Capítulo 8

Métodos iterativos para equações não lineares

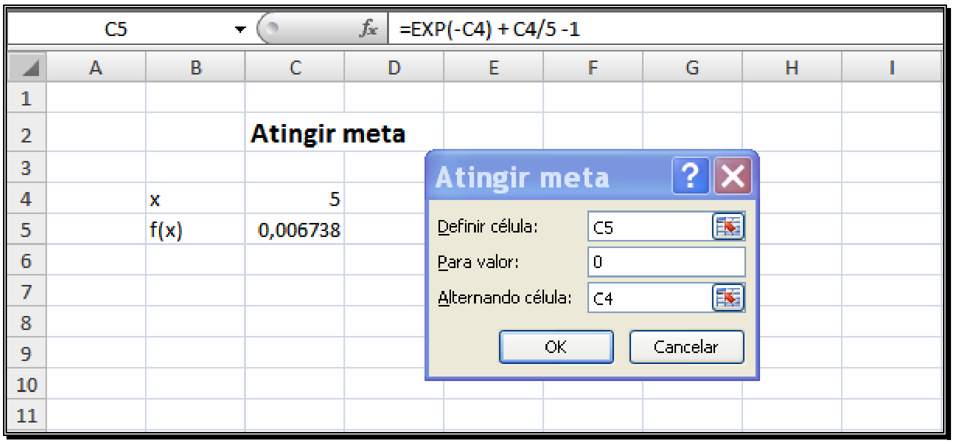
Usando o Goal Seek (Atingir meta)

O método mais simples de achar as soluções (ou raízes) duma equação "complicada" da forma f(x) = 0 consiste em traçar o gráfico da função, para ter uma idéia da localização aproximativa das raízes. Depois disso, podemos usar a ferramenta "Atingir meta" do Excel (*Dados>Ferramentas de Dados>Teste de Hipóteses>Atingir meta*). (O Excel tem uma segunda ferramenta, o "Solver", para achar numericamente os zeros de funções e para tratar problemas da estatística e análise de dados. Veja no capítulo 14 para mais detalhes.)

Exemplo: No estudo da radiação térmica aparece a equação

$$e^{-x} + \frac{x}{5} - 1 = 0$$

O gráfico desta função tem um ponto zero perto de x=5. Para obter este valor com maior exatidão, ativamos *Atingir meta.*



O resultado será: x = 4,965105 com f(x) = -1,8E-06.

Temos que falar um pouco sobre "Métodos iterativos" que servem, entre outras coisas, para encontrar soluções de equações não lineares como $x^4 - 4x^3 - x + 5 = 0$ ou $2e^x - x\,\text{sen}(x+3) = 0$. No primeiro caso, existe uma fórmula resolvente geral, como no caso da equação cúbica, mas, ela é complicada, e no segundo caso, não existe nenhuma fórmula resolvente.

Um **método iterativo**, consiste de um modo geral, numa aproximação inicial x_0, também designada **iterada inicial**, e num processo de obter sucessivamente novas iteradas x_{n+1} a partir das anteriores x_n,... Desta forma, pretendemos obter uma sucessão que convirja para *z*, solução da equação f(x) = 0, também designada por **raiz da equação**, ou **zero da função** f.

Um processo clássico para ilustrar uma iteração é o algoritmo der *Heron* para determinar a raiz quadrada de um número N>=0.
Segundo Heron, começa-se com x_1 = 1 como primeira estimativa para $N^{1/2}$.
Depois calcula-se com a fórmula

$$x_2 = \frac{1}{2}\left(x_1 + \frac{N}{x_1}\right)$$

um novo, oxalá melhor, valor para $N^{1/2}$. (Esta fórmula de Heron segue do método iterativo de Newton, veja mais adiante.) O novo valor utiliza-se como x_1, e, de novo, calcula-se um valor melhorado x_2 etc. Veja o seguinte esquema

B5 fx =0,5*(A5+B$1/A5)

	A	B	C	D	E
1	N=	3			
2					
3					Método de Heron para extrair raiz quadrada de N
4	x1	x2			
5	1	2			
6	2	1,75			
7	1,75	1,732142857			
8	1,732142857	1,73205081			
9	1,73205081	1,732050808			
10	1,732050808	1,732050808			
11	1,732050808	1,732050808			
12	1,732050808	1,732050808			
13					

que contém na coluna A os valores x_n e na coluna B os novos valores x_{n+1}

A5: 1; B5: =0,5*(A5+B$1/A5)
A6: =B5; B6: =0,5*(A6+B$1/A6)

Copie A6:B6 até que dois iterações sucessivas diferem em menos de um número pequeno ε (épsilon), por exemplo ε = 10^{-6}. A execução do algoritmo termina, se Abs(x_2-x_1) for menor ou igual a ε. O último valor de x_2 representa a raiz quadrada de N com a precisão estabelecida pelo ε.

Geralmente, escreve-se a fórmula de Heron na forma

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \text{ onde } n = 0,1,2,\ldots$$

x_0 é a aproximação inicial. Para calcular a p-ésima raiz de um número positivo a, podemos utilizar a seguinte fórmula de iteração

$$x_{n+1} = \frac{1}{p}\left((p-1)x_n + \frac{a}{x_n^{p-1}} \right)$$

A Sub-rotina "Heron" contém a constante de precisão ε como critério de parar o algoritmo.

(Geral) Heron

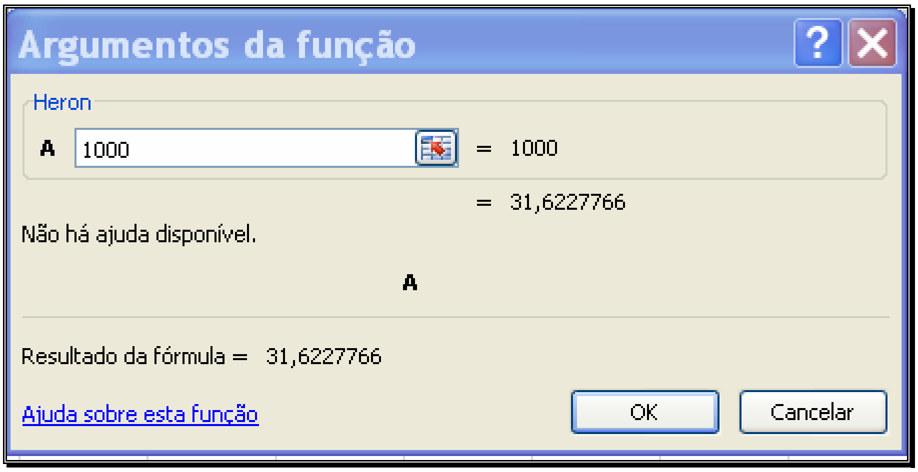
```
Sub Heron()
Dim a As Double, x As Double, y As Double

Const eps = 0.000001


    a = InputBox("De qual número quer calcular a raiz?:  ")

    x = 1: y = 0

     Do While Abs(y - x) > eps
      y = x 'valor anterior
      x = 0.5 * (x + a / x) ' valor novo

     Loop

    MsgBox "A raiz de " & a & " é igual a " & x

 End Sub
```

Segue aqui também uma versão *Função* "Heron(a)" com sua janela "Heron"

(Geral)

```
Function Heron(a As Double) As Double

Dim x As Double, y As Double

Const eps = 0.000001

    x = 1: y = 0

     Do While Abs(y - x) > eps
      y = x 'valor anterior
      x = 0.5 * (x + a / x) ' valor novo

     Loop

    Heron = x

 End Function
```

Após digitar esta função, pode-se voltar para a planilha Excel e escrever =Heron(1000), para extrair a raiz quadrada do número 1000 com a exatidão ε.



Método de Newton-Raphson

A raiz quadrada de *a* é a solução da equação f(x) = $x^2 - a = 0$. Para o cálculo duma raiz da equação f(x) = 0 utiliza-se muitas vezes o algoritmo de *Newton-Raphson*, dada pela seguinte fórmula de iteração (fórmula recursiva)

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

onde x_n é um valor aproximativo da raiz buscada e x_{n+1} é uma melhoria de x_n .
Já mais acima dissemos que para a maioria das equações não existe nenhuma fórmula resolvente geral, para determinar as raízes. Nesse caso, usamos métodos numéricos para obter uma solução aproximativa, tão perto quanto queiramos da solução exata. A seqüência $\{x_n\}$ convergirá para a raiz, se f(x), f'(x) e f"(x) forem contínuas no intervalo que contém a raiz.
A derivada da função $f(x_n) = x_n^2 - a$ é $f'(x_n) = 2\ x_n$, o que permite deduzir a fórmula de Heron imediatamente como caso especial da fórmula de Newton. (A fórmula de Newton-Raphson é obtida tomando uma série de Taylor para f(x) = 0, retendo os termos de primeira ordem.)

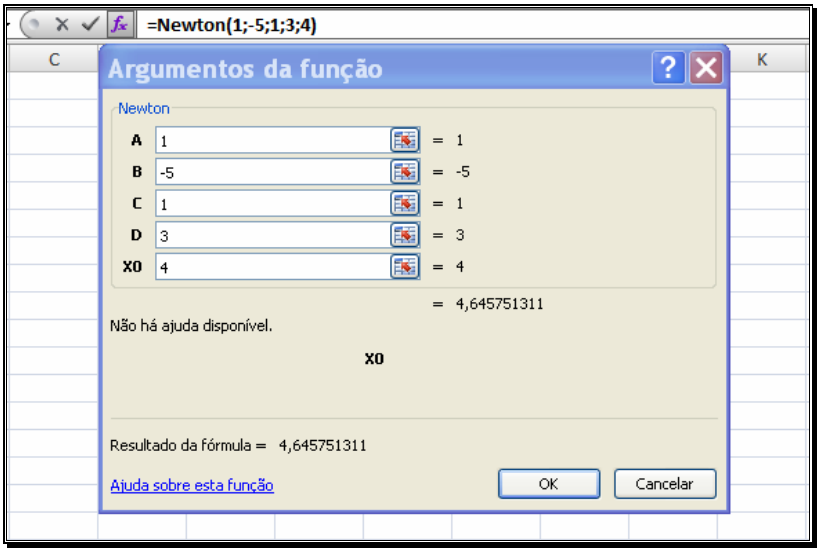
Primeiro, vamos escrever uma função VBA usando as idéias aplicadas na função "Heron". Tomemos o caso especial da função

$$f(x) = ax^3 + bx^2 + cx + d$$

partindo com a aproximação x_0.
Como **exemplo** particular, calculamos a raiz de $f(x) = x^3 - 5x^2 + x + 3$ com $x_0 = 4$ (valor que se tira do gráfico da função).

```
(Geral)                                            Newton
Function Newton(a, b, c, d, x0) As Double

Dim x As Double, y As Double

Const eps = 0.000001

    x = x0: y = 0

     Do While Abs(y - x) > eps
      y = x 'valor anterior
      x = x - (a * x ^ 3 + b * x ^ 2 + c * x + d) / (3 * a * x ^ 2 + 2 * b * x + c) ' valor novo

     Loop

    Newton = x

 End Function
```

A função tem os seguintes três zeros: x1= -0,645751, x2 = 1, x3 = 4,645751



Na maioria dos casos, basta aproximar a derivada por

$$f'(x) \approx \frac{f(x+h)-f(x)}{h}$$

Na seguinte figura vemos uma implementação do processo de Newton-Raphson numa planilha do Excel utilizando esta expressão para a derivada na coluna E. Pretendemos determinar as três soluções reais da equação $x^5 - x - 0,2 = 0$.

Sabemos (gráfico!) que elas devem ficar perto de -0,5; 1; -1. Na planilha buscamos a terceira raiz utilizando como valor initial x_0 = -0,9. (Os valores, exatos com quatro casas decimais, são -0,2003; 1,0448; -0,9421.)

Uma vez criada a planilha, nos será somente preciso introduzir a função em questão na célula B7, o resto faz uma macro, que será ativada com Ctrl-i.
A planilha é feita da seguinte maneira:

A7: =G$2
B7: aqui introduz-se o termo da função, em nosso caso =A7^5-A7-0,2
C7: =A7+G$3, copiar até C15 (8 iterações)
E7: =(D7-B7)/G$3, copiar até E15
G7: =A15 (resultado depois de 8 iterações)
A8: =A7-B7/E7, copiar até A15

As colunas A, C e E não se mudam mais. Com uma macro copiamos B7 até B15, depois B7 até D7. Em seguida copia-se D7 até D15. A macro faz isso com Ctrl-i.
(Já escrevemos varias macros para o Excel 2003. No Excel 2007 utilizamos o caminho *Desenvolvedor>Código>Gravar Macro*. Na seção *Código* aparecerá depois *Parar Macro*.)
Para analisar, depois, uma nova função, é só preciso introduzir o seu termino na célula B7, dar um valor inicial na G2 e ativar a macro com Ctrl-i.
Por **exemplo** =EXP(-A7)+A7/5-1 para a equação

$$e^{-x}+\frac{x}{5}-1=0$$

ou =A7^3-2*A7-5 para a famosa *equação de Wallis* $x^3-2x-5=0$ (solução: 2,094552..)

B7 f_x =A7^5-A7-0,2

	A	B	C	D	E	F	G	H
1								
2		**Newton-Raphson**				x0=	-0,9	
3						h=	0,0000001	
4								
5								
6	**x**	**f(x)**	**x+h**	**f(x+h)**	**f'(x)**			
7	-0,9	0,10951	-0,9	0,10951	2,280499	x=	-0,942087	
8	-0,948020186	-0,01773	-0,94802	-0,01773	3,038688			
9	-0,942184958	-0,00029	-0,942185	-0,00029	2,940167			
10	-0,942086893	-8E-08	-0,942087	2,14E-07	2,938526			
11	-0,942086866	1,67E-14	-0,942087	2,94E-07	2,938526			
12	-0,942086866	0	-0,942087	2,94E-07	2,938526			
13	-0,942086866	0	-0,942087	2,94E-07	2,938526			
14	-0,942086866	0	-0,942087	2,94E-07	2,938526			
15	-0,942086866	0	-0,942087	2,94E-07	2,938526			
16								
17								
18	Makro:	**Ctrl-i**						

A macro pode ser inspecionada por meio do *Código>Visual Basic* no *Desenvolvedor*.

```
(Geral)                                                    (Declaração)

Sub N_Raphson()
'
' N_Raphson Macro
' Newton_Raphson
'
' Atalho do teclado: Ctrl+i
'
    Range("B7").Select
    Selection.AutoFill Destination:=Range("B7:B15"), Type:=xlFillDefault
    Range("B7:B15").Select
    Range("B7").Select
    Selection.Copy
    Range("D7").Select
    ActiveSheet.Paste
    Application.CutCopyMode = False
    Selection.AutoFill Destination:=Range("D7:D15"), Type:=xlFillDefault
    Range("D7:D15").Select
    Range("G7").Select
End Sub
```

Primeiro, é selecionada B7 e copiada, com a alça de preenchimento, até B15. Retorna-se para B7 e copia-se o conteúdo dela para a célula D7. Também esta célula será copiada até a linha 15. Finalmente, estaciona-se o marcador na célula G7 do resultado.

O Método de Bolzano (1781-1848)

No método da bisseção ou método de Bolzano divide-se o intervalo [a,b], onde espera-se o zero da função em estudo, sucessivamente ao meio até encontrar o zero com a exatidão desejada.

Seja $f(x) = e^{-x} + x/5 - 1$ a função cujo zero, z, espera-se que fique entre a = 4 e b = 6. x = (a+b)/2 é o centro do intervalo. Tomemos f(a) como valor de comparação.

Se o meio x do intervalo já for o ponto zero, então serão f(x) e $f(a)\cdot f(x) = 0$, e não nos quedaria mais nada para fazer. Mas, geralmente, f(x) não será 0. (Em nosso caso temos x = 5 e f(5) = 0,00673795...)

Se o zero z da função f ficar do lado **esquerdo** do meio x, então temos $f(a)\cdot f(x) < 0$. Neste caso, seguimos buscando só no intervalo [a,x], ou seja, nós escolhemos b = x e calculamos o novo ponto médio x = (a+b)/2, em nosso exemplo x = (4+5)/2 = 4,5.

Se o zero fica à **direita** do ponto médio, resulta $f(a)\cdot f(x) > 0$.
Nos tomamos, então, a = x e dividimos o intervalo do lado direito ao meio.

O processo é repetido até que seja obtida uma aproximação para a raiz exata z com uma tolerância ε desejada.

O método não é muito rápido. Se buscarmos uma solução com a exatidão de $|z - x| < \varepsilon$, teremos de fazer N divisões. Pode-se demonstrar que

$$N \geq (\ln(b-a) - \ln(\varepsilon))/\ln 2 - 1$$

Ou seja, para um dado intervalo [a,b] são necessárias, no mínimo, N iterações para se calcular a raiz z com tolerância ε.

Para obter, em nosso caso, um resultado correto com três casas decimais (ε = 0,001), temos de fazer N > 10 divisões (iterações). A seguinte planilha confirma este cálculo, pois o valor x = 4,965... aparece apenas na célula C15.

E5 =EXP(-B5)+B5/5-1

	A	B	C	D	E	F	G	H	I
1				a=	4				
2			**Método da Bisseção**	b=	6				
3									
4		a	x	b	f(a)	f(x)		**Macro:**	(Cursor em E5)
5		4	5	6	-0,181684361	0,006737947		Ctrl+b	
6		4	4,5	5	-0,181684361	-0,088891003			
7		4,5	4,75	5	-0,088891003	-0,041348305			
8		4,75	4,875	5	-0,041348305	-0,017364906			
9		4,875	4,9375	5	-0,017364906	-0,005327493			
10		4,9375	4,96875	5	-0,005327493	0,000701832			
11		4,9375	4,953125	4,96875	-0,005327493	-0,002313692			
12		4,953125	4,9609375	4,96875	-0,002313692	-0,000806144			
13		4,9609375	4,96484375	4,96875	-0,000806144	-5,22089E-05			
14		4,96484375	4,966796875	4,96875	-5,22089E-05	0,000324798			
15		4,96484375	4,965820313	4,966796875	-5,22089E-05	0,000136291			

Uma vez montada a planilha, é só colocar a função na célula E5 e ativar a macro com Ctrl+b. (O cursor deve estar sobre E5.)

B5: =E$1; C5: =(B5+D5)/2; D5: =E$2 (copiar C5 até C20 –ou mais embaixo)

E5: escreva a forma analítica da **função** f(x), por exemplo =EXP(-B5)+B5/5-1. Depois, copiamos ela por meio duma macro a F5 e, em seguida, até onde queiramos, por exemplo, até a linha 20. Em B6 temos =SE(E5*F5>0;C5;B5) e em D6: =SE(E5*F5<0;C5;D5). Copie B6 e D6 até B20, D20 –ou mais embaixo.

A macro "Bolzano" copia só os conteúdos das colunas E e F até a linha 30

(Geral) Bolzano

```
Sub Bolzano()
'
' Bolzano Macro
' Método de Bolzano (Bisseção)
'
' Atalho do teclado: Ctrl+b
'
    Selection.AutoFill Destination:=Range("E5:F5"), Type:=xlFillDefault
    Range("E5:F5").Select
    Selection.AutoFill Destination:=Range("E5:F30"), Type:=xlFillDefault
    Range("E5:F30").Select
End Sub
```

O código VBA "Bolzano" com a função f é muito simples

```
(Geral)
Sub Bolzano()
Dim a As Double, b As Double, x As Double
a = 2: b = 3: N = 30

For i = 1 To N Step 1
    x = (a + b) / 2
    If f(a) * f(x) < 0 Then b = x Else a = x
Next

MsgBox (" x = " & x & " e f(x)= " & f(x))

End Sub
Function f(y) As Double
 f = y ^ 3 - y ^ 2 - 5
End Function
```

O método de Bolzano (bisseção) não exige o conhecimento das derivadas, mas tem uma convergência lenta. O método de Newton-Raphson tem, no entanto, uma convergência extraordinariamente rápida.

Para concluir esta seção, dá-se, a seguir, uma prova do critério da convergência do método da bisseção.

Como a cada iteração o intervalo [a,b] é dividido ao meio, na n-ésima iteração, o comprimento do intervalo é dado por $b_n - a_n = (b-a)^n$.

Isso podemos expressar como

$$|x_n - x_{n-1}| = \frac{b-a}{2^{n+1}} \leq \varepsilon$$

introduzindo uma tolerância ε para o valor da raiz desconhecida z. $n = 0,1,2,...$

Esta é uma relação para o error absoluto do cálculo e, ao mesmo tempo, nos da uma fórmula para o número máximo de iterações necessárias para obter o valor da raiz desconhecida z. Pois, da última desigualdade resulta

$$(n+1)\ln 2 \geq \ln\left(\frac{b-a}{\varepsilon}\right)$$

$$n \geq \frac{\ln\left(\frac{b-a}{\varepsilon}\right)}{\ln 2} - 1 .$$

Isso significa, que no método da bisseção conhece-se de **ante-mão** o número máximo de iterações necessárias para alcançar uma tolerância desejada.

Método da falsa posição (regula falsi)

A idéia deste método é a de tomar, em contraste com o método de Newton, *dois* valores iniciais: x_1 e x_2, posicionados de tal maneira que a raiz exata da equação f(x) = 0 esteja no intervalo $[x_1, x_2]$, ou seja, que se cumpra a desigualdade $f(x_1) \cdot f(x_2) < 0$, porque as ordenadas $f(x_1)$ e $f(x_2)$ têm sinais opostos. (Trata-se dum método de bisseção junto com uma interpolação linear.)

A distância entre x_1 e x_2 deve ser o suficientemente pequena para que possamos estar seguros de que não fique outra raiz no intervalo $[x_1, x_2]$. Por meio da seguinte fórmula de iteração

$$x_{n+2} = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}$$

calculamos uma série de novas posições x_i que, geralmente, acercam-se pouco a pouco à raiz buscada. (Se f é uma função contínua em um intervalo [a,b] e $f(a) \cdot f(b) < 0$, então o método da falsa posição converge.)

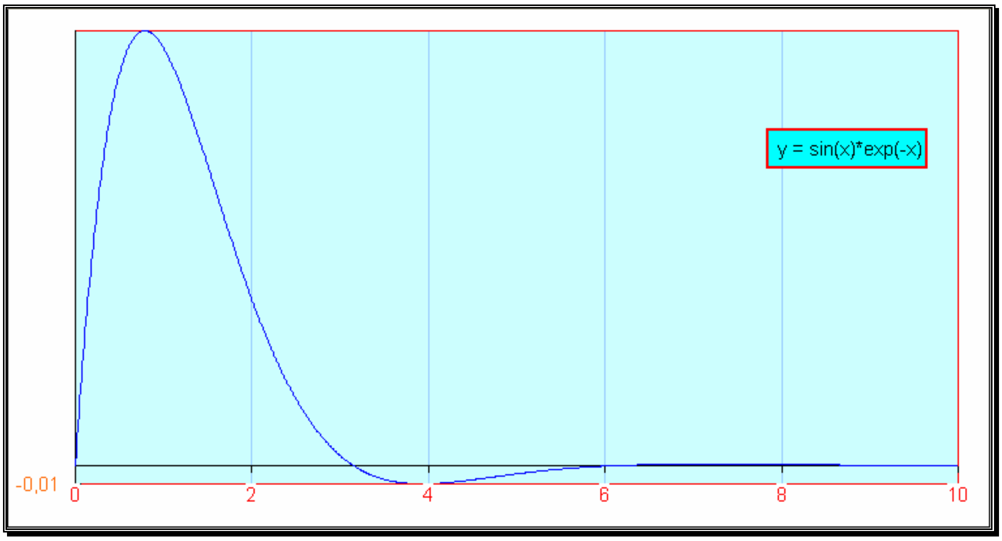
Para escrever o código VBA é aconselhável reescrever a fórmula de iteração da seguinte forma:

$$x = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

Exemplo: Utilize a macro "falspos", veja mais em frente, para encontrar as raízes da função $f(x) = sen(x) \cdot e^{-x}$.

Primeiro, traçamos o gráfico da função para ver onde, mais ou menos, estão localizados os zeros.

Como demonstra a seguinte figura, podemos esperar duas raízes no intervalo [1,7]. (Efetivamente trata-se de π e de 2π.)




```
Function falspos(a, b)
    Dim fa As Double, fb As Double, fx As Double
    Dim h As Double, teste As Double, n As Integer
    Dim x As Double
    Const nmax = 1000
    Const eps = 0.0000000001

    fa = f(a)
    fb = f(b)
    x = (a * fb - b * fa) / (fb - fa)
    n = 0

    Do
        'buscar um intervalo
        fx = f(x)
        teste = fa * fx
        If teste < 0 Then
            b = x
            fb = fx
        Else
            a = x
            fa = fx
        End If
        h = Abs(b - a)
        x = (a * fb - b * fa) / (fb - fa)

        n = n + 1
     Loop While h > eps And n <= nmax

      falspos = x

End Function
Function f(x)
  f = Sin(x) * Exp(x)
End Function
```

Falta ainda uma planilha do Excel para a Regula falsi:

E19 =SE(A19="";"";+C19-D19*(A19-C19)/(B19-D19))

	A	B	C	D	E	F	G	H
1						a	b	
2		Regula Falsi			Valores iniciais:	4	7	
3					eps:	1E-10		
4		Primeiro introduzir a função em B7						
5								
6	x1	f(x1)	x2	f(x2)	x			
7	4	-0,0138613	7	0,0005991	6,875710162			
8	7	0,0005991	6,8757102	0,0005766	3,683433183			
9	6,87571016	0,0005766	3,6834332	-0,0129633	6,739756103			
10	3,68343318	-0,0129633	6,7397561	0,0005215	6,621552741			
11	6,7397561	0,0005215	6,6215527	0,0004419	5,965132294			
12	6,62155274	0,0004419	5,9651323	-0,0008027	6,388465194			
13	5,96513229	-0,0008027	6,3884652	0,0001766	6,312109704			
14	6,38846519	0,0001766	6,3121097	5,247E-05	6,279844517			
15	6,3121097	5,247E-05	6,2798445	-6,26E-06	6,283283604			
16	6,27984452	-6,26E-06	6,2832836	1,835E-07	6,283185635			
17	6,2832836	1,835E-07	6,2831856	6,119E-10	6,283185307			
18	6,28318563	6,119E-10	6,2831853	-6,015E-14	6,283185307			
19	6,28318531	-6,015E-14	6,2831853	-4,576E-19	6,283185307			
20								

Entradas:

A7: =F$2; C7: =G$2

B7: =SE(A7="";"";EXP(-A7)*SEN(A7)) { trata-se de f(x) = sen(x)·e^{-x} }.
Copiar até B20, depois para as células D7:D20.
Em D7 temos então =SE(C7="";"";EXP(-C7)*SEN(C7))

E7: =SE(A7="";"";+C7-D7*(A7-C7)/(B7-D7)), copiar até E20

A8: =SE(ABS(A7-C7)>F$3;+C7;""); copiar até A20

C8: =SE(A8="";"";+E7); copiar até C20

Método de Gauss-Seidel

Para a resolução de sistemas de equações lineares existe um método iterativo desenvolvido por Gauss e melhorado por Seidel. Este método funciona, quando os coeficientes dos elementos na diagonal principal tem valores absolutos muito majores do que os coeficientes dos outros elementos.

Vejamos o seguinte exemplo:

$$25x + 2y + z = 69$$
$$2x + 10y + z = 63$$
$$x + y + 4z = 43$$

Primeiro, extraímos as variáveis da forma

$$x = (69 - 2y - z)/25$$
$$y = (63 - 2x - z)/10$$
$$z = (43 - x - y)/4$$

Somando os valores absolutos dos coeficientes das variáveis no lado direito, dá para a primeira equação (2+1)/25 = 0,12, para a segunda 0,3 e para a terceira 0,5. Pode-se demonstrar, que o método convirja para a solução exata, se estas somas de coeficientes são menores que 1. (Trata se de um critério suficiente.)

Começa-se dando aos coeficientes das variáveis valores iniciais arbitrários, por exemplo zero.

Primeira iteração:

$$x^{(1)} = (69 - 2y^{(0)} - z^{(0)})/25 = 2,76$$
$$y^{(1)} = (63 - 2x^{(1)} - z^{(0)})/10 = 5,748$$
$$z^{(1)} = (43 - x^{(1)} - y^{(1)})/4 = 8,623$$

Observe, que os valores já calculados são utilizados para obter o valor da variável na nova iteração. É isso a vantagem do método de Seidel com relação ao método de Gauss.

A implementação desse esquema numa planilha do Excel é de uma simplicidade surpreendente. Podemos escolher um esquema horizontal com C10: =(69-2*B6-B7)/25; C11: =(63-2*C10-B7)/10; C12: =(43-C10-C11)/4. Isso copiamos até conseguir uma convergência aceitável. Em nosso caso, paramos depois de 6 iterações. As soluções exatas são x = 2; y = 5, z = 9

G12 =(43-G10-G11)/4

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3			**Iteração de Gauss-Seidel**								
4										25x+2y+z=0	
5										2x+10y+z=63	
6										x+y+4z=43	
7					**Iterações**						
8		Início	1	2	3	4	5	6			
9								resultado			
10	x=	0	2,76	1,95524	1,996287	1,999908	2,000003	2			
11	y=	0	5,748	5,046652	5,00079	4,999945	4,999996	5			
12	z=	0	8,623	8,999527	9,000731	9,000037	9	9			
13											

No esquema vertical utilizamos as fórmulas

B11: =(69-2*C10-D10)/25;
C11: =(63-2*B11-D10)/10;
D11: =(43-B11-C11)/4

Gauss-Seidel

Iterações	x	y	z
0	0	0	0
1	2,76	5,748	8,623
2	1,95524	5,046652	8,999527
3	1,996287	5,00079	9,000731
4	1,999908	4,999945	9,000037
5	2,000003	4,999996	9
6	2	5	9

O número das iterações pode ser bem alto. Por exemplo precisamos para o seguinte sistema 77 Iterações para obter as soluções {2;1;-3}

$$\begin{array}{cccc} 2x & -y & -z & =6 \\ x & +3y & +2z & =-1 \\ 3x & +4y & +3z & =1 \end{array}$$

Mas, se mudarmos na última equação 3z por 4z, obteremos as soluções x=2,272727; y=-0,36364; z=-1,0909 depois 29 iterações apenas.

Aplicação de Gauss-Seidel (Distribuição de Temperatura)

Um método parecido ao método de Gauss-Seidel é usado para determinar a distribuição de temperatura numa placa metálica quadrada.

	A	B	C	D	E	F
1						
2		**0**	**100**	**100**	**0**	
3		**0**	**T1**	**T2**	**0**	
4		**0**	**T3**	**T4**	**0**	
5		**0**	**0**	**0**	**0**	
6						

As bordas da placa metálica são submetidas a fontes de temperaturas fixas de 0 e 100 graus. A temperatura no interior da área vai variar (subir) até um valor limite. Temos escolhido 4 pontos no interior da placa cujas temperaturas T1, T2, T3 e T4 devem ser calculadas. (Trabalhamos pelo momento com só 4 pontos, para poder explicar com mais facilidade o método a usar. O método de resolução desse problema consiste, na prática, em dividir a superfície em uma grade com um grande número de pontos.)

As temperaturas nas bordas não tem de ser 0, pode-se escolher qualquer outro valor.
O algoritmo consiste em calcular para cada ponto a média das temperaturas dos pontos da vizinhança.

T1 = (0+100+T2+T3)/4 = 25; no início, temos T2 = T3 = 0.
Este valor de T1 utilizamos já para calcular T2:
T2 = (T1+100+0+T4)/4 = (25+100+0+T4)/4 = 31,25
T3 = (0+T1+T4+0)/4 = (0+25+0+0)/4 = 6,25
T4 = (T4+T2+0+0)/4 = (6,25+31,25+0+0) = 9,375

Os valores de T1,..,T4 recalculamos (iteramos) até que se perceba claramente uma certa tendência (um valor limite). Fazemos uma segunda iteração com os mesmos valores de contorno:

T1 = (0+100+31,25+6,25)/4 = 34,375
T2 = (34,375+100+0+9,375)/4 = 35,9375
T3 = (0+34,375+9,375+0)/4 = 10,9375
T4 = (10,9375+35,9375+0+0)/4 = 11,71875

Agora vamos criar uma planilha. Primeiro, introduzimos os valores de contorno 0 e 100. Depois, escrevemos em B3 a fórmula =(A3+B2+C3+B4)/4. Excel vai anunciar que esta fórmula contém uma *referência circular*. (Quando uma fórmula volta a fazer referência à sua própria célula, tanto direta como *indiretamente*, este processo chama-se **referência circular**. Em nosso caso, queremos calcular B3 = (A3+B2+C3+B4)/4, mas C3 =(B3+C2+D3+C4)/4 e B4 =(A4+B3+C4+B5)/4 precisam o valor de B3, ou seja, eles referem-se de volta à B3.)

A referência circular é indicada por setas. Mas, você pode mover-se entre as células em uma referência circular clicando duas vezes nas setas. Em nosso caso, trata se de uma referência circular desejada. Para poder trabalhar com este "erro", temos que clicar em *Office>Excel Options* e eleger "Manual" e "Iteração". Cada Iteração efetua-se com F9. (Em 2003 vá a *Ferramentas> Opções>Cálculo* e escolha "Manual" e "Iteração".) A fórmula em B3 deve ser copiada até F7.
Depois de 24 iterações, aparece em D5 o valor de 25 graus exatos. Depois de mais 5 iterações, não haverá mais mudanças nos valores da planilha.

D13

	A	B	C	D	E	F	G	H
1								
2		100	100	100	100	100		
3	0	46,86869	62,92249	66,9425	62,92249	46,86869	0	
4	0	24,55225	37,87879	41,92502	37,87879	24,55225	0	
5	0	13,46154	22,11539	25	22,11539	13,46154	0	
6	0	7,178516	12,12121	13,84421	12,12121	7,178516	0	
7	0	3,131313	5,346737	6,134421	5,346737	3,131313	0	
8		0	0	0	0	0		
9								

A determinação da distribuição de temperatura numa superfície, conhecidas as temperaturas nas fronteiras, obedece à seguinte equação (equação de Laplace):

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (1)$$

ou também na forma

$$\nabla^2 T = 0$$

A equação de Laplace pertence as equações diferenciais parciais elípticas. A equação leva o seu nome em honra a Pierre-Simon de Laplace (1780).
No entanto, a equação apareceu pela primeira vez num artigo de Euler sobre hidrodinâmica em 1752.
A solução no caso de uma placa de 10*10 unidades de longitude é dada por

$$T(x,y)=\sum_{n}^{\infty}\frac{400}{n\pi senh(n\pi)}senh(\frac{n\pi}{10}(10-y))sen\frac{n\pi x}{10} \quad (2)$$

n= 1,3,5,...

O método de resolução desse problema consiste em dividir a superfície em uma grade de pontos, ou células, convertendo-o em um problema de diferenças finitas. Para calcular a série (2) com Excel, podemos preparar uma planilha da seguinte forma

H14 | =SOMA(G$5:G14)

	A	B	C	D	E	F	G	H	I
1				x=	1,3				
2				y=	2				
3									
4	**n**	**n*Pi**	**Senh1**	**Parentes.**	**Senh 2**	**Fator 3**	**Produto**	**Soma**	
5	1	3,141593	11,54874	2,513274	6,132141	0,397148	26,84973	26,84973	
6	3	9,424778	6195,824	7,539822	940,7476	0,940881	6,063139	32,91287	
7	5	15,70796	3317812	12,56637	143375,7	0,891007	0,980493	33,89336	
8	7	21,99115	1,78E+09	17,59292	21851315	0,278991	0,062413	33,95577	
9	9	28,27433	9,51E+11	22,61947	3,33E+09	-0,50904	-0,02521	33,93057	
10	11	34,55752	5,09E+14	27,64602	5,08E+11	-0,97592	-0,01125	33,91931	
11	13	40,8407	2,73E+17	32,67256	7,74E+13	-0,82708	-0,0023	33,91702	
12	15	47,12389	1,46E+20	37,69911	1,18E+16	-0,15643	-0,00011	33,91691	
13	17	53,40708	7,82E+22	42,72566	1,8E+18	0,612907	0,000105	33,91701	
14	19	59,69026	4,19E+25	47,75221	2,74E+20	0,995562	4,36E-05	33,91706	
15									

Soma-se 10 termos da série (2) para os valores x = 1,3 e y = 2. Os fatores ficam nas colunas A até G. Na coluna H formamos as somas parciais. Na célula H14 temos a soma total 33,91706.

Os valores de senh foram calculados com a fórmula

$$senh(x)=(e^{x}-e^{-x})/2 \quad (3)$$

Mas o Excel tem embutido a formula =SENH(), que houvéssemos podido usar em vez da formula (3), ou seja, =SENH(B5) em vez de =(EXP(B5)-EXP(-B5))/2 em C5.
Observe que em H5 fica =G5, mas em H6: =SOMA(G$5:G6) –copiar até H14.
A5: 1
A6: =A5+2
B5: =A5*PI()
D5: =B5*(10-E$5)/10
E5: =(EXP(D5)-EXP(-D5))/2, ou =SENH(D5)
F5: =SEN(B5*E$1/10)
G5: =400*E5*F5/(B5*C5), =primeira parte da soma
H5: =G5
H6: =SOMA(G$5:G6), = soma dos dois primeiros termos