^* \in (a, b)$. Se f é diferenciável no ponto x^*, então $f'(x^*) = 0$.*

TEOREMA 7.3 (TESTE DA PRIMEIRA DERIVADA) *Seja f uma função contínua em $[a, b]$. Suponhamos que $f'(x)$ está definida para todo o $x \in (a, b)$, excepto eventualmente em x^*.*

(i) Se $f'(x) < 0$ em (a, x^*) e $f'(x) > 0$ em (x^*, b), então $f(x^*)$ é um mínimo local.

(ii) Se $f'(x) > 0$ em (a, x^*) e $f'(x) < 0$ em (x^*, b), então $f(x^*)$ é um máximo local.

TEOREMA 7.4 (TESTE DA SEGUNDA DERIVADA) *Suponhamos que f é contínua em $[a, b]$ e que f' e f'' estão definidas em (a, b). Suponhamos ainda que $x^* \in (a, b)$ é um ponto crítico, isto é, onde $f'(x^*) = 0$.*

(i) Se $f''(x^*) > 0$, então $f(x^*)$ é um mínimo local de f.

(ii) Se $f''(x^*) < 0$, então $f(x^*)$ é um máximo local de f.

(iii) Se $f''(x^*) = 0$, então este teste é inconclusivo.

EXERCÍCIO 7.1 *Use o teste da segunda derivada para classificar os extremos locais da função $f(x) = x^3 + x^2 - x + 1$ no intervalo $[-2, 2]$.*

7.1.1 MÉTODO DA SECÇÃO DE OURO

Um outro método de encontrar o mínimo de uma função f é calcular vários valores da função em busca do mínimo. De modo a reduzir o número de valores a calcular é importante possuir uma boa estratégia para determinar em que pontos x_i calcular $f(x_i)$. Um dos métodos mais eficientes usa a razão de ouro na selecção dos pontos.

Consideremos o intervalo $[0, 1]$. Se $0.5 < r < 1$, então $0 < 1 - r < 0.5$ e o intervalo é dividido em três subintervalos $[0, 1 - r]$, $[1 - r, r]$ e $[r, 1]$. Teremos então de tomar uma decisão quanto à escolha de um dos subintervalos: $[0, r]$ ou $[1 - r, 1]$. Em seguida, o subintervalo seleccionado será, do mesmo modo, subdividido em três subintervalos.

Pretendemos escolher r de tal modo que um dos pontos antigos esteja na posição correcta como

indicado na figura. Isto implica que a proporção $(1-r):r$ seja a mesma que $r:1$. Por conseguinte, r verifica a equação $1-r=r^2$, ou seja, $r^2+r-1=0$. A solução verificando $0.5<r<1$ é $r=(\sqrt{5}-1)/2$ (denominado número de ouro).

Se a função f é estritamente convexa no intervalo $[a,b]$ podemos substituir este intervalo por um subintervalo no qual f toma o seu valor mínimo. O método usando a razão de ouro requer a escolha de dois pontos interiores $c=a+(1-r)(b-a)$ e $d=a+r(b-a)$, onde r é o número acima encontrado. Daqui resulta que $a<c<d<b$. A propriedade de f ser estritamente convexa garante que os valores $f(c)$ e $f(d)$ são inferiores ao $\max\{f(a),f(b)\}$. Teremos então dois casos a considerar ilustrados na figura ??.

Se $f(c)\leq f(d)$, o mínimo terá de ocorrer no subintervalo $[a,d]$: substituímos b por d e prosseguimos a busca no novo subintervalo. Se $f(d)<f(c)$, o mínimo deverá ocorrer em $[c,b]$ e substituímos a por c, continuando a busca neste subintervalo.

Exercício 7.2 *Determine o mínimo da função estritamente convexa $f(x)=x^2-sin(x)$ no intervalo $[0,1]$:*

(a) Resolvendo a equação $f'(x)=0$ pelo método da secante;

(b) Aplicando o método da secção de ouro. Analise os resultados obtidos.

7.1.2 Método de Fibonacci

7.2 Optimização sem restrições de funções de várias variáveis

Ao longo desta secção assumiremos que f é uma função de classe $\mathcal{C}^1$ em $\mathbb{R}^n$.
Representamos por

$$\text{grad } f(\mathbf{x})=\left(\frac{\partial f}{\partial x_1},\ldots,\frac{\partial f}{\partial x_n}\right)$$

o gradiente de f num ponto $\mathbf{x}$. Se f é uma função de classe $\mathcal{C}^2$ em $\mathbb{R}^n$, representamos por $\mathbf{H}(\mathbf{x})$ a matriz *Hessiana* de f calculada num ponto $\mathbf{x}$

$$\mathbf{H}(\mathbf{x})=\begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1\partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2\partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n\partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix}$$

Vamos considerar o caso particular $n=2$. Deste modo, o gradiente e a matriz Hessiana terão a seguinte representação

$$\text{grad } f(\mathbf{x})=\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \qquad \mathbf{H}(\mathbf{x})=\begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x\partial y} \\ \frac{\partial^2 f(\mathbf{x})}{\partial y\partial x} & \frac{\partial^2 f(\mathbf{x})}{\partial y^2} \end{bmatrix}$$

onde $\mathbf{x} = (x, y)$.
Representemos por $\mathcal{B}_R(\mathbf{x}^*) \subset \mathbb{R}^2$ o conjunto dos pontos do plano tais que a sua distância ao ponto $\mathbf{x}^* = (x^*, y^*)$ é inferior a R, isto é,

$$\mathcal{B}_R(\mathbf{x}^*) = \big\{(x, y) \in \mathbb{R}^2 \mid (x - x^*)^2 + (y - y^*)^2 < R\big\}.$$

Dada uma função $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$, pretendemos

$$\text{minimizar } f(\mathbf{x}) \text{ em } \mathbb{R}^2. \tag{7.2.2}$$

Diremos que a função f tem um mínimo local em x^* (ou que $\mathbf{x}^*$ é um minimizador local de f) se existir $R > 0$ tal que

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \text{para todo o } \mathbf{x} \in \mathcal{B}_R(\mathbf{x}^*).$$

Analogamente, a função f tem um máximo local em x^* se existir $R > 0$ tal que

$$f(\mathbf{x}) \leq f(\mathbf{x}^*), \quad \text{para todo o } \mathbf{x} \in \mathcal{B}_R(\mathbf{x}^*).$$

TEOREMA 7.5 (TESTE DAS DERIVADAS DE SEGUNDA ORDEM) *Seja x^* um ponto crítico de f, isto é,*

$$grad f(\mathbf{x}) = 0 \quad \Longleftrightarrow \quad \frac{\partial f}{\partial x}(\mathbf{x}) = \frac{\partial f}{\partial y}(\mathbf{x}) = 0$$

Suponhamos que f e as suas derivadas de primeira e segunda ordem são contínuas em $\mathcal{B}_R(\mathbf{x}^*)$. Então,

(i) Se $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2}\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial y^2} - \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x \partial y} > 0$ *e* $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2} > 0$, *então $f(\mathbf{x}^*)$ é um mínimo local de f;*

(ii) Se $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2}\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial y^2} - \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x \partial y} > 0$ *e* $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2} < 0$, *então $f(\mathbf{x}^*)$ é um máximo local de f;*

(iii) Se $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2}\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial y^2} - \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x \partial y} < 0$, *então $f(\mathbf{x})$ não tem extremo local em $\mathbf{x}^*$.*

(iv) Se $\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x^2}\dfrac{\partial^2 f(\mathbf{x}^*)}{\partial y^2} - \dfrac{\partial^2 f(\mathbf{x}^*)}{\partial x \partial y} = 0$, *então o teste é inconclusivo.*

EXERCÍCIO 7.3 *Determine o mínimo da função $f(x, y) = x^2 - 4x + y^2 - y - xy$.*

Vamos abordar em seguida métodos que permitem resolver o problema (7.2.2) exigindo apenas que a função f seja contínua. Estes são designados por *métodos directos.* Noutra secção trataremos os chamados *métodos de descida* que envolvem também o cálculo das derivadas de f e têm, em geral, melhores propriedades de convergência.

Os métodos directos são empregues quando f não é diferenciável ou quando o cálculo das derivadas é particularmente difícil. Podem, por outro lado, ser utilizados para encontrar uma solução aproximada, que será tomada como condição inicial para um método de descida.

7.2.1 Método de Hooke-Jeeves

Procuramos o minimizador de f partindo de um ponto $\mathbf{x}^{(0)}$ dado e exigimos que o erro do residual seja inferior a uma certa tolerância fixada ε. O método de Hooke e Jeeves calcula um novo ponto $\mathbf{x}^{(1)}$ usando os valores de f calculados em pontos adequados segundo as direcções determinadas pelas coordenadas ortogonais em torno de $\mathbf{x}^{(0)}$. O método consiste em dois passos: um passo de *exploração* e um passo de *avanço*.

O passo de exploração começa por calcular $f(\mathbf{x}^{(0)} + h_1\mathbf{e}_1)$ onde $\mathbf{e}_1 = (1, 0)$, o primeiro vector da base canónica de $\mathbb{R}^2$ e h_1 é um número real positivo escolhido adequadamente.

Se $f(\mathbf{x}^{(0)} + h_1\mathbf{e}_1) < f(\mathbf{x}^{(0)})$ então deslocamos o ponto inicial para $\mathbf{x}^{(0)} + h_1\mathbf{e}_1$, a partir do qual procedemos de forma idêntica relativamente ao ponto $\mathbf{x}^{(0)} + h_1\mathbf{e}_1 + h_2\mathbf{e}_2$ onde $\mathbf{e}_2 = (0, 1)$ e h_2 é um número real positivo.

Se, pelo contrário, $f(\mathbf{x}^{(0)} + h_1\mathbf{e}_1) \geq f(\mathbf{x}^{(0)})$ efectuamos uma pesquisa análoga em $\mathbf{x}^{(0)} - h_1\mathbf{e}_1$. Caso tenhamos agora $f(\mathbf{x}^{(0)} - h_1\mathbf{e}_1) < f(\mathbf{x}^{(0)})$ passamos a investigar o comportamento de f na direcção $\mathbf{e}_2$ partindo deste novo ponto. No caso de insucesso, o método passa directamente a examinar a direcção $\mathbf{e}_2$ mantendo $\mathbf{x}^{(0)}$ como o ponto de partida para o passo de exploração.

Para obtermos uma certa precisão, os comprimentos dos passos h_i devem de ser escolhidos de tal modo que as quantidades

$$\left| f(\mathbf{x}^{(0)} \pm h_j\mathbf{e}_j) - f(\mathbf{x}^{(0)}) \right|, \qquad j = 1, 2$$

tenham tamanhos comparáveis.

O passo de exploração termina assim que as 2 coordenadas cartesianas são examinadas. Deste modo, o método gera um novo ponto $\mathbf{y}^{(0)}$ ao fim de, no máximo, 5 cálculos da função. Apenas duas possibilidades podem suceder:

1. $\mathbf{y}^{(0)} = \mathbf{x}^{(0)}$. Neste caso, se $\max_{i=1,2}\{h_i\} \leq \varepsilon$ o método termina e encontra a solução aproximada $\mathbf{x}^{(0)}$. Caso contrário, o comprimento dos passos de exploração h_i é reduzido a metade e um novo passo de exploração é executado partindo de $\mathbf{x}^{(0)}$;

2. $\mathbf{y}^{(0)} \neq \mathbf{x}^{(0)}$. Se $\max_{i=1,2}\{h_i\} < \varepsilon$, então o método termina dando $\mathbf{y}^{(0)}$ como solução aproximada; caso contrário, começa o passo de avanço.

O passo de avanço consiste em deslocar $\mathbf{y}^{(0)}$ segundo a direcção $\mathbf{y}^{(0)} - \mathbf{x}^{(0)}$ (que é a direcção que registou o maior decréscimo de f durante o passo de exploração), em vez de tomar simplesmente $\mathbf{y}^{(0)}$ como o novo ponto de partida $\mathbf{x}^{(1)}$.

Este novo ponto de partida é, portanto, $2\mathbf{y}^{(0)} - \mathbf{x}^{(0)}$. A partir deste ponto será efectuada uma nova série de movimentos de exploração. Se esta exploração conduzir a um ponto $\mathbf{y}^{(1)}$ tal que $f(\mathbf{y}^{(1)}) < f(\mathbf{y}^{(0)} - \mathbf{x}^{(0)})$ então está encontrado o ponto de partida para o passo de exploração seguinte, caso contrário, o 'palpite' inicial para as explorações seguintes é dado por $\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \mathbf{x}^{(0)}$.
O método está agora pronto para recomeçar a partir do ponto $\mathbf{x}^{(1)}$ recém-calculado.

7.2.2 Método de Nelder-Mead

Nelder e Mead descobriram um método de simplex para determinar um mínimo local de uma função de várias variáveis. Para o caso de duas variáveis, um simplex é um triângulo e o método consiste

em comparar os valores da função nos vértices desse triângulo. O vértice pior, onde f toma o maior valor, é rejeitado e substituído por um novo vértice. Forma-se então um novo triângulo e repete-se a pesquisa. Assim, é construída geometricamente uma sucessão de triângulos (que podem tomar diferentes formas) para os quais os valores da função nos vértices é cada vez menor.

O método gera uma sucessão de aproximações de $\mathbf{x}^*$, partindo de $\mathbf{x}^{(k)}$ usando apenas três transformações possíveis: *reflecções* relativamente a um ponto, *expansões* e *contracções*.

Seja $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ a função a minimizar e sejam dados os vértices dum triângulo, $\{\mathbf{x}^{(k)}\}$, $k = 1, 2, 3$. A função f é calculada em cada um dos pontos $\mathbf{x}^{(k)}$.

Passo 1. Determinamos uma aproximação $\overline{\mathbf{x}}$ do minimizador $\mathbf{x}^*$ dada por

$$\overline{\mathbf{x}} = \frac{1}{3}\sum_{k=1}^{3} \mathbf{x}^{(k)}$$

e averiguamos se $\overline{\mathbf{x}}$ está suficientemente próximo de $\mathbf{x}^*$. Para tal, vamos exigir que o desvio-padrão dos valores $f(\mathbf{x}^{(1)})$, $f(\mathbf{x}^{(2)})$, $f(\mathbf{x}^{(3)})$ relativamente a

$$\overline{f} = \frac{1}{3}\sum_{k=1}^{3} f(\mathbf{x}^{(k)})$$

seja menor ou igual que uma dada tolerância ε, isto é,

$$\frac{1}{2}\sum_{k=1}^{3}\left(f(\mathbf{x}^{(k)}) - \overline{f}\right)^2 < \varepsilon. \tag{7.2.3}$$

Se a aproximação encontrada verificar (7.2.3) o algoritmo termina.

Passo 2. Renomeamos os pontos $\mathbf{x}^{(k)}$ de acordo com os valores da função: $\mathbf{x}^{(M)}$ o vértice onde a função toma o menor valor, $\mathbf{x}^{(P)}$ o vértice onde a função toma o seu maior valor e $\mathbf{x}^{(B)}$ o vértice onde a função toma o valor intermédio.
Determinamos o ponto médio $\mathbf{x}^{(m)}$ do lado que une $\mathbf{x}^{(M)}$ a $\mathbf{x}^{(B)}$, definido por

$$\mathbf{x}^{(m)} = \frac{\mathbf{x}^{(M)} + \mathbf{x}^{(B)}}{2}.$$

Não sendo $\overline{\mathbf{x}}$ uma aproximação verificando (7.2.3), o ponto $\mathbf{x}^{(P)}$ será reflectido relativamente a $\mathbf{x}^{(m)}$, ou seja, encontramos um novo ponto $\mathbf{x}^{(R)}$ dado por

$$\mathbf{x}^{(R)} = 2\mathbf{x}^{(m)} - \mathbf{x}^{(P)}.$$

Passo 3. Se $f(\mathbf{x}^{(M)}) \leq f(\mathbf{x}^{(R)}) \leq f(\mathbf{x}^{(B)})$, o ponto $\mathbf{x}^{(P)}$ é substituído por $\mathbf{x}^{(R)}$ e o algoritmo regressa ao **Passo 1**.

Passo 4. Se $f(\mathbf{x}^{(R)}) \leq f(\mathbf{x}^{(M)})$ o passo de reflecção produziu uma nova aproximação do minimizador. Isto significa que o minimizador poderá estar fora do triângulo considerado. Por conseguinte, vamos expandir o triângulo determinando um novo vértice $\mathbf{x}^{(E)}$ dado por

$$\mathbf{x}^{(E)} = 2\mathbf{x}^{(R)} + \mathbf{x}^{(m)}$$

Antes de regressarmos ao **Passo 1** duas possibilidades podem ocorrer:

(4a) Se $f(\mathbf{x}^{(E)}) < f(\mathbf{x}^{(M)})$ então $\mathbf{x}^{(P)}$ é substituído por $\mathbf{x}^{(E)}$;

(4b) Se $f(\mathbf{x}^{(E)}) \geq f(\mathbf{x}^{(M)})$ então $\mathbf{x}^{(P)}$ é substituído por $\mathbf{x}^{(R)}$ visto que $f(\mathbf{x}^{(R)}) < f(\mathbf{x}^{(M)})$.

Passo 5. Se $f(\mathbf{x}^{(R)}) > f(\mathbf{x}^{(B)})$ então o minimizador encontrar-se-á provavelmente no interior do triângulo e, assim, duas abordagens diferentes podem ser efectuadas para contrair o triângulo.

Se $f(\mathbf{x}^{(R)}) < f(\mathbf{x}^{(P)})$, a contracção gera um ponto $\mathbf{x}^{(C)}$ da forma

$$\mathbf{x}^{(C)} = 0.5\,\mathbf{x}^{(R)} + 0.5\,\mathbf{x}^{(m)};$$

caso contrário,

$$\mathbf{x}^{(C)} = 0.5\,\mathbf{x}^{(P)} + 0.5\,\mathbf{x}^{(m)}.$$

Por último, antes de regressarmos ao **Passo 1**,

(5a) Se $f(\mathbf{x}^{(C)}) < f(\mathbf{x}^{(P)})$ e $f(\mathbf{x}^{(C)}) < f(\mathbf{x}^{(R)})$ então o ponto $\mathbf{x}^{(P)}$ é substituído por $\mathbf{x}^{(C)}$;

(5b) Se $f(\mathbf{x}^{(C)}) \geq f(\mathbf{x}^{(P)})$ ou se $f(\mathbf{x}^{(C)}) \geq f(\mathbf{x}^{(R)})$ então são gerados dois novos pontos $\mathbf{x}^{(k)}$, $k = 1, 2$ dividindo por dois as distâncias dos pontos $\mathbf{x}^{(B)}$ e $\mathbf{x}^{(P)}$ a $\mathbf{x}^{(M)}$.

Exercício 7.4 *Comparar os resultados obtidos através do método de Hooke e Reeves e do método de Nelder e Mead para minimizar a função de Rosembrock*

$$f(\mathbf{x}) = 100\,(y - x^2)^2 + (1 - x)^2.$$

O ponto inicial para ambos os métodos é $\mathbf{x}^{(0)} = (-1.2, 1)$*, sendo o tamanho dos passos* $h_1 = 0.6$ *e* $h_2 = 0.5$*. A tolerância* ε *no cálculo do residual é igual a* 10^{-4}*.*

7.2.3 Métodos de descida

Nesta secção introduzimos métodos iterativos que são mais sofisticados que os examinados na secção anterior. Podemos formulá-los do seguinte modo

dado um vector inicial $\mathbf{x}^0 \in \mathbb{R}^2$, calcular para $k \geq 0$ até obter convergência

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k,$$

onde $\mathbf{d}^k$ é uma direcção escolhida adequadamente e α_k é um parâmetro positivo (designado comprimento do passo) que mede o passo segundo a direcção $\mathbf{d}^k$.

7.3 Exercícios

1. (FB422) Aplique o método da descida mais acentuada para aproximar o mínimo da seguinte função, iterando até que $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_\infty < 0.005$,

$$f(x, y) = \cos(x + y) + \sin x + \cos y.$$

Bibliography

[1] Melvin J. Maron & Robert J. Lopez, *Numerical Methods, A Practical Approach*, Wadsworth Publishing Company (1991).