

## 2 Zeros reais de uma função real

### Contents

---

<b>2.1 Isolamento das raízes</b>	<b>55</b>
<b>2.2 Método da Bissecção</b>	<b>62</b>
<b>2.3 Método da Corda</b>	<b>66</b>
<b>2.4 Método da tangente</b>	<b>70</b>
<b>2.5 Método de iteração de Picard-Peano</b>	<b>72</b>
<b>2.6 Resolução de Sistemas pelo Método de Iteração</b>	<b>77</b>

---

### Figures

---

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

---

### Tables

---

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

---

## 2 Zeros reais de uma função real

O problema de achar as raízes reais de uma dada função real contínua é um dos mais frequentes problemas de cálculo numérico em Engenharia.

### Exemplo: 22.1 Raízes

Para a equação

$$x^2 + 80x + 1 = 0$$

as raízes são, com 5 algarismos significativos,  $x_1 = -79.987$  e  $x_2 = -0.012502$ . Ora, a fórmula resolvente normal dá, na precisão da nossa máquina hipotética

$$x'_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} = \frac{-0.800 \times 10^2 - \sqrt{0.640 \times 10^4 - 0.400 \times 10^1}}{0.200 \times 10^1}$$

$$x'_1 = -0.800 \times 10^2$$

$$x'_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{-0.800 \times 10^2 + \sqrt{0.640 \times 10^4 - 0.400 \times 10^1}}{0.200 \times 10^1}$$

$$x'_2 = \pm 0.000 \times 10^{-9}$$

sendo manifesto que  $f(x'_1) = 1$  e  $f(x'_2) = 1$  mas

$$(x - x'_1) \cdot (x - x'_2) = x^2 + 80x + 1$$

Que tipo de situação é que o leitor preferiria?

Para o matemático, um zero de uma função  $y = f(x)$  é qualquer número  $\xi$  que, substituído na expressão de  $f(x)$ , produz um resultado exactamente nulo:  $f(\xi) = 0$ . Porém, em cálculo numérico, dada a finitude dos números e da sua precisão, há apenas um conjunto discreto de números que podemos ensaiar como candidatos a zeros, e habitualmente sucede que nenhum deles produz exactamente zero para o valor da função.

Porém, em cálculo numérico, o significado de uma raiz é extremamente ambíguo: veja-se (no capítulo 1), por exemplo, o que se passa com o cálculo das formas desenvolvidas do polinómio do 4º grau  $(5x - 14)^2 \cdot (x^2 + 3)$  nas vizinhanças do zero teórico  $x = 2.8$ .

Usando o **Maxima**<sup>1</sup>:

#### wxMaxima: Significado de uma raiz,

Para ilustrar os problemas do cálculo desta expressão, foi necessário organizar o cálculo do **Maxima** de maneira a evitar algumas das suas optimizações: reorganização das expressões, recorrendo à decomposição em funções, gráfico a partir de pontos discretos. Note-se que a solução numérica proposta pelo **Maxima**, um método intervalar baseado na mudança de sinais em pontos da função, só é possível porque perto da raiz da função surgem valores negativos espúrios; como é facilmente verificável, a função é sempre

<sup>1</sup>(ver o ficheiro [ZEROS-2-2](#))

### Significado de uma raiz (cont.)

positiva!

(% i1)  $f(x):=(5*x-14)^2*(x^2+3);$

(% o1)  $f(x) := (5x - 14)^2 (x^2 + 3)$

(% i2)  $f1(x):=25*x^4;$

(% o2)  $f1(x) := 25x^4$

(% i3)  $f2(x):=271*x^2;$

(% o3)  $f2(x) := 271x^2$

(% i4)  $f3(x):= 588-140*x^3;$

(% o4)  $f3(x) := 588 - 140x^3$

(% i5)  $f4(x):= -420*x;$

(% o5)  $f4(x) := (-420)x$

(% i6) "Cálculo numérico das raízes, a partir de vários intervalos."

(% i7)  $\text{find\_root}('f1(x)+f2(x)+f3(x)+f4(x), x, 2.799999964, 2.80001);$

(% o7) 2.799999964

(% i8)  $\text{find\_root}('f1(x)+f2(x)+f3(x)+f4(x), x, 2.79, 2.800000029);$

(% o8) 2.799999971779375

(% i9)  $\text{find\_root}('f1(x)+f2(x)+f3(x)+f4(x), x, 2.70, 2.800000029);$

(% o9) 2.799999981316271

(% i10) "Construção das listas de pontos para o gráfico."

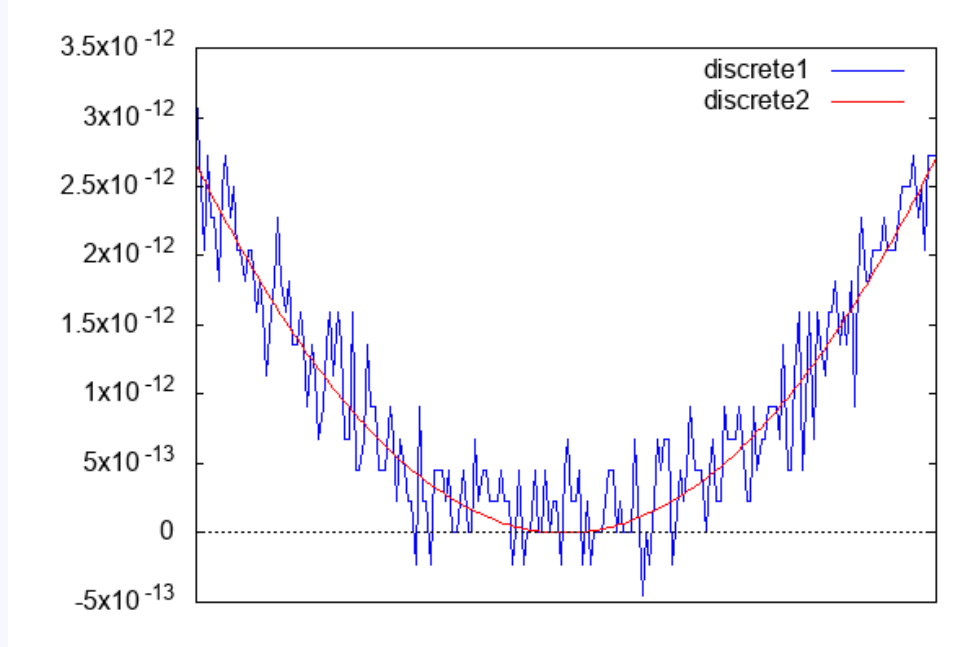
(% i11)  $xx:\text{makelist}(x/1000000000+2.799999,x,1,200),\text{numer}\$$

(% i12)  $yy:\text{makelist}(f1(x)+f2(x)+f3(x)+f4(x),x,xx),\text{numer}\$$

(% i13)  $zz:\text{makelist}(f(x),x,xx),\text{numer}\$$

Significado de uma raiz (cont.)

```
(% i14) wxplot2d([[discrete,xx,yy],[discrete,xx,zz]],
[plot_format, gnuplot],
[gnuplot_preamble, "set zeroaxis; unset xtics"])$
(% t14)
```



Assim sendo, temos o direito (e o dever) de nos interrogar: o que pretendemos dos zeros de uma função:

- que sejam números  $\xi_i$  tão próximos quanto possível do valor teórico?
- ou, pelo contrário, que nos pontos  $\xi_i$  o valor calculado de  $|f(\xi_i)|$  seja tão pequeno quanto possível?

Aquilo que pretendemos depende, evidentemente, do contexto de cada problema particular, mas é evidente que a pergunta não é inocente, até porque a estratégia do cálculo numérico depende fortemente da resposta; com efeito, no primeiro caso estaremos perante um problema de *resolução de equações*, enquanto no segundo o problema será de *optimização*. Seja como for, uma consideração cuidadosa dos casos concretos que se nos deparam na prática mostrará claramente que são mais frequentes os do segundo que os do primeiro tipo, ao contrário do que poderíamos pensar e do que correntemente se pratica.

Quanto à técnica a utilizar para calcular uma raiz de uma equação (ou, o que é o mesmo, o zero de uma função), teremos que distinguir o caso de se dispor de uma solução por via analítica, caso em que, porque a estrutura do algoritmo está, em termos genéricos, fixada, o problema se reduz ao cálculo de uma expressão, como o é o caso da equação do 2º grau; diz-se, neste caso, que se dispõe de um *método directo* ou *cerrado*, isto é, de um processo que envolve apenas a

execução finita e não-iterativa de operações aritméticas e lógicas. Embora um método directo possa ter alternativas de percurso (como acabámos de ver para a equação do 2º grau), eventualmente a escolher com base nos próprios valores obtidos durante o cálculo, distingue-se por cada passo ser executado, em geral, apenas uma vez e o resultado ser exacto, a menos dos erros introduzidos pela representação dos números no processo de cálculo utilizado. Como se sabe, os casos de resolubilidade analítica são em número muito reduzido, embora sejam frequentes nas mais simples das situações correntes:

- das equações algébricas, apenas têm solução analítica geral as de grau inferior ao quinto;
- das equações racionais, apenas aquelas que possam reduzir-se a equações algébricas resolúveis (e, mesmo assim, com os problemas marginais da possível introdução de raízes estranhas pela multiplicação da equação por expressões que contêm a incógnita);
- para as equações irracionais e transcendentais não existem soluções gerais e são raros os casos particulares resolúveis analiticamente.

No caso de não estar disponível uma solução analítica em forma cerrada, diremos que o problema se reduz, na generalidade, a procurar um par de números,  $x_1, x_2$  suficientemente próximos um do outro e entre os quais se encontre uma e uma só raiz da equação, o que exige um *método indirecto* ou *aberto*. O problema geral da resolução numérica indirecta deveria, portanto, comportar duas partes distintas:

1. - encontrar um intervalo que enquadre uma e uma só raiz;
2. - reduzir esse intervalo a dimensões compatíveis com a nossa necessidade (e possibilidade) de precisão.

Na prática, dados os efeitos dos arredondamentos que inevitavelmente ocorrem no cálculo da função, não teremos sequer, em geral, uma única mudança de sinal em torno do zero verdadeiro, mas eventualmente, uma pequena sucessão de números consecutivos que dão uma sucessão de mudanças de sinal nos valores calculados numericamente (reveja-se o caso da função do 4º grau do Capítulo anterior). Embora, em princípio, uma tal situação possa confundir-se com uma sucessão de zeros muito próximos, na prática não existirão dificuldades especiais, desde que tenhamos uma razoável ideia da ordem de grandeza dos erros numéricos cometidos no cálculo numérico da função. Esta consideração mostra desde já como foram importantes o tempo e o esforço que no Capítulo anterior consagramos ao problema dos erros de arredondamento, dado que eles são inevitáveis nos processos de cálculo numérico de resolução de equações na medida em que o cancelamento exacto de termos positivos e negativos faz parte da própria natureza do problema.

## 2.1 Isolamento das raízes

O seguinte teorema bem conhecido dá uma boa medida das ambiguidades e das dificuldades associadas ao problema do isolamento das raízes:

Se uma função contínua  $f(x)$  toma valores de sinais opostos nos extremos de um intervalo  $(a, b)$ , isto é, se  $f(a) \times f(b) < 0$ , então esse intervalo contém um número

## 2 Zeros reais de uma função real

ímpar de raízes da equação  $f(x) = 0$ ; se, pelo contrário, toma valores do mesmo sinal, isto é, se  $f(a) \times f(b) > 0$ , então esse intervalo contém um número par de raízes (incluindo nenhuma raiz).

No caso de, nos extremos do intervalo, a função tomar valores de sinal idêntico, não podemos, naturalmente, garantir que não existem raízes nesse intervalo (podem existir duas ou, mais geralmente, um número par). A garantia de que não existem raízes no intervalo só pode ser obtida mediante recurso a informação suplementar, como, por exemplo, a de que a função muda de sinal algures no intervalo.

### Exemplo: 22.2 Isolamento de raízes

Consideremos o problema de isolar as raízes da equação

$$f(x) = x^4 - 4x + 1 = 0$$

dado que  $f(-\infty) = +\infty > 0$  e  $f(+\infty) = +\infty > 0$ , nada podemos concluir.

Porém, se considerarmos a derivada

$$f'(x) = 4x^3 - 4$$

resulta que  $f'(1) = 0$  e, por outro lado,  $f(1) = -2 < 0$ , de modo que a equação tem, pelo menos, duas raízes, uma no intervalo  $(-\infty, 1)$ , outra no intervalo  $(1, +\infty)$ . Usemos as facilidades do **Maxima** para visualizar os comportamentos da função e da sua derivada na figura 2.1<sup>a</sup>.

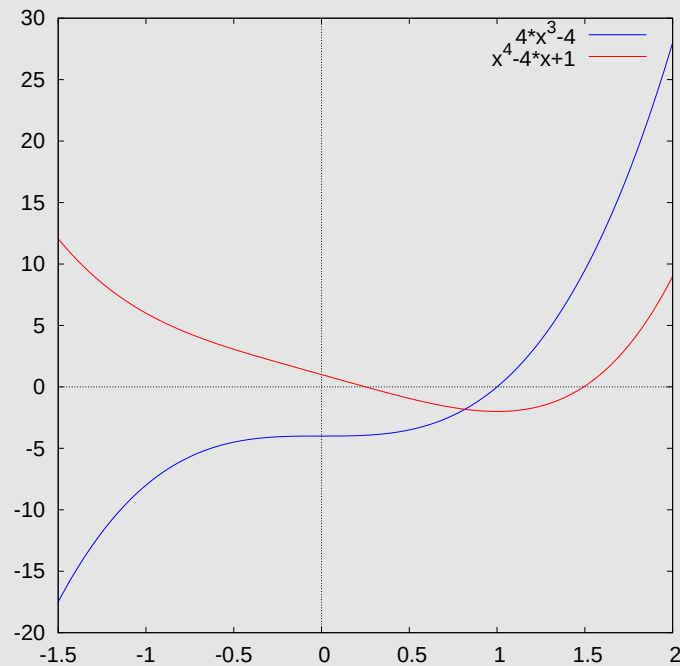


Figura 2.1: Isolamento de raízes

## Exemplo: 22.2 Isolamento de raízes (cont.)

<sup>a</sup>(ver o ficheiro **ZEROS-2-3** )**wxMaxima: Função e derivada,****(% i1)** `x^4-4*x+1;`**(% o1)**  $x^4 - 4x + 1$ **(% i2)** `diff(%, x,1);`**(% o2)**  $4x^3 - 4$ **(% i3)** `plot2d([%o1,%o2], [x,-1.5,2],  
[plot_format, gnuplot],  
[gnuplot_preamble, "set xrange [-1.5:2]; set size ratio 1; set zeroaxis;"])$`

No caso dos sinais opostos, a única coisa que o teorema garante é a existência de um número ímpar de raízes. A garantia de que existe uma raiz única só pode ser obtida mediante recurso a informação mais pormenorizada e raramente acessível, como, por exemplo, a de que a primeira derivada mantém o sinal no intervalo considerado, o que corresponde a saber que a função é monótona no intervalo.

## Exemplo: 22.3 Isolamento de raízes II

Seja a função

$$f(x) = x^4 - 5x^2 + 1$$

tal que  $f(-\infty) = +\infty > 0$  e  $f(+\infty) = +\infty > 0$ .

A sua derivada,

$$f'(x) = 4x^3 - 10x$$

tem um zero óbvio em  $x = 0$ , no qual a função vale  $f(0) = 1 > 0$ , pelo que continuamos sem saber nada sobre a eventual existência de raízes. Porém,  $f'(-\infty) = -\infty < 0$  e  $f'(+\infty) = +\infty > 0$ , de modo que a derivada tem, pelo menos, um zero no intervalo  $(-\infty, 0)$  e, portanto, também um outro no intervalo  $(0, +\infty)$ , o que completa o máximo de três que pode ter (por ser do 3º grau). Como anteriormente, utilizaremos as facilidades gráficas do **Maxima** para estudar o comportamento da função e da sua derivada na figura 2.2<sup>a</sup>:

## 2 Zeros reais de uma função real

### Exemplo: 22.3 Isolamento de raízes II (cont.)

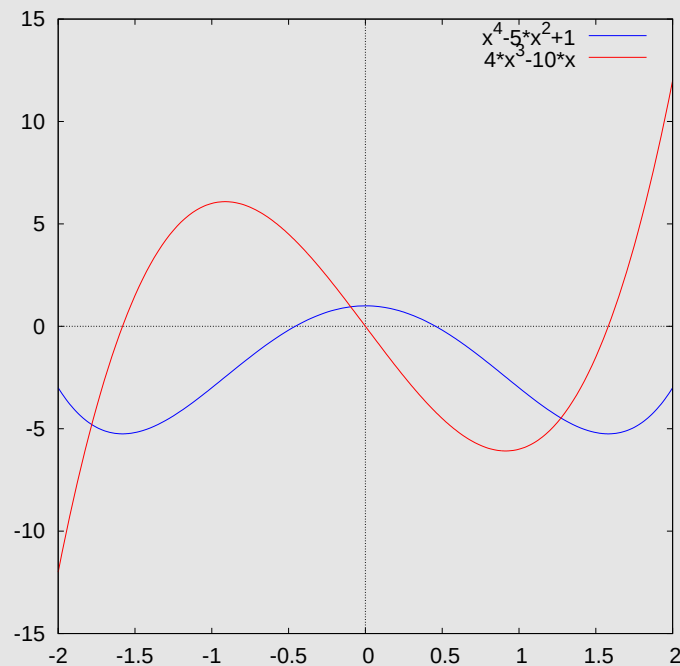


Figura 2.2: Isolamento de raízes II

<sup>a</sup>(ver o ficheiro [ZEROS-2-4](#))

### wxMaxima: Função e derivada,

```
(% i1) x^4-5*x^2+1;
```

```
(% o1) x^4 - 5x^2 + 1
```

```
(% i2) diff(%, x);
```

```
(% o2) 4x^3 - 10x
```

```
(% i3) plot2d([%o1,%o2], [x,-2,2],  
[plot_format, gnuplot],  
[gnuplot_preamble, "set xrange [-2:2]; set size ratio 1; set zeroaxis;"])$
```

Nestas condições, a única solução geral que se pode aconselhar para tratar o problema do isolamento das raízes é a de utilizar todos os recursos ao nosso alcance para obter um conhecimento suficientemente exacto do comportamento da função (e, eventualmente, das suas derivadas) no



intervalo de interesse, o qual, em muitos casos, será fixado com precisão suficiente pelas restrições impostas pela física, pela economia, ou pelo simples bom senso, no contexto do problema concreto que temos diante de nós; só na total falta de tais critérios, o intervalo terá que ser tomado como  $] -\infty, +\infty[$ .

Assim, por exemplo, se existe e é conhecida uma derivada contínua e se os zeros desta puderem ser mais facilmente calculados que os da função original, o processo de isolamento das raízes pode ser acelerado; com efeito, bastará apenas tomar os sinais da função nos zeros da derivada e nos extremos do intervalo para obter importante informação suplementar.

#### Exemplo: 22.4 Isolamento de raízes III

Determinar o número de raízes reais da equação

$$f(x) = x + e^x = 0$$

Dado que  $f'(x) = 1 + e^x > 0$

e que  $f(-\infty) = -\infty < 0$ ,  $f(+\infty) = +\infty > 0$ ,

concluimos que a equação tem uma e uma só raiz real, como mostra a figura 2.3<sup>a</sup>.

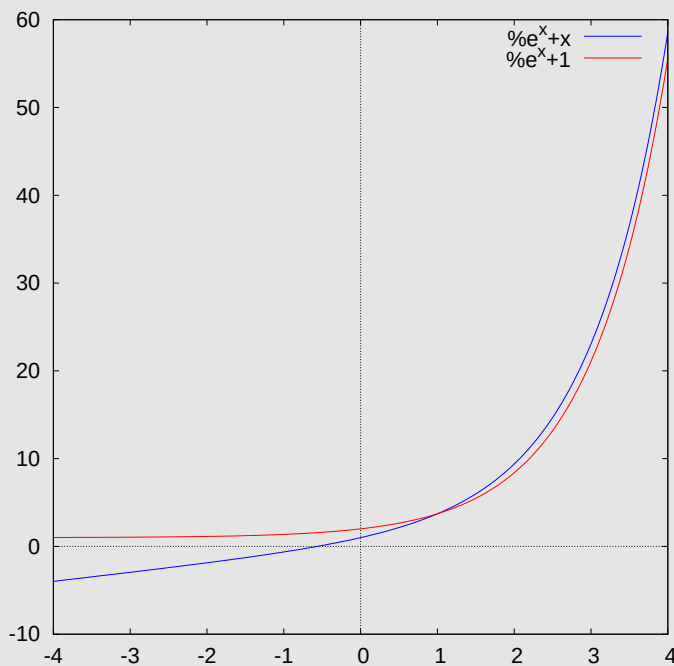


Figura 2.3: Isolamento de raízes III

<sup>a</sup>(ver o ficheiro ZEROS-2-5 )

**wxMaxima: Uma só raiz,**

(% i1) `x+%e^x;`

(% o1)  $e^x + x$

(% i2) `diff(%, x);`

(% o2)  $e^x + 1$

(% i3) `plot2d([%o1,%o2], [x,-4,4],  
[plot_format, gnuplot],  
[gnuplot_preamble, "set xrange [-4:4]; set size ratio 1; set zeroaxis;"])$`

Do mesmo modo, se o nosso problema é algébrico e de grau  $n$ , sabemos antecipadamente que tem  $n$  e só  $n$  raízes, das quais alguns pares poderão ser complexos conjugados e as restantes serão reais.

No caso das funções para as quais dispomos de uma expressão analítica, um utilitário matemático do tipo **matlab**, **maple**, **derive** ou **Maxima**, (especialmente estes últimos, dadas as suas possibilidades de cálculo simbólico) dotado de uma boa rotina de traçado de gráficos com possibilidade de *zoom* constituirá um auxiliar precioso nesta tarefa. No caso de a função ter vários parâmetros livres que modificam o seu comportamento, pode ser necessário muito trabalho, apoiado em uma boa dose de conhecimentos e intuição matemática para levar a tarefa a cabo. Pelo contrário, se a função é definida por um algoritmo não exprimível sob forma cerrada, tornar-se-á em geral necessário escrever um programa informático que permita explorar os seus valores em diferentes pontos e traçar o respectivo gráfico. Neste caso, em particular, torna-se necessário estar muito atento à possibilidade de a função apresentar descontinuidades.

Seja como for, o problema apresenta-se desde o início com a possibilidade de dois níveis de dificuldade muito distintos:

1. quando se trata de resolver apenas o problema isolado de determinação das raízes de uma equação particular, pode estudar-se cada caso particular *de per si*;
2. no caso de a resolução da equação ser apenas parte de um problema mais vasto e, portanto, o respectivo algoritmo constituir apenas uma rotina dentro de um programa, toda a questão do isolamento da raiz terá que ser resolvida na generalidade, na fase da análise do problema, isto é, antes de passar à implementação informática da solução; como se compreende, esta situação é de longe mais complexa que a anterior.

Dadas estas circunstâncias particularmente desfavoráveis, o objectivo do isolamento da raiz é, na prática corrente, substituído pelo objectivo menos ambicioso de encontrar um intervalo em que a função muda de sinal.

**Exemplo: 22.5 Isolamento de raízes IV**

Isolar os zeros da função

$$p_3(x) = x^3 - 6x + 2$$

considerando a expressão analítica como uma mera especificação de cálculo de valores e sem a utilizar para outros fins.

Começemos por observar que se trata de um polinómio do 3º grau e, portanto, contínuo e com um máximo de 3 raízes distintas. Em seguida, tabulemos (ver tabela 2.1) alguns valores encontrados, começando por  $\pm\infty$ , em seguida 0, depois  $\pm 10$  (supondo que estamos especialmente interessados em raízes pequenas) e depois por bissecções sucessivas aproximadas.

	1º passo		2º passo		3º passo		4º passo		5º passo	
	$x$	$f(x)$	$x$	$f(x)$	$x$	$f(x)$	$x$	$f(x)$	$x$	$f(x)$
valor	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
raízes	<b>1 a 3</b>		<b>0 a 2</b>		<b>0 a 2</b>		<b>0 a 2</b>		<b>0</b>	
valor			-10	-938	-10	-938	-10	-938	-10	-938
raízes			<b>1 a 3</b>		<b>0 a 2</b>		<b>0 a 2</b>		<b>0</b>	
valor					-5	-93	-5	-93	-5	-93
raízes					<b>1 a 3</b>		<b>0 a 2</b>		<b>0</b>	
valor							-3	-7	-3	-7
raízes							<b>1 a 3</b>		<b>1</b>	
valor									-1	+7
raízes									<b>0</b>	
valor	0	+2	0	+2	0	+2	0	+2 +2	0	+2
raízes	<b>0 a 2</b>		<b>0 a 2</b>		<b>0 a 2</b>		<b>0 a 2</b>		<b>1</b>	
valor									+1	-3
raízes									<b>1</b>	
valor							+3	+11	+3	+11
raízes							<b>0 a 2</b>		<b>0</b>	
valor					+5	+97	+5	+97	+5	+97
raízes					<b>0 a 2</b>		<b>0 a 2</b>		<b>0</b>	
valor					+10	+942	+10	+942	+10	+942
raízes					<b>0 a 2</b>		<b>0 a 2</b>		<b>0</b>	
valor	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Tabela 2.1: Isolamento de raízes

De repente, a separação das raízes aparece completada. A equação tem, portanto, 3 raízes reais, situadas nos intervalos  $(-3, -1)$ ,  $(0, +1)$ ,  $(+1, +3)$ . Do ponto de vista da separação, nada mais é preciso, mas é evidente que os intervalos poderiam continuar a ser reduzidos usando o mesmo método.

### Exercício 2.1

1. Determine, por meios gráficos, valores grosseiramente aproximados das raízes positivas da equação  $\tan(x) = x$
  2. Determine a dois algarismos significativos
    - a) as duas primeiras raízes positivas de  $x \cdot \sin(x) = 1$
    - b) a raiz de  $x \cdot e^x = 1$
    - c) a menor raiz positiva de  $\tan(x) = x^2 + 1$
    - d) Mostre que não há raízes reais da equação  $e^x = 1 + \ln(x)$
- 

## 2.2 Método da Bissecção

Suposto identificado um intervalo  $(a, b)$  em cujos extremos a função tem sinais opostos, o método mais simples de reduzir esse intervalo é o *método da bissecção* que constitui também, e de longe, o método mais seguro para implementar em uma rotina informática de uso geral. Neste método, a redução do intervalo é realizada mediante o cálculo da função no ponto médio do intervalo,  $f((x_1 + x_2)/2)$ . Se este valor não for nulo (porque, se fosse o caso, teríamos logo encontrado a raiz "exacta"), então ocorrerá necessariamente uma mudança de sinal em uma (e uma só) das metades do intervalo original; a redução desejada far-se-á substituindo o intervalo original por essa metade; deste modo,  $(x_1 + x_2)/2$  substitui  $x_1$  ou  $x_2$ , conforme o caso, e o processo repete-se.

---

### Algoritmo 2.1

```
DADOS a0 , b0 : f(a0) . f(b0) < 0
PARA n=0,1,2,... ATÉ ONDE FOR CONVENIENTE
    SEJA m=(an +bn)/2
    SE f(an) . f(m) < 0 SEJAM an+1 = an ; bn+1 = m
    CASO CONTRÁRIO SEJAM an+1 = m; bn+1 = bn
```

---

#### Exemplo: 22.6 Bissecção

Vamos usar o método da bissecção para melhorar a raiz da equação

$$x^4 + 2x^3 - x - 1 = 0$$

situada no intervalo  $(0, 1)$ , trabalhando a cinco algarismos significativos.

## Exemplo: 22.6 Bissecção (cont.)

a	f(a)	b	f(b)	intervalo
0,000000E+00	-1,000000E+00	1,000000E+00	1,000000E+00	1,000000E+00
5,000000E-01	-1,187500E+00	1,000000E+00	1,000000E+00	5,000000E-01
7,500000E-01	-5,898440E-01	1,000000E+00	1,000000E+00	2,500000E-01
7,500000E-01	-5,898440E-01	8,750000E-01	5,102540E-02	1,250000E-01
8,125000E-01	-3,039400E-01	8,750000E-01	5,102540E-02	6,250000E-02
8,437500E-01	-1,355730E-01	8,750000E-01	5,102540E-02	3,125000E-02
8,593750E-01	-4,461470E-02	8,750000E-01	5,102540E-02	1,562500E-02
8,593750E-01	-4,461470E-02	8,67188E-01	2,615420E-03	7,813000E-03
8,632820E-01	-2,114390E-02	8,67188E-01	2,615420E-03	3,906000E-03
8,652350E-01	-9,301190E-03	8,67188E-01	2,615420E-03	1,953000E-03
8,662120E-01	-3,349080E-03	8,67188E-01	2,615420E-03	9,760000E-04
8,667000E-01	-3,691450E-04	8,67188E-01	2,615420E-03	4,880000E-04
8,667000E-01	-3,691450E-04	8,66944E-01	1,122560E-03	2,440000E-04
8,667000E-01	-3,691450E-04	8,66822E-01	3,765620E-04	1,220000E-04
8,667000E-01	-3,691450E-04	8,66761E-01	3,672460E-06	6,100000E-05
8,667310E-01	-1,796890E-04	8,66761E-01	3,672460E-06	3,000000E-05
8,667310E-01	-1,796890E-04	8,66746E-01	-8,801060E-05	1,500000E-05
8,667390E-01	-1,307950E-04	8,66746E-01	-8,801060E-05	7,000000E-06

Tabela 2.2: Exemplo de bissecção sucessiva  
A melhor aproximação à raiz é, portanto,  $x = 0.86674$ .

Cada iteração reduz a metade o comprimento do intervalo, de modo que 10 iterações, por exemplo, reduzem-no de um factor de  $2^{10} = 1024 > 10^3$ , enquanto que 20 iterações o reduzem de  $2^{20} = 1048576 > 10^6$ .

E como se decide a paragem do processo? Existem basicamente quatro tipos de critérios distintos, aplicáveis de acordo com o objectivo em vista:

**critério de precisão absoluta**

$$|x_1 - x_2| \leq \varepsilon$$

Consiste em parar quando o intervalo que contém a raiz for menor que um dado valor (pequeno) pré-definido (sob a condição evidente de  $\varepsilon$  ser maior que a distância dos dois números representáveis consecutivos na vizinhança da raiz, sob pena de o processo não terminar);

**critério de precisão relativa**

$$|(x_1 - x_2)/x_1| \leq \varepsilon \text{ ou } |(x_1 - x_2)/x_2| \leq \varepsilon$$

Consiste em parar quando a razão entre o intervalo que contém a raiz e a própria raiz (ou um valor aproximado) for menor que um dado valor (pequeno) pré-definido (sob a condição de a raiz não ser demasiado próxima de zero, sob pena de o processo não terminar);

## 2 Zeros reais de uma função real

### critério de anulação da função

$$|f(x_1) - f(x_2)| \leq \varepsilon$$

Consiste em parar quando diferença entre o valor da função nos extremos do intervalo que contém a raiz for menor que um dado valor (pequeno) pré-definido (sob a condição de a derivada da função não ter, nas vizinhanças da raiz, valor absoluto demasiado alto, sob pena de o processo não terminar);

### critério do número de iterações

$$n = N$$

Consiste em parar quando tiverem sido feitas  $N$  iterações.

Devido às restrições aos três primeiros tipos de critérios, que nem sempre são fáceis de implementar, aconselha-se vivamente o uso do quarto. Para estabelecer um critério simples e eficiente deste último tipo podemos partir do critério do erro absoluto da raiz procurando que a máquina estabeleça o intervalo mais pequeno que pode representar. Com efeito, se, na máquina utilizada, a representação em vírgula flutuante for da forma  $m \cdot 2^p$  e  $(a, b)$  for o intervalo inicial, deverá ser

$$n \leq n_m + \log_2(b - a)$$

em que  $n_m$  é o número de bits da mantissa<sup>2</sup>.

#### Exemplo: 22.7 Iterações de Bissecção

se constatarmos que  $\log_b 1 = 0$ , qualquer que seja a base  $b$  e que  $\log_b x = \frac{\log_k x}{\log_k b}$ , podemos facilmente construir uma tabela de valores de  $N$  em função da amplitude do intervalo inicial  $b - a$ :

$(b - a)$	$N$
100	30
10	26
1	23
0,1	20
0,01	17

Se não quisermos trabalhar directamente com a representação interna (binária) da máquina, que não é acessível às linguagens de programação normais, uma boa regra é tomar

$$n \leq 3.3 \times n_M + \log_{10}(b - a)$$

em que  $n_M$  é o número de dígitos da mantissa da representação decimal externa (isto é, dos outputs).

A detecção da mudança de sinal pode ser feita pela multiplicação do valor da função nos extremos do intervalo, isto é, por  $f(x_1) \times f(x_2) < 0$ . Computacionalmente muito mais eficiente é, no entanto, a comparação directa dos sinais, recorrendo a funções que extraíam o bit de sinal da mantissa, ou mesmo a encadeamentos de condicionais de comparação de sinais ou expressões lógicas complexas

$$\text{NÃO } (f(x_1) < 0) \quad \text{E} \quad f(x_2) < 0 \quad \text{OU} \quad f(x_1) > 0 \quad \text{E} \quad f(x_2) > 0)$$

<sup>2</sup>Usando a norma IEEE para float, o número de bits da mantissa  $n_m$  é 23. O número de iterações  $N$  depende então da amplitude do intervalo inicial, e contas bastante simples mostram que, para  $b - a = 1$  então  $N = 23$ .

Terminada deste modo a iteração, manda o bom senso que escolhamos como melhor aproximação à raiz o extremo do intervalo que corresponde ao menor valor absoluto da função. O método da bissecção tem duas importantes vantagens do ponto de vista do cálculo automático:

- por um lado, a facilidade de programação;
- por outro lado, o facto de, mesmo quando as raízes não estão bem isoladas, o método permitir sempre encontrar uma delas, embora não denuncie a existência das outras (é precisamente por causa da existência desta propriedade que nos permitimos habitualmente relaxar a condição de isolamento da raiz para a condição de a função ter sinais contrários nos extremos do intervalo).

Apesar da sua extrema robustez, o método da bissecção tem um ponto fraco que é frequentemente ignorado: a existência de sinais contrários nos valores da função nos extremos do intervalo só garante a existência de um zero no intervalo se a função for contínua; no caso de uma função descontínua pode ocorrer um polo de primeira ordem e, então, o método da bissecção reduzirá o intervalo inicial a um pequeno intervalo na vizinhança desse polo, sem denunciar o facto (salvo se, acidentalmente, ocorrer um *overflow*); por isso, após a terminação da iteração, devemos sempre prever um teste dos valores de  $f(a_n)$  e de  $f(b_n)$  antes de aceitarmos o intervalo como enquadrante de um zero.

### Exercício 2.2

1. Construa um fluxograma para o método da bissecção, incluindo todos os cuidados na terminação.
2. Escreva um programa para o fluxograma anterior e teste-o sobre os exercícios e problemas anteriores, ou sobre uma das seguintes funções:

$$f(x) = 2^{\sqrt{x}} - 10x + 1$$

$$f(x) = x - \ln x - x$$

$$f(x) = e^{\sin x} \cos(2x + 1)$$

$$f(x) = \cot x \sin(3x) - x - 1$$

---

Excelente pela simplicidade e robustez para implementação informática, o método da bissecção é pouco apreciado no cálculo manual, devido a uma certa lentidão de convergência. Tal lentidão resulta essencialmente do facto de o algoritmo utilizar relativamente mal a informação que ele próprio vai gerando; com efeito, calculando os valores da função nos dois extremos do intervalo, utiliza apenas o sinal desses valores para o cálculo do novo ponto. O método que se apresenta em seguida procura explicitamente obviar a esse inconveniente, embora à custa de um cálculo ligeiramente mais trabalhoso.

## 2 Zeros reais de uma função real

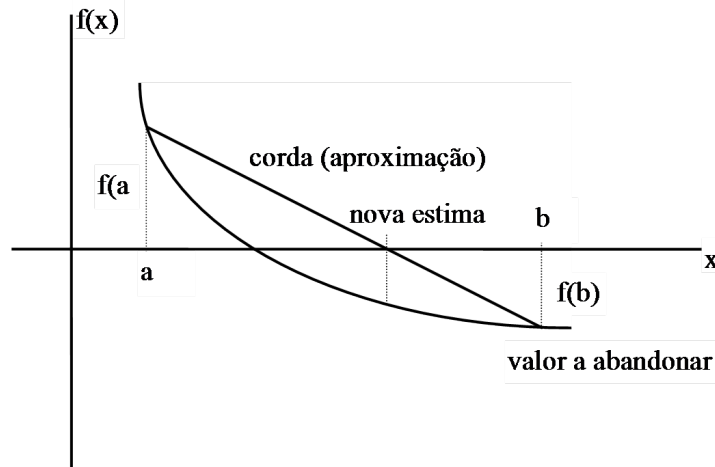


Figura 2.4: Interpretação geométrica do método da corda

### 2.3 Método da Corda

A ideia por trás do *método da corda*, também chamado *método da falsa posição* ou *regula falsi*, é muito simples: ao tentar reduzir o intervalo, se um valor extremo for grande (em valor absoluto) e o outro for pequeno, então o zero encontra-se, provavelmente mais perto do valor pequeno que do grande. Uma maneira simples de implementar este conceito consiste em traçar uma recta que passa pelos pontos extremos do intervalo,  $(x_1, f(x_1))$  e  $(x_2, f(x_2))$  e utilizá-la como aproximação da função. Esta recta tem por equação

$$y(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1)$$

e tem o zero

$$x = \frac{x_1 \cdot f(x_2) - x_2 \cdot f(x_1)}{f(x_2) - f(x_1)}$$

que constitui o próximo ponto a ensaiar e, portanto, o próximo ponto de divisão do intervalo. Encontrado este novo ponto, calcula-se o valor  $f(x)$  e abandona-se o ponto antigo em que a função tiver o mesmo sinal que neste, tal como se fazia na bissecção.

---

#### Algoritmo 2.2

Algoritmo da corda

DADOS  $a_0, b_0$  :  $f(a_0) \cdot f(b_0) < 0$

PARA  $n=0, 1, 2, \dots$  ATÉ ONDE FOR CONVENIENTE

SEJA  $w = (a_n \cdot f(b_n) - b_n \cdot f(a_n)) / (f(b_n) - f(a_n))$

SE  $f(a_n) \cdot f(w) < 0$  SEJAM  $a_{n+1} = a_n$  ;  $b_{n+1} = w$

CASO CONTRÁRIO SEJAM  $a_{n+1} = w$  ;  $b_{n+1} = b_n$



O inconveniente desta técnica em relação à da bissecção consiste em não ser possível prever *a priori* o momento da paragem, o que obriga, a cada iteração, a testar se já se verifica a condição de terminação pretendida, o que pode corresponder a uma perda de tempo significativa, principalmente se o cálculo dos valores da função for relativamente expedito.

#### Exemplo: 22.8 Método da Corda

Vamos usar o método da corda para melhorar a raiz da equação

$$x^4 + 2x^3 - x - 1 = 0$$

situada no intervalo  $(0, 1)$ , trabalhando com cinco algarismos significativos. Para efeitos de comparação, note-se que este é o mesmo problema que propusemos atrás para o método da bissecção.

a	f(a)	b	f(b)	x	intervalo
0,00000E+00	-1,00000E+00	1,00000E+00	1,00000E+00	5,00000E-01	1,00000E+00
5,00000E-01	-1,18750E+00	1,00000E+00	1,00000E+00	7,71429E-01	5,00000E-01
7,71429E-01	-4,99123E-01	1,00000E+00	1,00000E+00	8,47530E-01	2,28571E-01
8,47530E-01	-1,13991E-01	1,00000E+00	1,00000E+00	8,63132E-01	1,52470E-01
8,63132E-01	-2,20505E-02	1,00000E+00	1,00000E+00	8,66085E-01	1,36868E-01
8,66085E-01	-4,12384E-03	1,00000E+00	1,00000E+00	8,66635E-01	1,33915E-01
8,66635E-01	-7,66330E-04	1,00000E+00	1,00000E+00	8,66737E-01	1,33365E-01
8,66737E-01	-1,43018E-04	1,00000E+00	1,00000E+00	8,66756E-01	1,33263E-01
8,66756E-01	-2,68891E-05	1,00000E+00	1,00000E+00	8,66760E-01	1,33244E-01
8,66760E-01	-2,43988E-06	1,00000E+00	1,00000E+00	8,66760E-01	1,33240E-01
8,66760E-01	-2,43988E-06	1,00000E+00	1,00000E+00	8,66760E-01	1,33240E-01

Tabela 2.3: Aplicação do método da corda

A observação da tabela 2.3 e do gráfico da figura 2.5 permite compreender o desenvolvimento do método.

Exemplo: 22.8 Método da Corda (cont.)

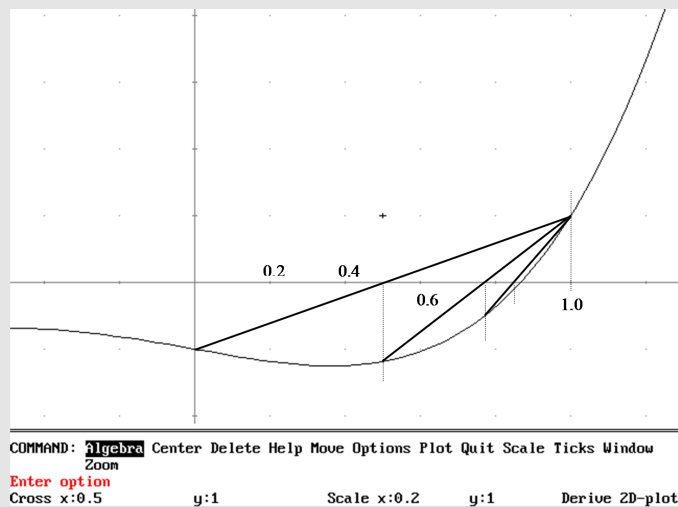


Figura 2.5: Aplicação do método da corda

O algoritmo termina na décima iteração porque se obtêm duas estimativas idênticas da raiz e não é possível, com esta precisão de representação, progredir para além deste ponto. A terminação obtém-se em 10 iterações, enquanto no método da bissecção, para o mesmo problema, precisamos de 17.

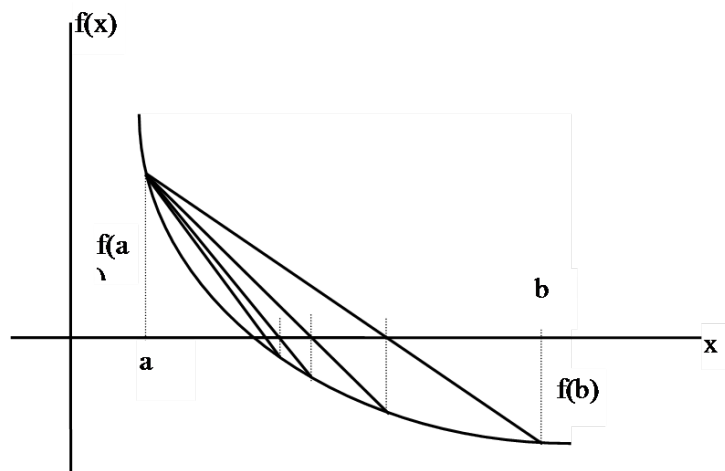


Figura 2.6: Convergência do método da corda

Para estudar a convergência do processo, admitamos que a raiz está isolada e que a segunda derivada da função tem sinal constante no intervalo  $(a, b)$ <sup>3</sup>. Suponhamos também, o que não arrasta perda de generalidade, que  $f''(x) > 0$  (o caso  $f''(x) < 0$  reduz-se a este fazendo  $f(x) =$

<sup>3</sup>Se assim não sucedesse, o método levar-nos-ia, apesar de tudo, rapidamente a esta situação.

$-f(x)$ , o que não afecta a raiz procurada). A curva é, portanto, côncava para cima, como na figura 2.6, e, portanto, situa-se inteiramente abaixo da corda.

Neste caso acontece que o extremo  $a$  permanece fixo e as aproximações sucessivas são

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(a)} \cdot (x_n - a)$$

e formam uma sucessão monotonamente decrescente tal que

$$a < \xi < \dots < x_{n+1} < x_n \dots < x_1 < b$$

Concluimos, portanto, após uns momentos de reflexão, que

- o ponto que se torna fixo é o extremo em que o sinal da função coincide com o da sua segunda derivada;
- as aproximações sucessivas encontram-se, em relação à raiz, do lado em que a função tem sinal oposto ao da sua segunda derivada.

Além disso, implementado nesta forma, o método tem ainda um ponto fraco tornado evidente pelo resultado anterior: a partir do momento em que um extremo permanece fixo, a convergência é retardada pelo facto de a corda se aproximar cada vez mais da vertical. Por isso, é frequente usar uma variante que consiste em, a cada iteração, reduzir a metade o valor da função que se conserva.

Uma outra variante que frequentemente se ouve aconselhar é a de conservar sempre os dois últimos pontos calculados, dentro da ideia de que os valores mais recentes serão os mais próximos e, portanto, a redução do intervalo é mais drástica. Porém, pode suceder, e sucede frequentemente, que os dois pontos venham a cair do mesmo lado da raiz, o que invalida o princípio do método e o transforma, de *método de interpolação*, em *método de extrapolação*, com todos os riscos inerentes. Por este facto, uma tal variante, correntemente distinguida pelo nome de *método da secante*, é formalmente desaconselhável.

### Exercício 2.3

1. Construa um fluxograma do método da corda e escreva um programa para esse fluxograma, propondo ao utilizador a escolha entre diferentes critérios de terminação; teste o programa sobre os exemplos e os exercícios propostos para o método da bissecção. Compare e comente os resultados.
2. Modifique o programa anterior para obter o método da corda melhorado e teste-o sobre os mesmos casos. Compare e comente os resultados.
3. A escolha, no método da corda melhorado, do divisor 2 para a ordenada do ponto fixo é, evidentemente, arbitrária. Discuta outras escolhas possíveis do divisor e as condições em que podem ser usadas com vantagem.
4. Mostre, por meio de diagramas, como pode falhar o método da secante.

## 2.4 Método da tangente

Os métodos discutidos atrás são *métodos intervalares* e têm o inconveniente óbvio de exigir o passo prévio de isolamento da raiz ou, pelo menos, de identificação de um intervalo em que a função mude de sinal, o que, como vimos, nem sempre é tarefa fácil.

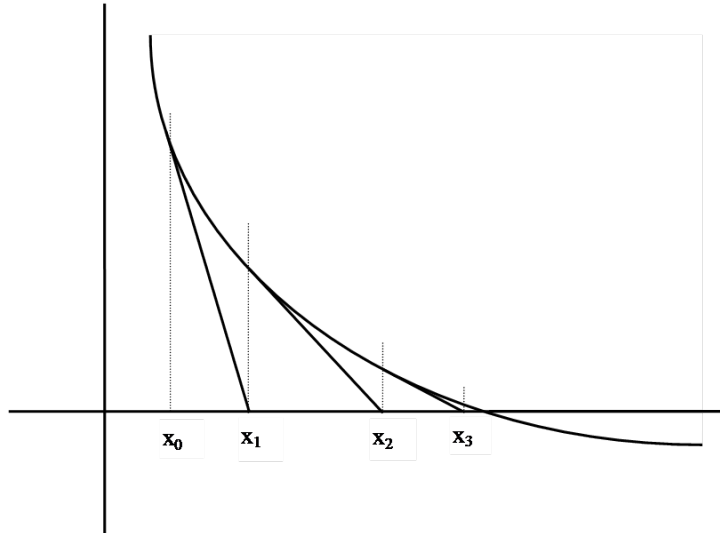


Figura 2.7: Método da tangente

Por isso se criou um método, dito *método da tangente* ou *método de Newton*, que parte apenas de um valor plausível, embora eventualmente grosseiramente errado, da raiz. Conceptualmente, pode ser considerado como apenas uma extensão, ou passagem ao limite, do método da secante e tem, portanto, todos os inconvenientes potenciais deste: em termos geométricos, consiste em substituir o gráfico da função não pela secante que liga os pontos extremos do intervalo, mas pela tangente no ponto considerado, usando o zero desta como nova aproximação à raiz.

Seja  $x_k$  o valor aproximado actual; a equação da tangente à curva nesse ponto é

$$y(x) = f(x_k) + f'(x_k) \cdot (x - x_k)$$

de modo que o valor  $y(x) = 0$  é

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

que constitui a nova aproximação.

## Exemplo: 22.9 S

veja, uma vez mais, o cálculo da raiz da equação

$$x^4 - 2x^3 - x - 1 = 0$$

tal que

$$f'(x) = 4x^3 - 6x^2 - 1$$

Comecemos pelo extremo direito  $x = 1$ , construindo a tabela 2.4.

$n$	$x_n$	$f(x_n)$ $x_n^4 - 2x_n^3 - x_n - 1$	$f'(x_n)$ $4x_n^3 - 6x_n^2 - 1$	$x_{n+1}$ $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
0	1	1	9	0,888889
1	0,888889	0,140071	6,55007	0,867504
2	0,867504	0,00455051	6,12679	0,866761
3	0,866761	3,67246E-06	6,11235	0,86676
4	0,86676	-2,43988E-06	6,11233	0,86676
5	0,86676	-2,43988E-06	6,11233	0,86676

Tabela 2.4: Aplicação do método da tangente

A convergência deu-se em 4 iterações o que, em comparação com os métodos anteriores, é muito bom.

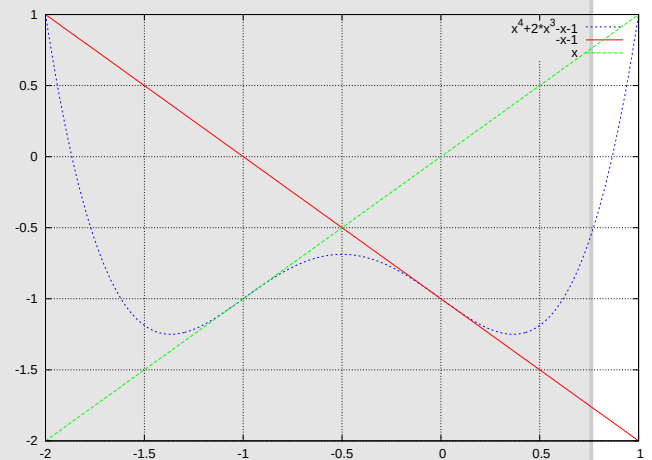
Tentemos, porém, partir do outro extremo  $x = 0$ , construindo a tabela 2.5.

$n$	$x_n$	$f(x_n)$ $x_n^4 - 2x_n^3 - x_n - 1$	$f'(x_n)$ $4x_n^3 - 6x_n^2 - 1$	$x_{n+1}$ $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
0	0	-1	-1	-1
1	-1	-1	1	0
2	0	-1	-1	-1
3	-1	-1	1	0
4	0	-1	-1	-1

Tabela 2.5: Aplicação do método da tangente, com ponto inicial diferente.

O que se passou desta vez?

### Exemplo: 22.9 S (cont.)



O gráfico da figura 2.8 ilustra a situação.

Figura 2.8: Não convergência no método da tangente

Quando funciona bem, o método da tangente é excelente, até porque, nas vizinhanças da raiz, tende, a cada iteração, a dobrar o número de algarismos exactos da solução. Porém, as suas limitações são muitas e muito severas, como mostra o exemplo anterior. A existência dessas limitações faz com que o método da tangente deva ser evitado, a menos que se conheça muito bem a estrutura local da função, o que é um tanto contraditório, porque, se essa estrutura local fosse bem conhecida não seria certamente difícil encontrar um intervalo de partida para um dos métodos anteriores.

Um inconveniente óbvio do método da tangente é o de exigir o conhecimento da derivada da função, e o seu cálculo em cada iteração; por este facto, tem sido sugerido que, em vez de calcular o valor da derivada em cada ponto, se trabalhe sempre com o seu valor no ponto original; uma tal estratégia, diminuindo o esforço de cálculo, aumenta o número de iterações necessárias para obter uma dada precisão. Além disso, em certos casos, pode tender a aumentar a robustez do método, mas uma tal propriedade depende, uma vez mais, da estrutura local da função.

Em suma, trata-se de um método indiscutivelmente útil, mas apenas para usar em terreno muito bem conhecido, e sempre com grande precaução.

## 2.5 Método de iteração de Picard-Peano

Todos os métodos apresentados até agora são, evidentemente, iterativos, visto usarem repetidamente o mesmo algoritmo para obterem aproximações sucessivamente melhores ao resultado pretendido e todos eles repousam na mesma ideia de substituir a função por uma aproximação adequadamente simplificada. O método, também iterativo, que vamos apresentar agora repousa

sobre uma ideia completamente diferente e constitui, no plano conceptual, um dos mais importantes métodos de resolução numérica de equações, embora a sua aplicação prática tenda a ser um tanto limitada.

A ideia básica é a seguinte: suponhamos uma equação  $f(x) = 0$  e, por não sabermos resolvê-la analiticamente, transformêmo-la de modo a dar-lhe a forma

$$x = g(x)$$

Se, por qualquer processo, tivermos obtido uma aproximação  $x_0$  da raiz e a substituirmos no segundo membro, obtemos um valor  $x_1 = g(x_0)$  que pode de novo ser utilizado para produzir  $x_2 = g(x_1)$  e assim sucessivamente:

$$x_n = g(x_{n-1})$$

Se esta sucessão for convergente (o que, como veremos, não é de modo nenhum garantido), isto é, se existir o limite

$$\xi = \lim_{n \rightarrow \infty} (x_n)$$

então, por passagem ao limite na expressão do termo geral, obtemos

$$\lim_{n \rightarrow \infty} (x_n) = g \left( \lim_{n \rightarrow \infty} (x_{n-1}) \right) = g \left( \lim_{n \rightarrow \infty} (x_n) \right)$$

isto é,

$$\xi = g(\xi)$$

o que prova que o limite, se existir, é, efectivamente, raiz da equação proposta.

Geometricamente, o método de iteração pode ser interpretado do seguinte modo: tracemos em um plano  $(x, y)$  os gráficos das funções

$$y = x$$

$$y = g(x)$$

e cada raiz  $\xi$  real da equação  $x = g(x)$  será a abcissa de um ponto de intersecção da curva  $y = g(x)$  com a recta  $y = x$ .

1. Se começarmos com uma abcissa  $x_0$ , calculamos  $y_0 = g(x_0)$ , isto é, subimos de  $x_0$  até encontrar a curva  $y = g(x)$  em  $A_0$ ;
2. em seguida, fazemos  $x_1 = y_0$ , isto é, deslocamo-nos horizontalmente até encontrar a recta  $y = x$  em  $B_1$  e descemos de novo até encontrar o eixo dos  $xx$  em  $x_1$ ;
3. Em seguida, calculamos de novo  $y_1 = g(x_1)$ , isto é, subimos de  $x_1$  a  $A_1$ , e assim sucessivamente, subindo a "escada"  $A_0, B_1, A_1, B_2, A_2, B_3, A_3, \dots$  (ver figura 2.9) até chegarmos ao ponto de intersecção da curva com a recta que, evidentemente, representa a raiz da equação proposta.

## 2 Zeros reais de uma função real

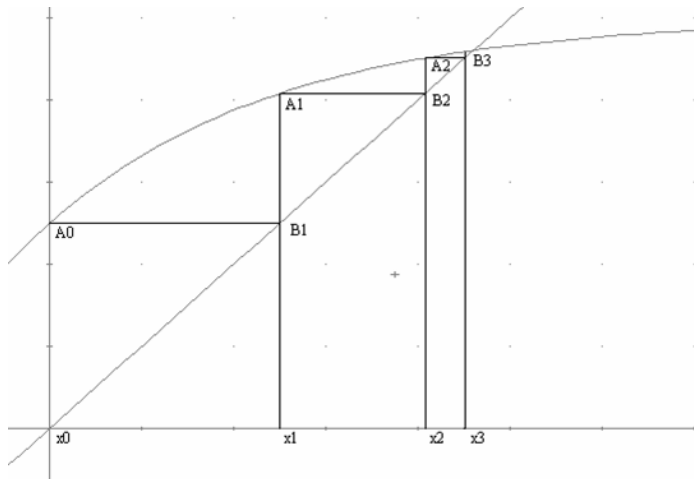


Figura 2.9: Método de Picard-Peano em escada

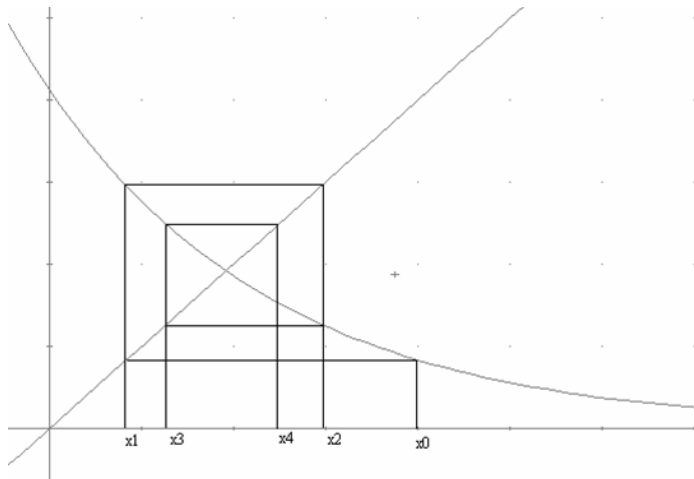


Figura 2.10: Método de Picard-Peano em teia de aranha

Uma outra configuração possível é a da "*teia de aranha*" (ver figura 2.10), correspondente ao caso de  $y = g(x)$  ser decrescente, caso em que, em vez de uma sucessão de aproximações que tendem monotonicamente para a raiz, obtemos uma sucessão oscilante que tende para a raiz enquadrando-a sucessivamente de mais perto.

Porém, nem sempre o método resulta convergente (não se esqueça o leitor que, na prova anterior, pusemos explicitamente a condição *se o limite  $\xi$  existir* ...), como resulta claramente das figuras seguintes (ver figuras 2.11 e 2.12).

Há, pois, para poder utilizar-se o método com segurança, que investigar as condições em que



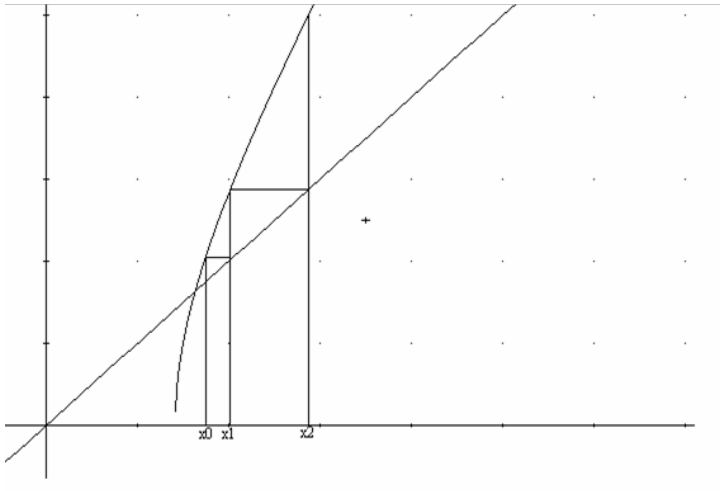


Figura 2.11: Método de Picard-Peano em escada divergente

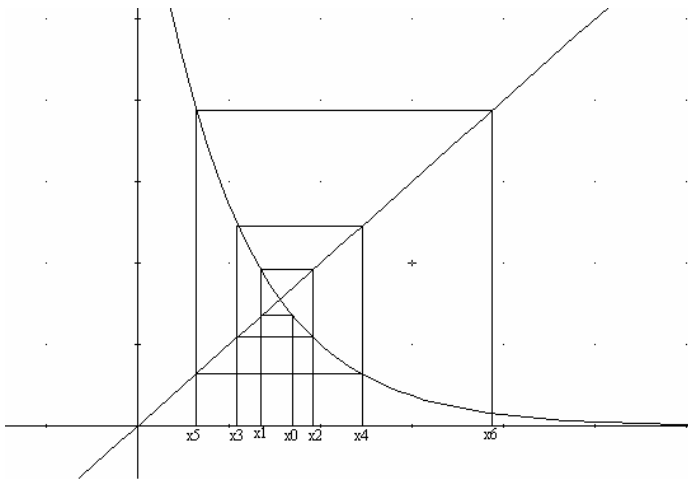


Figura 2.12: Método de Picard-Peano em teia de aranha divergente

converge. Uma simples análise das figuras mostrará facilmente que a condição de convergência corresponde a ser

$$|g'(x)| < 1$$

Uma outra questão interessante que pode pôr-se é a de saber se, dado um conjunto de iterações ainda não suficientemente aproximado da raiz, será possível acelerar o processo, extrapolando as tendências observadas.

## 2 Zeros reais de uma função real

Imaginemos uma coluna de iterações

$$\begin{aligned}x_0 \\x_1 &= g(x_0) \\x_2 &= g(x_1) \\&\dots \\x_k &= g(x_{k-1}) \\x_{k+1} &= g(x_k)\end{aligned}$$

e suponhamos, antes de mais, que pretendemos interpolar entre  $x_k$  e  $x_{k+1}$ ; para isso, teríamos que formar

$$(1-w).x_k + w.x_{k+1} = (1-w).x_k + w.g(x_k)$$

com  $0 < w < 1$ . Pois bem, para extrapolar, faremos exactamente o mesmo, mas com  $w > 1$ :

$$x_{k+1}^* = (1-w).x_k + w.g(x_k)$$

e procuremos determinar o melhor valor possível de  $w$  que será, naturalmente, função de  $k$ . Para isso, consideraremos que existe um número  $z_k \in (x, x_k)$  ou  $z_k \in (x_k, x)$  tal que

$$\begin{aligned}\epsilon_{k+1} &= (x_{k+1}^* - x) \\&= (x_k - x) + w. [g(x_k) - g(x) - (x_k - x)] \\&= (x_k - x) + w. \left[ \frac{g(x_k) - g(x)}{x_k - x} - \frac{x_k - x}{x_k - x} \right] \cdot (x_k - x) \\&= (x_k - x) + w. [g'(z_k) - 1] \cdot (x_k - x) \\&= [1 + w. (g'(z_k) - 1)] \cdot \epsilon_k\end{aligned}$$

A escolha óptima de  $w$  seria aquela que anulasse o factor multiplicativo de  $\epsilon_k$ ; porém, dado que  $z_k$  não é conhecido, a nossa melhor opção é tomar  $z_k = x_k$ , isto é, fazer,

$$1 + w. (g'(x_k) - 1) = 0$$

ou, supondo  $g'(x_k) \neq 1$

$$w = \frac{1}{1 - g'(x_k)}$$

isto é, considerando que  $f(x) = x - g(x)$  e  $f'(x) = 1 - g'(x)$

$$\begin{aligned}x_{k+1} &= x_k - \frac{x_k}{1 - g'(x_k)} + \frac{g(x_k)}{1 - g'(x_k)} \\&= x_k - \frac{f(x_k)}{f'(x_k)}\end{aligned}$$

o que corresponde, senão à iteração óptima, pelo menos à melhor que podemos calcular.

Ora, esta iteração de Picard-Peano melhorada não é senão a iteração proposta pelo método da tangente. Por esta razão, preferiremos a variante de Newton, sempre que a derivada de  $f(x)$  seja calculável sem complicação de maior.

**Exercício 2.4** Da condição geral de convergência,  $|g'(x)| < 1$ , será capaz de deduzir uma condição particular adequada ao caso da variante de Newton?

## 2.6 Resolução de Sistemas pelo Método de Iteração

No caso geral, não linear, o problema da resolução numérica de sistemas de equações é um problema muito complexo, para o qual não existe grande variedade de estratégias disponíveis. Com efeito, quando falham os métodos analíticos (directos) ou semi-analíticos (substituições sucessivas até se obter uma única equação), o único método com razoável possibilidade de utilização genérica (e, mesmo assim, com as dificuldades que desde já se antevêm) é uma extensão óbvia do método de iteração, naturalmente, sempre que possível, na sua versão melhorada de método de Newton.

### Exemplo: 22.10 Resolução de um sistema de duas equações a duas incógnitas

Seja o sistema

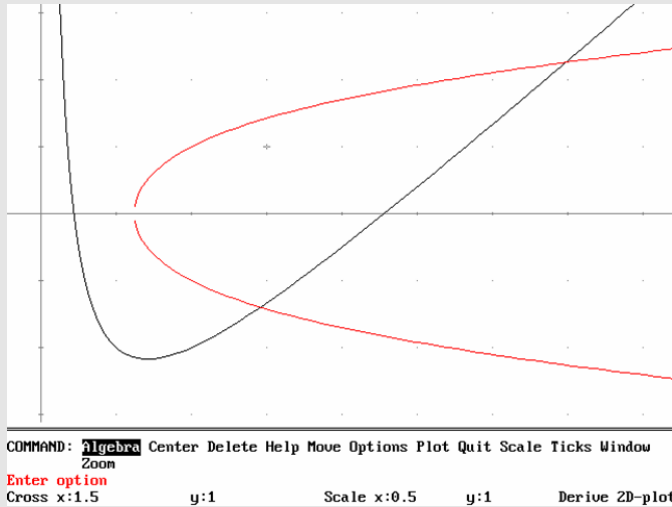
$$\begin{cases} f_1(x, y) = 2x^2 - x \cdot y - 5x + 1 = 0 \\ f_2(x, y) = x + 3 \log_1 0(x) - y^2 = 0 \end{cases}$$

de que pretendemos conhecer as raízes positivas, aproximadas a 5 casas significativas. Começemos por traçar gráficos aproximados das funções  $f_1$ ,  $f_2$  e, nas suas intersecções obteremos, como primeira aproximação das raízes,

$$\begin{cases} x'_0 = +3,5 \\ y'_0 = +2,3 \end{cases} \quad \begin{cases} x''_0 = +1,46 \\ y''_0 = -1,41 \end{cases}$$

## 2 Zeros reais de uma função real

### Exemplo: 22.10 Resolução de um sistema de duas equações a duas incógnitas (cont.)



Sejam dadas duas equações simultâneas em duas incógnitas

$$\begin{cases} f_1(x,y) = 0 \\ f_2(x,y) = 0 \end{cases}$$

de que pretendemos encontrar raízes reais com um dado grau de precisão. Sejam

$$\begin{cases} x = x_0 \\ y = y_0 \end{cases}$$

valores aproximados de uma raiz isolada, valor que foi obtido por um processo qualquer. Se representarmos o sistema na forma

$$\begin{cases} x = g_1(x,y) \\ y = g_2(x,y) \end{cases} \quad (2.1)$$

e construirmos aproximações sucessivas

$$\begin{cases} x_1 = g_1(x_0, y_0) \\ y_1 = g_2(x_0, y_0) \end{cases} \quad \begin{cases} x_2 = g_1(x_1, y_1) \\ y_2 = g_2(x_1, y_1) \end{cases} \quad \dots \quad \begin{cases} x_{n+1} = g_1(x_n, y_n) \\ y_{n+1} = g_2(x_n, y_n) \end{cases}$$

e se o processo de iteração convergir, isto é, se existir o duplo limite

$$\begin{cases} \xi = \lim_{n \rightarrow \infty} (x_n) \\ \eta = \lim_{n \rightarrow \infty} (y_n) \end{cases}$$

então

$$\begin{cases} \lim_{n \rightarrow \infty} (x_n) = \lim_{n \rightarrow \infty} (g_1(x_n, y_n)) \\ \lim_{n \rightarrow \infty} (y_n) = \lim_{n \rightarrow \infty} (g_2(x_n, y_n)) \end{cases}$$

e, portanto,

$$\begin{cases} \xi = g_1(\xi, \eta) \\ \eta = g_2(\xi, \eta) \end{cases}$$

isto é, o limite em questão é raiz do sistema proposto.

Porém, se o limite não existir, então o processo iterativo diverge e o método de iteração não pode ser usado, salvo mediante uma nova escolha mais adequada das funções  $g_1(\dots)$ ,  $g_2(\dots)$ . O teorema seguinte, que não demonstraremos, dá uma condição de convergência:

**Teorema 2.1** *Se, em uma vizinhança  $(a \leq x \leq A, b \leq y \leq B)$  existir uma e uma só raiz  $(\xi, \eta)$ , se  $g_1(\dots)$ ,  $g_2(\dots)$  forem diferenciáveis, se existirem tanto  $(x_0, y_0)$  como todas as aproximações sucessivas  $(x_n, y_n)$  e se forem válidas as desigualdades*

$$\left| \frac{\partial g_1}{\partial x} \right| + \left| \frac{\partial g_2}{\partial x} \right| \leq q_x < 1 \quad \left| \frac{\partial g_1}{\partial y} \right| + \left| \frac{\partial g_2}{\partial y} \right| \leq q_y < 1$$

*então o processo converge;*

*o teorema vale também para as desigualdades*

$$\left| \frac{\partial g_1}{\partial x} \right| + \left| \frac{\partial g_2}{\partial y} \right| \leq q_x < 1 \quad \left| \frac{\partial g_1}{\partial x} \right| + \left| \frac{\partial g_2}{\partial y} \right| \leq q_y < 1$$

A variante de Newton aplica-se na seguinte forma: sendo

$$\begin{cases} f_1(x, y) = 0 \\ f_2(x, y) = 0 \end{cases}$$

e escrevendo

$$\begin{cases} x = x_n + h_n \\ y = y_n + k_n \end{cases}$$

obtemos

$$\begin{cases} f_1(x_n + h_n, y_n + k_n) = 0 \\ f_2(x_n + h_n, y_n + k_n) = 0 \end{cases}$$

Se o determinante jacobiano

$$J(x_n, y_n) = \begin{vmatrix} f'_{1,x}(x_n, y_n) & f'_{1,y}(x_n, y_n) \\ f'_{2,x}(x_n, y_n) & f'_{2,y}(x_n, y_n) \end{vmatrix}$$

## 2 Zeros reais de uma função real

não for nulo, o sistema dá

$$h_n = - \frac{\begin{vmatrix} f_1(x_n, y_n) & f'_{1,y}(x_n, y_n) \\ f_2(x_n, y_n) & f'_{2,y}(x_n, y_n) \end{vmatrix}}{J(x_n, y_n)} \quad k_n = - \frac{\begin{vmatrix} f'_{1,x}(x_n, y_n) & f_1(x_n, y_n) \\ f'_{2,x}(x_n, y_n) & f_2(x_n, y_n) \end{vmatrix}}{J(x_n, y_n)}$$

de modo que

$$\begin{cases} x_{n+1} = x_n - \frac{f_1(x_n, y_n) \cdot f'_{2,y}(x_n, y_n) - f_2(x_n, y_n) \cdot f'_{1,y}(x_n, y_n)}{f'_{1,x}(x_n, y_n) \cdot f'_{2,y}(x_n, y_n) - f'_{2,x}(x_n, y_n) \cdot f'_{1,y}(x_n, y_n)} \\ y_{n+1} = y_n - \frac{f_2(x_n, y_n) \cdot f'_{1,x}(x_n, y_n) - f_1(x_n, y_n) \cdot f'_{2,x}(x_n, y_n)}{f'_{1,x}(x_n, y_n) \cdot f'_{2,y}(x_n, y_n) - f'_{2,x}(x_n, y_n) \cdot f'_{1,y}(x_n, y_n)} \end{cases}$$

**Exercício 2.5** Aplique o método de iteração ao sistema

$$\begin{cases} f_1(x, y) = 2x^2 - x \cdot y - 5x + 1 = 0 \\ f_2(x, y) = x + 3 \log_1 0(x) - y^2 = 0 \end{cases}$$

na forma

$$\begin{cases} x = \sqrt{\frac{x \cdot (y + 5) - 1}{2}} = g_1(x, y) \\ y = \sqrt{x + \log_1 0(x)} = g_2(x, y) \end{cases}$$

usando os valores iniciais

$$\begin{cases} x'_0 = +3,5 \\ y'_0 = +2,3 \end{cases} \quad \begin{cases} x''_0 = +1,46 \\ y''_0 = -1,41 \end{cases}$$

Aplique ao mesmo sistema a variante de Newton e compare resultados e desempenhos.

---



## Bibliografia

- [CB81] Conte and De Boor. *Elementary Numerical Analysis, an algorithmic approach*. McGraw-Hill International Editions, Singapore, 1981.
- [DB74] Dahlquist and Björck. *Numerical Methods*. Prentice-Hall, Inc., Englewood Cliffs, 1974.
- [Gol90] Herman H. Goldstine. *Remembrance of things past*. ACM Press - Addison-Wesley, Reading, Massachusetts, 1990.
- [Gol91] Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1), March 1991.
- [GU99] René Goscinny and Albert Uderzo. *Le Tour de Gaule d'Astérix*. Hachette, Paris, 1999.
- [Ham71] Hamming. *Introduction to applied numerical analysis*. McGraw-Hill-Kogakusha, Ltd., Tokyo, 1971.
- [IEE19] IEEE. Ieee standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, July 2019.
- [JR75] Jensen and Rowland. *Methods of Computation*. Scott, Foresman & Co, Glenview, 1975.
- [Knu81] D.E. Knuth. *The Art of Computer Programming: Seminumerical algorithms*. Addison-Wesley series in computer science and information processing. Addison-Wesley, 1981.
- [Lie68] Lieberstein. *A course in numerical analysis*. Harper and Row, New York, 1968.
- [Max19] Maxima. Maxima, a computer algebra system. version 5.43.0, 2019.
- [Moo66] R.E. Moore. *Interval analysis*. Prentice-Hall series in automatic computation. Prentice-Hall, 1966.
- [Mul89] Muller. *Arithmétique des ordinateurs*. Masson, Paris, 1989.
- [Sof] Soft WhareHouse. Derive, a mathematical assistant <sup>TM</sup>. actualmente disponíve nas calculadoras TI-Nspire CAS.
- [Wal90] Wallis. *Improving floating-point programming*. Wiley, New York, 1990.
- [Wik19] Wikipedia contributors. Ieee 754 — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=IEEE\\_754&oldid=917946439](https://en.wikipedia.org/w/index.php?title=IEEE_754&oldid=917946439), 2019. [Online; accessed 26-September-2019].
- [Wil65] Wilkinson. *Rounding errors in algebraic processes*. Clarendon Press, Oxford, 1965.



## *Bibliografia*

- [Zac96] Joseph L. Zachary. *Introduction to scientific programming: computational problem solving using Maple and C*. Springer Verlag New York Inc., New York, 1 edition, 1996.