

Capítulo 15

Resolução numérica de equações diferenciais

Para podermos investigar exemplos de **simulação** que surgem na Física, Engenharia, Biomatemática etc., estudamos, neste capítulo, alguns métodos de resolução numérica de equações diferenciais ordinárias de primeira e segunda ordens.

A solução de uma equação diferencial é uma *função* que satisfaz a equação diferencial sobre algum intervalo aberto. Uma equação diferencial *ordinária* tem a forma geral

$$\varphi(x, y, y', y'', y''', \dots, d^n y/dx^n) = 0 \quad (1)$$

Esta equação é de n-ésima ordem e tem somente uma variável independente, x.

A função $y = F(x)$ é uma solução de (1) se ela é n vezes diferenciável e se satisfaz a Eq. (1).

As equações $y' := dy/dx = x+y$; $y''+(1-y^2)y' + y = 0$ são exemplos de equações diferenciais ordinárias. Uma equação diferencial $\varphi(x, y(x), dy/dx) = 0$ pode geralmente ser escrita como

$$dy/dx = y' = f(x, y) \quad (2)$$

As equações diferenciais ordinárias têm várias soluções. Para escolher uma única solução, são necessárias informações adicionais, normalmente n para uma equação de n-ésima ordem. Se todas as n condições adicionais forem especificadas para um mesmo valor de x, por exemplo x_0 , temos um *Problema de Valor Inicial*, **PVI**. Caso estas n condições adicionais sejam dadas para mais de um valor de x, temos um *Problema de Valor de Contorno*, **PVC**.

O gráfico de uma solução da equação diferencial chama-se de *curva solução*. (Uma curva solução é também uma curva integral.)

Existem métodos gráficos e numéricos para obter uma idéia sobre a forma da solução, e aos quais pode-se recorrer se não existe nenhuma fórmula explícita da solução ou se a fórmula é complicada demais para ser útil.

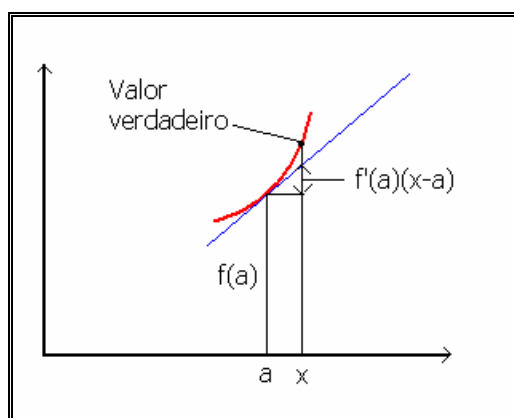
Para as aplicações na Física, Biomatemática etc. precisamos muitas vezes de **métodos numéricos** que aproximam uma solução exata com praticamente qualquer precisão.

Método de Euler para $y' = f(x,y)$

Vamos considerar agora a equação diferencial ordinária de primeira ordem $y' = f(x,y)$ junto com uma condição inicial $y(x_0) = y_0$.

O nosso objetivo é obter numericamente uma solução $y(x)$ que satisfaça a equação diferencial e as condições iniciais. O método numérico mais simples é o de Euler (1707-1783) e baseia-se na idéia de aproximar os valores de $y(x)$ pela reta tangente, como é ilustrado na figura a seguir.

(Euler descreveu o seu método em 1768 em "*Institutiones Calculi Integralis*, sectio secunda, caput VII")



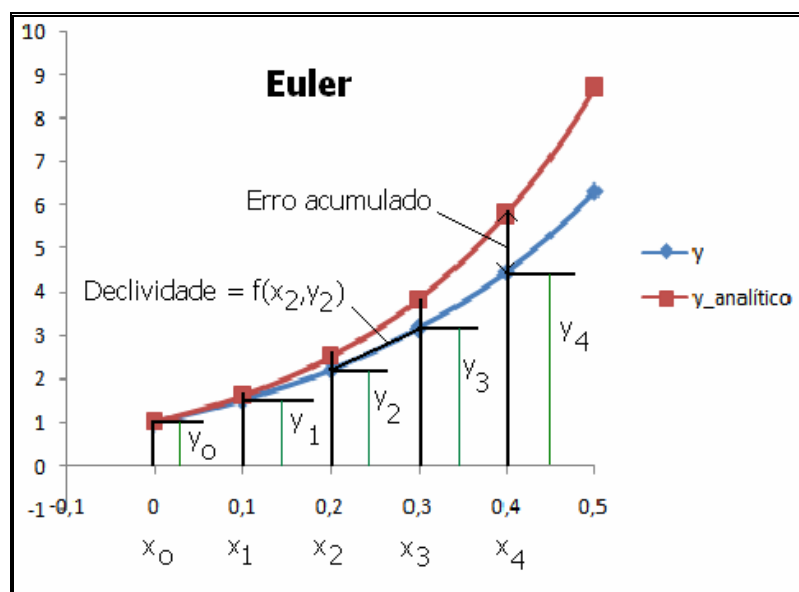
Se fizermos um zoom no gráfico de uma função lisa de uma variável $y = f(x)$ perto de um ponto $x = a$, o gráfico parece cada vez mais uma reta e assim se torna indistinguível de sua reta tangente nesse ponto. A inclinação da reta tangente é a derivada $f'(a)$ e a reta passa pelo ponto $(a, f(a))$ de modo que sua equação é

$$y = f(a) + f'(a)(x-a) \quad (3)$$

Agora aproximamos os valores de f pelos valores- y da reta tangente. Para valores de x próximos de a , podemos escrever para o verdadeiro valor $f(x)$ da função f no ponto x

$$f(x) \approx f(a) + f'(a)(x-a) \quad (4)$$

O traço da tangente até o ponto do valor verdadeiro indica o erro que fazemos aproximando $f(x)$ pelo valor de $f(a) + f'(a)(x-a)$. O fato de f ser aproximadamente uma função linear em x perto de a é expresso dizendo que f é *localmente linear* perto de $x = a$.



Vimos que o método de Euler se baseia na suposição que a reta tangente à curva solução (curva integral) de $y' = f(x,y)$ com $y(x_0) = y_0$ em $(x_i, y(x_i))$ aproxima a curva solução sobre o intervalo $[x_i, x_{i+1}]$.

Visto que a inclinação (ou declividade) da curva solução em $(x_i, y(x_i))$ é $y'(x_i) = f(x_i, y(x_i))$, a equação da reta tangente à curva integral em $(x_i, y(x_i))$ é

$$y = y(x_i) + f(x_i, y(x_i))(x-x_i) \quad (5)$$

Fazendo $x = x_{i+1} = x_i + h$, obtemos

$$y_{i+1} = y(x_i) + h \cdot f(x_i, y(x_i)) \quad (6)$$

sendo h o tamanho do passo e y_{i+1} o valor de y até a *reta tangente* no ponto x_{i+1} . y_{i+1} tomamos como uma aproximação a $y(x_{i+1})$.

Já que foi dado $y(x_0) = y_0$, podemos usar (6) com $i = 0$ para calcular y_1

$$y_1 = y(x_0) + h \cdot f(x_0, y(x_0)) = y_0 + h \cdot f(x_0, y_0) \quad (7)$$

Agora fazemos $i = 1$ e Eq. (6) se torna

$$y_2 = y(x_1) + h \cdot f(x_1, y(x_1)) \quad (8)$$

mas, esta equação não é útil, pois não conhecemos $y(x_1)$. (Só $y(x_0)$ está conhecido e o chamamos y_0 .) Bem, vem aqui a aproximação:

O valor $y(x_1)$, que não conhecemos, substituímos pelo valor y_1 , que só chega até a reta tangente e que é, no exemplo da figura anterior, nitidamente inferior ao valor real da função $y(x)$ em x_1 .

$$y(x_2) \approx y_2 = y_1 + h \cdot f(x_1, y_1)$$

No próximo passo vamos substituir $y(x_2)$ por y_2 :

$$y_3 = y_2 + h \cdot f(x_2, y_2)$$

O processo pode ser repetido até que o valor desejado de x seja alcançado.

Em geral, o método de Euler começa com o valor conhecido $y(x_0) = y_0$ e calcula y_1, y_2, \dots, y_n por meio da fórmula de recorrência

$$y_{i+1} = y_i + h \cdot f(x_i, y_i), \quad 0 \leq i \leq n - 1 \quad (9)$$

(Trata-se de uma fórmula de recorrência e não de iteração, pois no caso de uma iteração, que é um caso especial da recorrência, se busca, em geral, um valor limite para o processo. Mas, o uso das palavras neste sentido estrito não é muito comum.)

Os números y_1, y_2, y_3 etc. são aproximações de $y(x_1), y(x_2), y(x_3)$ etc.

Exemplo:

$y' = 1 - x + 4y$, $y(0) = 1$; solução exata: $y(x) = x/4 - 3/16 + (19/16) \cdot e^{4x}$

Solução segundo Euler: ($h = 0.1$)

$f(x, y) = 1 - x + 4y$

$y_1 = y_0 + h \cdot f(x_0, y_0) = 1 + 0.1 \cdot (1 - 0 + 4 \cdot 1) = 1 + 0.5 = 1,5$; $x = x_1 = h = 0.1$

valor exato (ou analítico): $y(0,1) = 1,609041828$

$y_2 = y_1 + h \cdot f(x_1, y_1) = 1,5 + 0,1 \cdot (1 - 0,1 + 4 \cdot 1,5) = 1,5 + 0,69 = 2,19$; $x = x_2 = 0,2$

valor exato: $y(0,2) = 2.505329853$

$y_3 = y_2 + h \cdot f(x_2, y_2) = 2,19 + 0,956 = 3,146$; $x = x_3 = 0,3$

valor exato: $y(0,3) = 3,830138846$

É fácil escrever um programa VBA para o método de Euler:

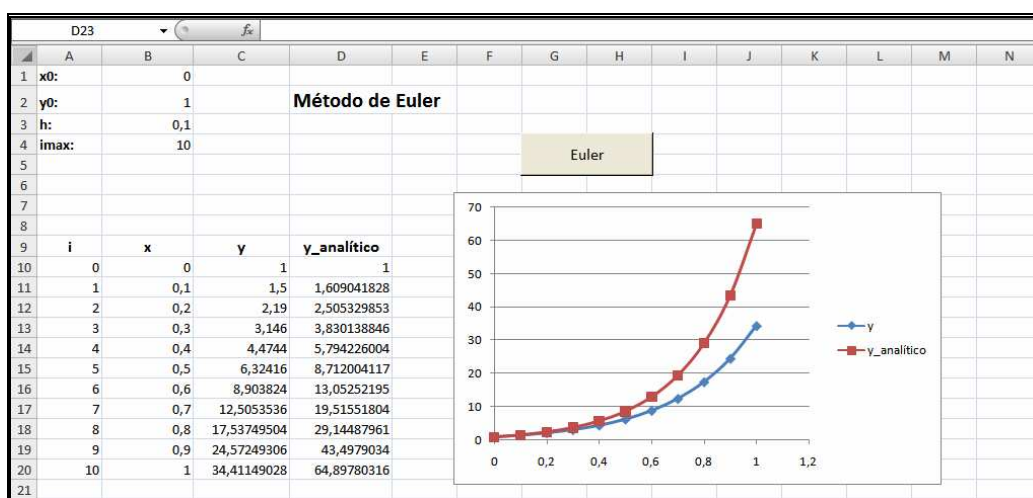
```
Sub Euler1()
    Range("A10:D200").Clear
    x = Cells(1, 2).Value
    y = Cells(2, 2).Value
    h = Cells(3, 2).Value
    imax = Cells(4, 2).Value
    Cells(10, 1).Value = 0
    Cells(10, 2).Value = x
    Cells(10, 3).Value = y
    Cells(10, 4).Value = FO(x, y) 'analítico

    For i = 1 To imax Step 1
        y = y + h * F(x, y)
        x = x + h
        Cells(10 + i, 1).Value = i
        Cells(10 + i, 2).Value = x
        Cells(10 + i, 3).Value = y
        Cells(10 + i, 4).Value = FO(x, y)
    Next i
End Sub
```

Formulamos as duas funções F e FO da seguinte maneira:

```
Function F(x, y)
  F = 1 - x + 4 * y
End Function
Function FO(x, y) 'analítico
  FO = x / 4 - 3 / 16 + (19 / 16) * Exp(4 * x)
End Function
```

Na seguinte planilha, vemos os cálculos anteriores estendidos até x = 1.



Aplicação: Modelo de crescimento logístico

O primeiro exemplo trata do crescimento de uma "população" de árvores, animais, palavras, armas etc.

N_0 é o tamanho da população no início do estudo, $N(t)$ é o tamanho no tempo t . Num primeiro momento, poderíamos supor que a velocidade de crescimento (= taxa de crescimento) dN/dt fosse proporcional ao tamanho atual $N(t)$, ou seja, poderíamos tentar usar a seguinte expressão:

$$dN(t)/dt = aN(t) \quad (10)$$

sendo a o coeficiente do crescimento. Facilmente podemos ver que esta equação diferencial de primeira ordem tem como solução a seguinte função exponencial

$$N(t) = N_0 \cdot e^{a(t-t_0)} \quad (11)$$

que descreve um crescimento sem limite. No máximo ao começo do processo podemos supor um crescimento exponencial, pois, após de certo tempo, devemos observar um processo de frenagem devido a influencias externas (por exemplo por falta de alimento). (Ainda é possível descrever aproximadamente o crescimento populacional mundial pela eq. (11), mas, o ritmo máximo de crescimento da população mundial foi atingido por volta da segunda metade da década de 1960.)

O matemático belga Pierre F. Verhulst introduzia em 1837 um termo de frenagem na equação (10) e propôs o seguinte modelo:

$$dN(t)/dt = aN(t) - bN(t)^2 \quad (12)$$

b = capacidade de suporte ambiental (a e b são fatores positivos). Esta equação tem uma solução analítica:

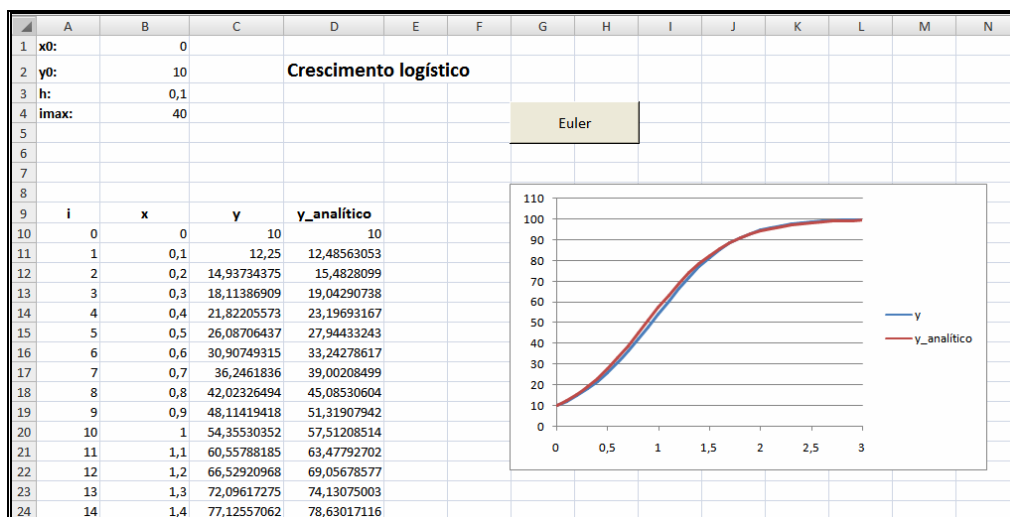
$$N(t) = \frac{a}{b\left(1 + \frac{a - bN_0}{bN_0} e^{-at}\right)} \quad (13)$$

A *função logística*, expressa pela equação (13), costuma também ser designada como lei universal do crescimento. Sua aplicabilidade como ferramenta matemática para a descrição do crescimento de populações *em geral* foi demonstrada nos anos 20 pelo estadístico e zoólogo americano Raymond Pearl (1925), razão pela qual a equação logística é, às vezes, referida como equação de Pearl. O nosso objetivo é determinar a solução da equação (12) numericamente por meio do método de Euler. Podemos facilmente adaptar o nosso programa à nova situação, identificando x com t e y com $N(t)$:

```
Sub Euler1()
  Range("A10:D200").Clear
  x = Cells(1, 2).Value
  y = Cells(2, 2).Value
  h = Cells(3, 2).Value
  imax = Cells(4, 2).Value
  a = 2.5
  b = 0.025
  y0 = y
  Cells(10, 1).Value = 0
  Cells(10, 2).Value = x
  Cells(10, 3).Value = y
  Cells(10, 4).Value = FO(a, b, x, y) 'analítico
  For i = 1 To imax Step 1
    y = y + h * F(a, b, y)
    x = x + h
    Cells(10 + i, 1).Value = i
    Cells(10 + i, 2).Value = x
    Cells(10 + i, 3).Value = y
    Cells(10 + i, 4).Value = FO(a, b, x, y0)
  Next i
End Sub

Function F(a, b, y)
  F = a * y - b * y ^ 2
End Function

Function FO(a, b, x, y0) 'analítico
  FO = a / (b * (1 + ((a - b * y0) / (b * y0)) * Exp(-a * x)))
End Function
```



A concordância entre as soluções numéricas e analíticas é satisfatória. As curvas logísticas em S (curva sigmóide) constituem hoje em dia uma das mais importantes ferramentas matemáticas para a prática quantitativa da *previsão tecnológica*, isto é, para a avaliação do potencial de crescimento e difusão de novas tecnologias.

Embora o método de Euler seja bastante simples, o mesmo é pouco utilizado para a solução numérica do problema do valor inicial, já que há outros métodos, como veremos adiante, que possuem uma melhor eficiência e exatidão. Os outros métodos são mais complicados, mas, na maioria dos casos, podemos seguir utilizando o nosso esquema básico que aplicamos acima.

Métodos melhorados de Euler

O método de Euler na formulação $y_{i+1} = y_i + h \cdot f(x_i, y_i)$ utiliza sempre a inclinação da reta tangente à curva solução no começo do intervalo $[x_i, x_{i+1}]$ e supõe que esta inclinação permaneça constante durante o intervalo inteiro. Mas, normalmente, vemos que a curva solução de $y(x)$ muda a inclinação da reta tangente no intervalo $[x_i, x_{i+1}]$. Pode-se esperar um melhoramento do método tomando a inclinação no centro do intervalo ou tomando uma média de vários inclinações em $[x_i, x_{i+1}]$.

Como **Método melhorado de Euler** ou **Método de Heun** (pron.: hoin) se conhece um procedimento que calcula primeiro a média das inclinações das retas tangentes à curva integral nos extremos do intervalo $[x_i, x_{i+1}]$, depois se segue os passos que nos levaram às fórmulas (5) até (9). Voltemos, então, à Eq. (5) e substituamos a inclinação $(f(x_i, y(x_i)))$ pela média

$$m_i = (f(x_i, y(x_i)) + f(x_{i+1}, y(x_{i+1}))) / 2 \quad (14)$$

A Eq. (6) reza agora $y_{i+1} = y(x_i) + h (f(x_i, y(x_i)) + f(x_{i+1}, y(x_{i+1}))) / 2$ e é uma aproximação a $y(x_{i+1})$. Como antes, aproximamos $y(x_i)$ por seu valor aproximado y_i quando $i > 0$.

Pero $y(x_{i+1})$ também será, normalmente, desconhecido e vamos substituí-lo pela aproximação $y(x_{i+1}) \approx y_{i+1} = y_i + h f(x_i, y_i)$.

A fórmula de recorrência do *método melhorado de Euler* ou a *fórmula de Heun* fica finalmente assim

$$y_{i+1} = y_i + h/2 \cdot (f(x_i, y_i) + f(x_{i+1}, y_i + h f(x_i, y_i))) \quad (15)$$

Para o cálculo prático é útil a introdução das seguintes expressões

$$k_{1i} = f(x_i, y_i) \quad (16)$$

$$k_{2i} = f(x_i + h, y_i + h k_{1i}) \quad (17)$$

$$y_{i+1} = y_i + h (k_{1i} + k_{2i})/2 \quad (18)$$

O método de Heun produz resultados bastante mais exatos do que o método de Euler simples, especialmente uma variante del processo que melhora em cada ponto x_i primeiro os resultados por meio de uma *iteração interna* antes de avançar ao próximo ponto x_{i+1} . No programa de "Heun2 " realizamos isso.

(O método de Heun possui um erro de truncamento global da ordem de $O(h^2)$ e podemos verificar que, se diminuirmos o tamanho do passo de h para $h/2$, o erro será reduzido de um fator $1/4$, e assim sucessivamente. O símbolo $O(h^2)$ quer dizer que o método de Heun coincide com a série de Taylor até os termos de ordem h^2 .)

D16		f _x						
	A	B	C	D	E	F	G	H
1	x0:	0						
2	y0:	1		Método de Heun simples				
3	h:	0,1						
4	imax:	5		Heun1				
5								
6								
7								
8								
9	i	x	y	y_analítico	Fo - y			
10	0	0	1	1				
11	1	0,1	0,820040937	0,818751221	-0,00129			
12	2	0,2	0,672734445	0,670588174	-0,00215			
13	3	0,3	0,552597643	0,54992298	-0,00267			
14	4	0,4	0,455160637	0,452204669	-0,00296			
15	5	0,5	0,376681251	0,373627557	-0,00305			
16								

Tomamos as equações para esta planilha do seguinte programa.


```

Sub Heun2 ()
  Range("A10:E200").Clear
  x = Cells(1, 2).Value
  y = Cells(2, 2).Value
  h = Cells(3, 2).Value
  imax = Cells(4, 2).Value

  Cells(10, 1).Value = 0
  Cells(10, 2).Value = x
  Cells(10, 3).Value = y
  Cells(10, 4).Value = FO(x) 'analítico

  For i = 1 To imax Step 1
    ya = y 'valor antigo de y
    k1 = F(x, y)
    x = x + h
    ye = y + h * k1 ' valor Euler
    For j = 1 To 4 Step 1 '4 iterações internas
      k2 = F(x, ye)
      dy = h * (k1 + k2) / 2
      y = ya + dy
      ye = y
    Next j
    ya = y
    Cells(10 + i, 1).Value = i
    Cells(10 + i, 2).Value = x
    Cells(10 + i, 3).Value = y
    Cells(10 + i, 4).Value = FO(x)
    Cells(10 + i, 5).Value = FO(x) - y
  Next i
End Sub

Function F(x, y)
  F = -2 * y + x ^ 3 * Exp(-2 * x)
End Function

Function FO(x) 'analítico
  FO = Exp(-2 * x) * (x ^ 4 + 4) / 4
End Function

```

O método de Runge-Kutta

Mas, tampouco este método pode concorrer com o método clássico de Runge e Kutta, como veremos em seguida.

O método de Runge-Kutta é o rei entre aqueles métodos apropriados para resolver os problemas de valor inicial (C. Runge 1856-1927, W. Kutta 1867-1944). Os seus atrativos são simplicidade, alta precisão e versatilidade. A idéia detrás do método RK é bastante parecido ao raciocínio detrás do *método melhorado de Euler* (fórmula de Heun), mas agora calculamos a função $f(x,y)$ não apenas duas vezes, como no método de Heun, antes quatro vezes, reduzindo assim o erro global de truncamento para $O(h^4)$! (O método de RK, que mais adiante vamos usar, é também conhecido como RK de quarta ordem. O método de Heun é chamado de RK de segunda ordem e o método de Euler como RK de primeira ordem.)

Não vamos desenvolver rigorosamente as fórmulas do método RK. Vou, porém, apresentar o algoritmo como se fosse uma simples modificação do método de Euler.

Algoritmo Runge-Kutta de quarta ordem para $y' = f(t,y)$.

(Já que, nas aplicações, temos geralmente o tempo como variável independente, utilizaremos t em vez de x e v em vez de y' . O símbolo $\langle v \rangle$ indica o valor médio de 4 derivadas -velocidades- do método de RK.)

$$t_{n+1} = t_n + h$$

$$y_{n+1} = y_n + h\langle v \rangle,$$

onde

$$\langle v \rangle := (v_1 + 2v_2 + 2v_3 + v_4)/6 \quad (19)$$

Calculam-se as quatro derivadas segundo o seguinte esquema:

$$v_1 := f(t, y)$$

$$v_2 := f(t + h/2, y + v_1 \cdot h/2)$$

$$v_3 := f(t + h/2, y + v_2 \cdot h/2)$$

$$v_4 := f(t + h, y + v_3 \cdot h) \quad (20)$$

(Mais adiante veremos que a generalização deste método para equações de segunda ordem, $y'' = f(t, x, y')$, é muito simples.)

	A	B	C	D	E	F	G	H
1	x0:	0						
2	y0:	1		Método de Runge-Kutta $y'(x,y)$				
3	h:	0,1						
4	imax:	5		Runge_Kutta1				
5								
6								
7								
8								
9	i	x	y	y_analítico	F0-y			
10	0	0	1	1				
11	1	0,1	1,222101667	1,222104137	2,47E-06			
12	2	0,2	1,497730642	1,497737046	6,4E-06			
13	3	0,3	1,843165873	1,843178201	1,23E-05			
14	4	0,4	2,278290464	2,278311393	2,09E-05			
15	5	0,5	2,82738964	2,827422743	3,31E-05			
16								

Agora vamos ver como é simples a implementação computacional das esquemas (19) e (20).

(Para mostrar que o novo programa, também, é uma modificação do programa de Euler, seguimos utilizando, no código, x em vez de t.)

```

Sub Runge_Kutta1()
  Range("A10:E200").Clear
  x = Cells(1, 2).Value
  y = Cells(2, 2).Value
  h = Cells(3, 2).Value
  imax = Cells(4, 2).Value

  Cells(10, 1).Value = 0
  Cells(10, 2).Value = x ' = t nas aplicações
  Cells(10, 3).Value = y
  Cells(10, 4).Value = FO(x) 'analítico

  For i = 1 To imax Step 1
    v1 = F(x, y)
    v2 = F(x + h / 2, y + v1 * h / 2)
    v3 = F(x + h / 2, y + v2 * h / 2)
    v4 = F(x + h, y + v3 * h)
    y = y + h * (v1 + 2 * v2 + 2 * v3 + v4) / 6
    x = x + h
    Cells(10 + i, 1).Value = i
    Cells(10 + i, 2).Value = x
    Cells(10 + i, 3).Value = y
    Cells(10 + i, 4).Value = FO(x)
    Cells(10 + i, 5).Value = FO(x) - y
  Next i
End Sub

Function F(x, y)
  F = 2 * (x ^ 2 + y)
End Function

Function FO(x) 'analítico
  FO = 1.5 * Exp(2 * x) - x ^ 2 - x - 0.5
End Function

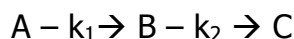
```

Os resultados mostram que os erros são apenas da ordem de 10^{-6} . No caso do método de Euler, encontramos erros da ordem de 10^{-2} . O método de Euler tem, porem, o seu valor didático e ajuda no entendimento dos métodos mais exatos.

Aplicação: Desintegração radioativa

A desintegração é, de certo modo, o processo inverso do crescimento exponencial. Uma substância radioativa consista no momento $t = 0$ de um número $A(0) := A_0$ de átomos radioativos. Depois do tempo t seja desintegrado um certo número de átomos da substancia-mãe A , que, por sua vez, produzem uma substancia-filha B , que, por sua vez, decai produzindo C .

A cinética da reação pode ser simbolizada na seguinte forma



A, B, C são as concentrações dos participantes da reação, k_1 e k_2 são as velocidades (taxas) da reação. No caso da desintegração radioativa, chama-se estas taxas λ_A e λ_B . λ_A é a constante de desintegração da substância-mãe. O número dos átomos caindo na unidade do tempo é proporcional à quantidade dos átomos não decaída:

$$\frac{dA(t)}{dt} = -\lambda_A A(t) \quad (21)$$

Enquanto B decai, ela recebe permanentemente as partículas que vem da substância-mãe A. Ou seja:

$$\frac{dB(t)}{dt} = -\lambda_B B(t) + \lambda_A A(t) \quad (22)$$

As equações diferenciais (21) e (22) descrevem, juntas, a desintegração de A e B e devem ser resolvidas juntamente. Trata-se de um sistema de duas equações acopladas de primeira ordem. No seguinte programa "Euler2" foi só preciso introduzir o tempo t e uma segunda função G(x,y) para dB(t)/dt:

```
Sub Euler2()
  Range("A10:D200").Clear
  t = Cells(1, 2).Value
  x = Cells(2, 2).Value
  y = Cells(3, 2).Value
  h = Cells(4, 2).Value
  imax = Cells(5, 2).Value

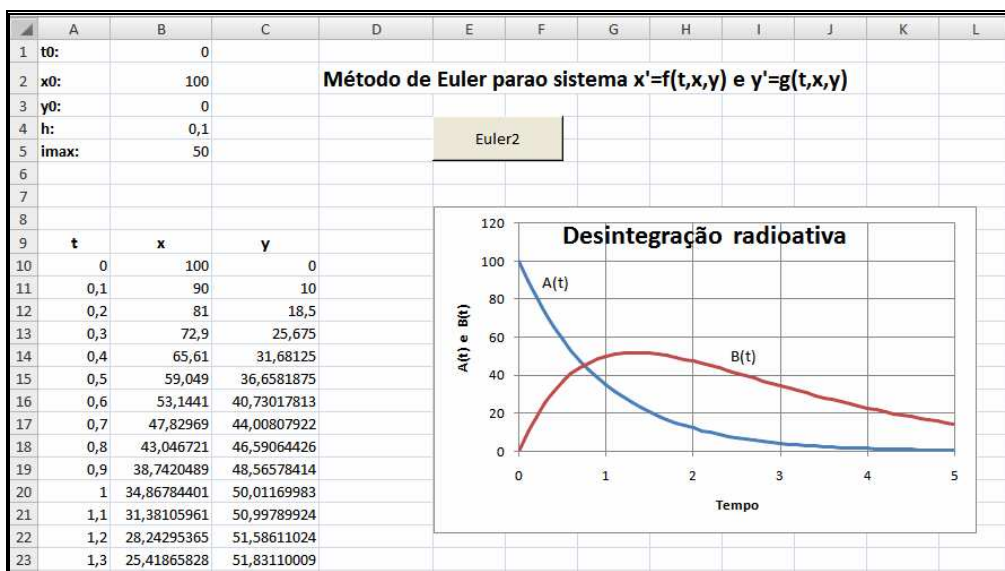
  Cells(10, 1).Value = t
  Cells(10, 2).Value = x
  Cells(10, 3).Value = y

  For i = 1 To imax Step 1
    F1 = F(x)
    G1 = G(x, y)
    x = x + h * F1
    y = y + h * G1
    t = t + h
    Cells(10 + i, 1).Value = t
    Cells(10 + i, 2).Value = x
    Cells(10 + i, 3).Value = y
  Next i
End Sub

Function F(x)
  F = -1 * x
End Function

Function G(x, y)
  G = -0.5 * y + 1 * x
End Function
```

Na planilha, vemos o processo de desintegração radiativo para as constantes $\lambda_A = 1$ e $\lambda_B = 0,5$. Os valores iniciais ficam em B0 até B3. Para $x_0 = A_0$ temos o valor 100, para $y_0 = B_0 = 0$. ($x := A$ e $y := B$).

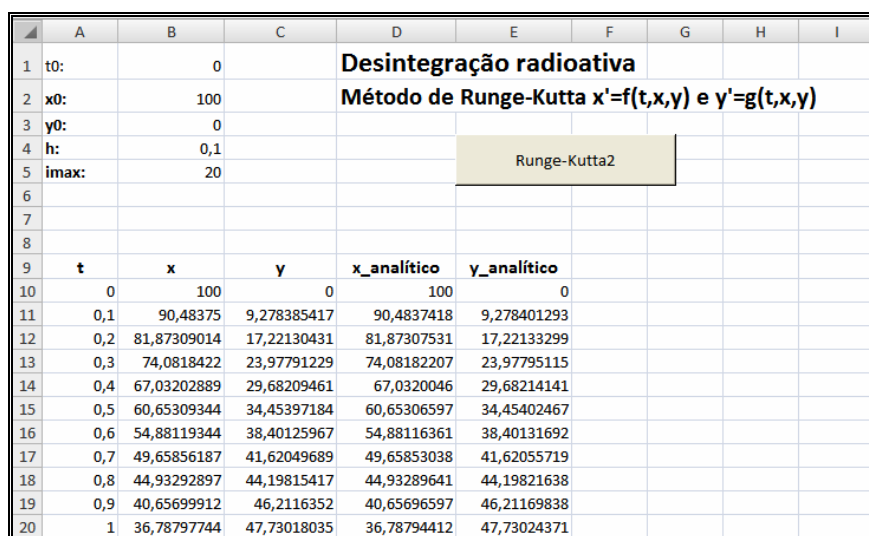


Podemos controlar a solução numérica (Euler) com os valores da solução analítica:

$$A(t) = A_0 e^{-\lambda_A t}$$

$$B(t) = \frac{A_0 \lambda_A}{\lambda_B - \lambda_A} (e^{-\lambda_A t} - e^{-\lambda_B t}) \quad (23)$$

A comparação mostra que a concordância para $h=0,1$ não é muito satisfatória e que é preferível utilizar um método mais exato, como, por exemplo, o método de Runge – Kutta. A planilha seguinte mostra uma excelente concordância entre as soluções numéricas e analíticas.



Para produzir esta planilha, foi necessária modificar o algoritmo RK1 para resolver nosso sistema de dois equações de primeira ordem $x' = f(t,x,y)$ e $y' = g(t,x,y)$. Em última análise, foi somente necessário introduzir uma segunda equação, $g(t,x,y)$. Com o intuito de aplicar o programa em outras situações na Física, utilizamos os símbolos v (velocidade) e a (aceleração) para x' e y' .

```
Sub Runge_Kutta2()
Range("A10:E200").Clear
t = Cells(1, 2).Value
x = Cells(2, 2).Value
y = Cells(3, 2).Value
h = Cells(4, 2).Value
imax = Cells(5, 2).Value

Cells(10, 1).Value = t
Cells(10, 2).Value = x
Cells(10, 3).Value = y
Cells(10, 4).Value = FO(t) 'x_analítico
Cells(10, 5).Value = GO(t) 'y_analítico

For i = 1 To imax Step 1
v1 = F(t, x, y)
a1 = G(t, x, y)
v2 = F(t + h / 2, x + v1 * h / 2, y + a1 * h / 2)
a2 = G(t + h / 2, x + v1 * h / 2, y + a1 * h / 2)
v3 = F(t + h / 2, x + v2 * h / 2, y + a2 * h / 2)
a3 = G(t + h / 2, x + v2 * h / 2, y + a2 * h / 2)
v4 = F(t + h, x + v3 * h, y + a3 * h)
a4 = G(t + h, x + v3 * h, y + a3 * h)
x = x + h * (v1 + 2 * v2 + 2 * v3 + v4) / 6
y = y + h * (a1 + 2 * a2 + 2 * a3 + a4) / 6
t = t + h
Cells(10 + i, 1).Value = t
Cells(10 + i, 2).Value = x
Cells(10 + i, 3).Value = y
Cells(10 + i, 4).Value = FO(t)
Cells(10 + i, 5).Value = GO(t)

Next i
End Sub
```

As funções são:

```
Function F(t, x, y)
F = -1 * x
End Function
Function G(t, x, y)
G = -0.5 * y + 1 * x
End Function
Function FO(t) 'analítico
FO = 100 * Exp(-1 * t)
End Function
Function GO(t) 'analítico
GO = 100 * (Exp(-1 * t) - Exp(-0.5 * t)) / (0.5 - 1)
End Function
```

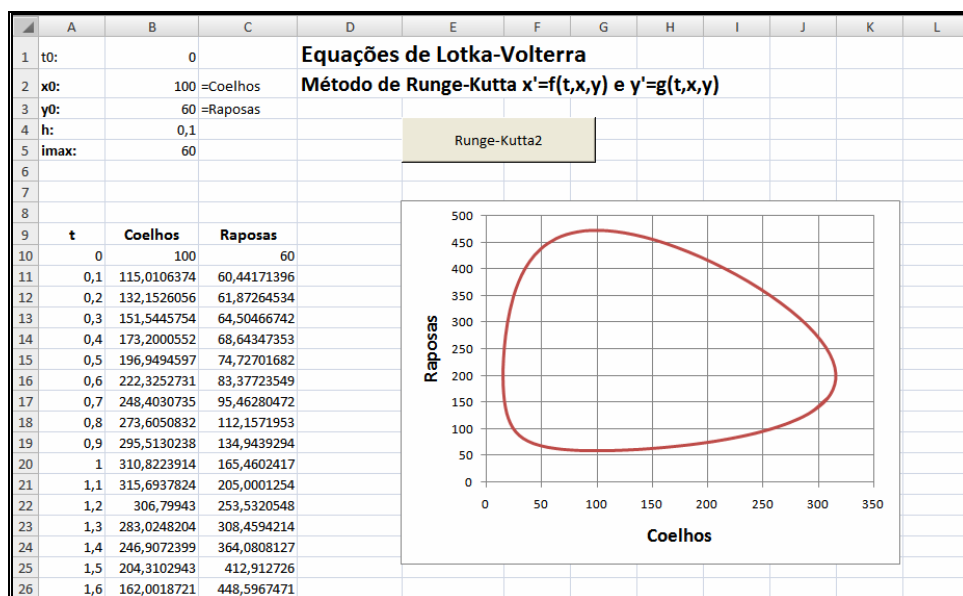
Aplicação: Equações de Lotka e Volterra

O modelo de presa-predador de **Lotka-Volterra** é um modelo de importância histórica na modelagem matemática de sistemas ecológicos como, por exemplo, na análise da coexistência de duas espécies que interagem, uma como presa (coelhos) e outra como predadora (raposas). A.J. Lotka (1925) e V. Volterra (1926) propuseram independentemente o seguinte modelo para calcular a evolução do número de coelhos e raposas num determinado ecossistema, ano após ano:

$$x' = ax - bxy \text{ e } y' = -cy + dxy \quad (24)$$

y = número de raposas, x = número de coelhos, t = tempo de interação (coexistência). Os parâmetros a , b , c , d representam a interação entre as duas espécies.

As equações de **Lotka-Volterra** são um par de equações diferenciais, não lineares e de primeira ordem, que podemos facilmente resolver utilizando o nosso programa "Runge_Kutta2".




```

Sub Runge_Kutta2()
Range("A10:E200").Clear
t = Cells(1, 2).Value
x = Cells(2, 2).Value
y = Cells(3, 2).Value
h = Cells(4, 2).Value
imax = Cells(5, 2).Value

Cells(10, 1).Value = t
Cells(10, 2).Value = x
Cells(10, 3).Value = y

For i = 1 To imax Step 1
v1 = F(t, x, y)
a1 = G(t, x, y)
v2 = F(t + h / 2, x + v1 * h / 2, y + a1 * h / 2)
a2 = G(t + h / 2, x + v1 * h / 2, y + a1 * h / 2)
v3 = F(t + h / 2, x + v2 * h / 2, y + a2 * h / 2)
a3 = G(t + h / 2, x + v2 * h / 2, y + a2 * h / 2)
v4 = F(t + h, x + v3 * h, y + a3 * h)
a4 = G(t + h, x + v3 * h, y + a3 * h)
x = x + h * (v1 + 2 * v2 + 2 * v3 + v4) / 6
y = y + h * (a1 + 2 * a2 + 2 * a3 + a4) / 6
t = t + h
Cells(10 + i, 1).Value = t
Cells(10 + i, 2).Value = x
Cells(10 + i, 3).Value = y

Next i
End Sub

Function F(t, x, y)
F = 2 * x - 0.01 * x * y
End Function

Function G(t, x, y)
G = -1 * y + 0.01 * x * y
End Function

Function FO(t) 'analítico
FO = 100 * Exp(-1 * t)
End Function

Function GO(t) 'analítico
GO = 100 * (Exp(-1 * t) - Exp(-0.5 * t)) / (0.5 - 1)
End Function

```