

Resolução Numérica de Equações Diferenciais Ordinárias

Diogo Pinheiro Fernandes Pedrosa

Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia de Computação e Automação
<http://www.dca.ufrn.br/>

1 Introdução

Uma *Equação Diferencial* é uma equação que envolve derivadas de uma ou mais funções. Elas servem para descrever o comportamento de sistemas dinâmicos e possuem enorme aplicação em áreas como engenharia (comportamento de um circuito elétrico ou do movimento oscilatório de estruturas), biologia (crescimento de populações de bactérias) ou economia (aplicações financeiras). Elas são classificadas de acordo com o seu tipo, ordem e grau. Se uma equação diferencial envolve derivadas de uma função de uma única variável independente, ela é dita ser *Equação Diferencial Ordinária*. Caso a equação diferencial envolva as derivadas parciais de uma função de duas ou mais variáveis independentes, ela é uma *Equação Diferencial Parcial*.

Uma equação diferencial ordinária (ou E.D.O.) de ordem n pode ser expressa na seguinte forma:

$$\frac{d^n y}{dx^n} = G\left(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \dots, \frac{d^{n-1} y}{dx^{n-1}}\right) \quad (1)$$

onde x é a variável independente, y é uma função desta variável independente e $\frac{d^k y}{dx^k}$, com $k = 1, 2, \dots, n$ são as derivadas de y em relação a x .

O problema a ser tratado em Equações Diferenciais Ordinárias consiste em encontrar uma função y (ou solução) que satisfaça a equação 1. Esta solução é uma função que não possui derivadas nem diferenciais e ela pode ser uma solução geral ou particular. Uma solução geral de uma E.D.O. de ordem n é uma solução contendo n constantes de integração independentes e arbitrárias, como:

$$f(y) = f(x) + c_1 + c_2 + \dots + c_n$$

Uma solução particular é obtida a partir da solução geral, dando-se valores específicos às constantes. Frequentemente são dadas as seguintes condições que permitem encontrar os valores das constantes obtidas pelas integrações:

$$\begin{aligned} y(x_0) &= y_0 \\ y'(x_1) &= y_1 \\ y''(x_2) &= y_2 \\ &\vdots \\ y^{(n-1)}(x_{n-1}) &= y_{n-1} \end{aligned}$$

Nestes casos, se $x_0 = x_1 = x_2 = \dots = x_{n-1}$ então o problema é dito ser de valor inicial. Caso contrário, o problema é de valor de contorno.

De uma maneira geral, a busca de uma solução para uma equação diferencial ordinária com problema de valor inicial apresenta alguns problemas. Primeiro porque constata-se que os procedimentos para a busca de uma solução analítica não é trivial. O segundo problema é que, além disso, muitas questões práticas não possuem solução conhecida. Por fim, em muitos casos os coeficientes ou as funções existentes na equação diferencial são dados somente na forma de um conjunto tabelado de informações experimentais, o que torna impossível o uso de um procedimento analítico para determinar a solução da equação. Por isso há a necessidade do uso de métodos numéricos para equações diferenciais ordinárias.

Serão tratados aqui métodos numéricos para se conseguir os valores de $y(x)$ em pontos iniciais, ou seja, problemas de valor inicial. Tais problemas serão abordados em sua forma mais simples que são as equações diferenciais ordinárias de primeira ordem:

$$\begin{cases} \frac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases} \quad (2)$$

onde y_0 é um número dado. Os problemas de valor inicial (P.V.I.s) de ordem superior podem ser reduzidos a sistemas de primeira ordem através de variáveis auxiliares, o que permite a utilização dos métodos numéricos aqui apresentados.

Para resolver numericamente uma E.D.O. com P.V.I. (equação 2), supõem-se que ela satisfaz às condições de existência e unicidade. Esta solução numérica será encontrada para um conjunto finito de pontos (um intervalo fechado $[a, b]$) no eixo das abscissas.

Tomando-se m subintervalos deste intervalo $[a, b]$, sendo $m \geq 1$, é possível determinar $m + 1$ pontos onde as soluções numéricas devem ser calculadas. Estes pontos $x_j \in [a, b]$ são igualmente espaçados entre si por um fator h , onde $x_j = x_0 + j \cdot h$, sendo que $x_0 = a$, $x_m = b$, $h = (b - a)/m$ e $j = 0, 1, 2, \dots, m$.

O conjunto $\{x_0, x_1, x_2, \dots, x_m\}$ obtido denomina-se rede ou malha de $[a, b]$. A solução numérica é uma função linear por partes, ou seja, é aplicada a cada subintervalo, cujo gráfico apresenta-se como uma poligonal com vértices nos pontos (x_j, y_j) , sendo que y_j é calculado por algum método a ser apresentado. Embora existam métodos que apresentem uma boa precisão, os métodos numéricos sempre apresentarão erros quando comparados com as soluções exatas obtidas de uma solução analítica. A figura 1 ilustra como os erros afetam uma solução numérica.

Para facilitar a interpretação dos métodos, convencionou-se a seguinte notação:

- $y(x_j)$ é a solução exata do P.V.I., obtida analiticamente;
- y_j é a solução numérica.

Há três métodos principais que serão vistos na seguinte seqüência:

1. Método de Euler;
2. Métodos de Runge-Kutta; e
3. Métodos de Adams-Bashforth.

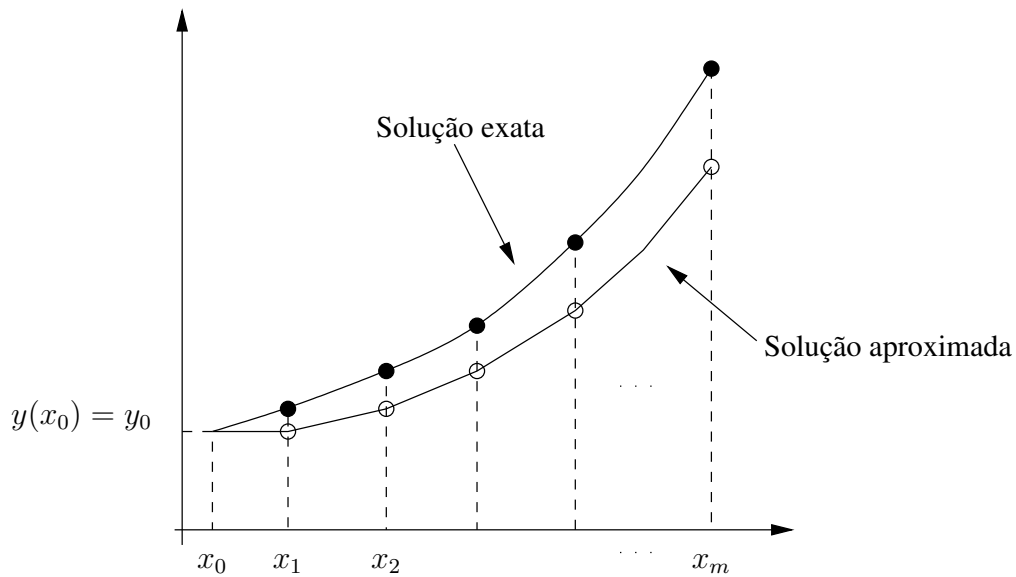


Figura 1: Comparação ilustrativa entre uma solução exata e uma aproximada de uma Equação Diferencial Ordinária

1.1 Série de Taylor de Função de uma Variável

Uma função f a uma variável x , contínua e indefinidamente derivável e aproximada em torno do ponto $x = a$, pode ser representada por uma série de potências da forma:

$$f(x) = f(a) + f'(a)\frac{(x-a)}{1!} + f''(a)\frac{(x-a)^2}{2!} + \dots + f^{(n)}(a)\frac{(x-a)^n}{n!} + \dots$$

Nas aplicações práticas da Série de Taylor, onde se utiliza métodos numéricos, não é possível computar todos os seus termos. O que se faz, então, é considerar apenas um número finito deles, truncando a série após o n -ésimo termo. Assim:

$$f(x) \cong f(a) + f'(a)\frac{(x-a)}{1!} + f''(a)\frac{(x-a)^2}{2!} + \dots + f^{(n-1)}(a)\frac{(x-a)^{n-1}}{(n-1)!} + R_n(x)$$

onde $R_n(x)$ é o erro de truncamento que pode ser expressado da seguinte forma:

$$R_n(x) = f^{(n)}(\xi)\frac{(x-a)^n}{n!}$$

com $a < \xi < x$.

2 Método de Euler

Seja uma Equação Diferencial Ordinária com Problema de Valor Inicial dada pela equação 2. O que se deseja é encontrar as aproximações y_1, y_2, \dots, y_m para as soluções exatas $y(x_1), y(x_2), \dots, y(x_m)$.

Sendo que o ponto inicial (x_0, y_0) é fornecido pelo problema, o primeiro passo então é a busca de y_1 . Para isto, aproximando-se a solução $y(x)$ por uma Série de Taylor no ponto $x = x_0$ e truncando no segundo termo, tem-se:

$$y(x) = y(x_0) + y'(x_0) \cdot (x - x_0)$$

Para $x = x_1$ tem-se:

$$y(x_1) = y(x_0) + (x_1 - x_0) \cdot y'(x_0)$$

Lembrando que, como os valores exatos $y(x_j)$ são desconhecidos, são utilizados os valores aproximados y_j e que $x_1 - x_0 = h$ e $y'(x_0) = f(x_0, y_0)$, onde h é a distância entre os pontos x_j , então:

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

Para encontrar y_2 na abscissa $x = x_2$ adota-se o mesmo procedimento. Assim, a solução aproximada é:

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

Como é possível verificar, pode-se aplicar a formulação de Taylor para todos os pontos dos subintervalos, o que permite definir uma regra geral para o Método de Euler:

$$y_{j+1} = y_j + h \cdot f(x_j, y_j)$$

com $j = 0, 1, 2, \dots, m - 1$.

Exemplo 1 Achar a solução numérica para:

$$\begin{cases} y' = x - y + 2 \\ y(0) = 2 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.1$.

Como é uma equação diferencial ordinária com problema de valor inicial então, pelas condições iniciais dadas: $x_0 = 0$ e $y_0 = 2$. Sendo o intervalo $[0, 1]$ e o espaçamento igual a 0.1, então a quantidade de subintervalos na malha é:

$$m = \frac{b - a}{h} = \frac{1 - 0}{0.1} = 10$$

o que implica em $j = 0, 1, 2, \dots, 9$.

Assim, para $j = 0$ tem-se:

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

É sabido que $y' = f(x, y) = x - y + 2$. Assim:

$$y_1 = y_0 + h \cdot (x_0 - y_0 + 2)$$

Substituindo os valores tem-se:

$$y_1 = 2 + 0.1 \cdot (0 - 2 + 2) = 2$$

Para $j = 1$, calcula-se primeiramente o x_1 :

$$\begin{aligned} x_j &= x_0 + j \cdot h \\ x_1 &= 0 + 1 \cdot 0.1 \\ x_1 &= 0.1 \end{aligned}$$

para, em seguida, encontrar a aproximação y_2 :

$$\begin{aligned}y_2 &= y_1 + h \cdot (x_1 - y_1 + 2) \\y_2 &= 2 + 0.1 \cdot (0.1 - 2 + 2) \\y_2 &= 2.01\end{aligned}$$

Os cálculos prosseguem com este mesmo procedimento até $j = 9$. Como o trabalho apresenta-se longo e rotineiro, torna-se viável a elaboração de um programa computacional para resolvê-lo. Assim, a listagem mostrada no apêndice A.1 apresenta o programa de resolução deste exemplo implementado em linguagem C.

O resultado apresentado por este programa foi:

```
Digite o valor do x inicial:
0
Digite o valor do y inicial:
2
Digite o valor do espacamento h:
0.1
Digite o número de subintervalos m:
10
Os valores de x e y sao:
0.000000, 2.000000;
0.100000, 2.000000;
0.200000, 2.010000;
0.300000, 2.029000;
0.400000, 2.056100;
0.500000, 2.090490;
0.600000, 2.131441;
0.700000, 2.178297;
0.800000, 2.230467;
0.900000, 2.287421;
1.000000, 2.348679;
```

A solução analítica para esse problema é $y(x) = e^{-x} + x + 1$. Para visualizar o seu comportamento pode-se utilizar o Scilab. Assim, definindo-se a malha do intervalo $[a, b]$:

```
--> x = [0:0.1:1];
```

e calculando-se o valor exato de $y(x)$ por:

```
-->for i=1:11, y(i) = exp(-x(i))+x(i)+1; end;
```

uma vez que serão encontrados $m + 1$ pontos, o resultado obtido é:

```
-->y
y =

!   2.          !
!  2.0048374   !
!  2.0187308   !
!  2.0408182   !
```

```
! 2.07032 !
! 2.1065307 !
! 2.1488116 !
! 2.1965853 !
! 2.249329 !
! 2.3065697 !
! 2.3678794 !
```

e, por fim, para contruir o gráfico no Scilab basta digitar na linha de comando:

```
-->plot2d(x,y)
```

A figura 2 mostra uma comparação gráfica entre o resultado numérico obtido pelo método de Euler (através do programa escrito em C) e o resultado analítico (calculado no Scilab).

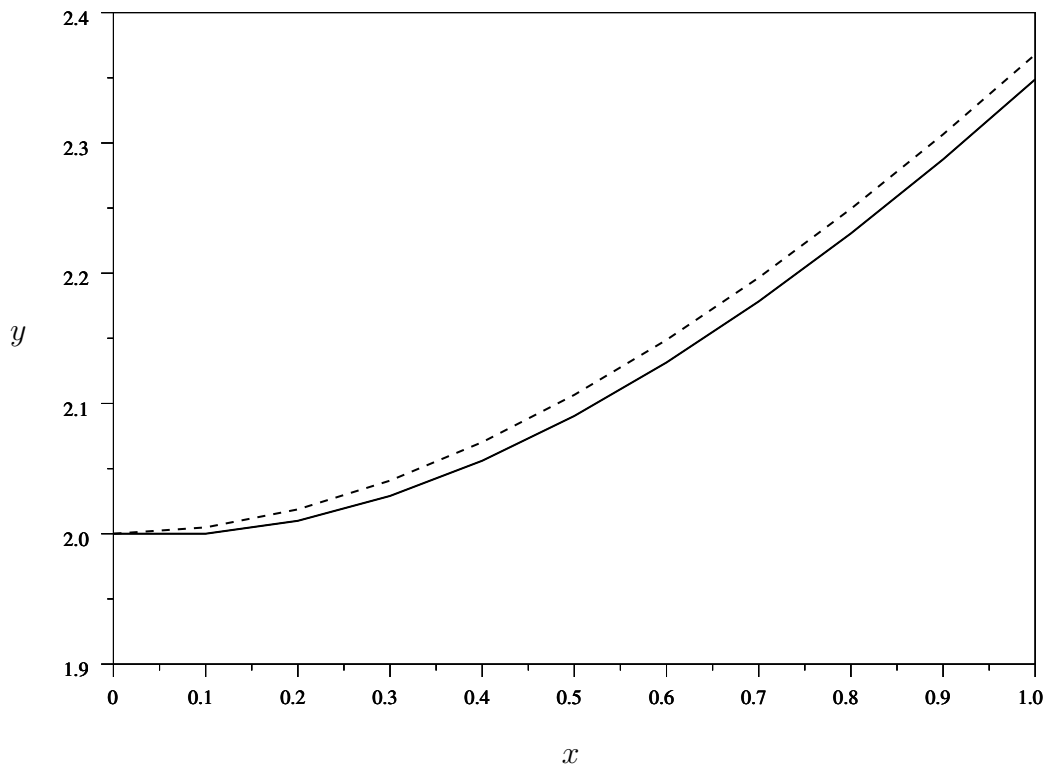


Figura 2: Comparação entre o resultado analítico (linha tracejada) e o Método de Euler (linha sólida).

Embora seja simples, o Método de Euler tem o problema de apresentar erros, uma vez que as aproximações para o valor exato $y(x_j)$ são dependentes de aproximações anteriores. De certa forma, é possível ter uma idéia deste erro. Como foi utilizado a Série de Taylor para deduzir o Método de Euler, então o erro de truncamento é dado por:

$$e_{j+1} = R_2(x_j) = y''(\xi) \frac{h^2}{2!}$$

com $x_j < \xi < x_{j+1}$. Este erro é chamado de *erro local de truncamento*. Nota-se que ele depende do espaçamento h .

Dessa forma, pode-se afirmar que o Método de Euler sofre com o erro local de truncamento citado e pelo fator de propagação deste erro, uma vez que as aproximações, em cada iteração, dependem dos valores das aproximações calculadas anteriormente.

Uma maneira de reduzir estes erros é diminuir o valor do espaçamento h , porém, dependendo da aplicação, isto aumentaria o esforço computacional. Outra alternativa é utilizar a seguinte regra para o Método de Euler:

$$y_{j+1} = y_j + h \cdot \phi(x_j, y_j, h)$$

com $j = 0, 1, 2, \dots, m - 1$.

Esta regra diz que no lugar da função $y' = f(x, y)$ deve-se utilizar uma outra função $\phi(x, y, h)$ que envolva também o espaçamento h . Como a Série de Taylor foi utilizada para definir o Método de Euler então, para encontrar esta função $\phi(\cdot)$, basta apenas expressar a aproximação y_{j+1} utilizando três termos da Série. Isto resulta na seguinte fórmula:

$$y_{j+1} = y_j + h \cdot y'(x_j) + \frac{h^2}{2} \cdot y''(x_j)$$

Sendo $y'(x_j) = f(x_j, y_j)$, então:

$$y''(x_j) = \frac{\partial}{\partial x} f(x_j, y_j) + f(x_j, y_j) \cdot \frac{\partial}{\partial y} f(x_j, y_j)$$

Para encontrar a solução basta aplicar o método normalmente utilizando esta fórmula apresentada. Resolvendo o exemplo anterior por este método, tem-se como resultado:

Digite o valor do x inicial:

0

Digite o valor do y inicial:

2

Digite o valor do espacamento h:

0.1

Digite o número de subintervalos m:

10

Os valores de x e y são:

0.000000, 2.000000;

0.100000, 2.005000;

0.200000, 2.019025;

0.300000, 2.041218;

0.400000, 2.070802;

0.500000, 2.107076;

0.600000, 2.149403;

0.700000, 2.197210;

0.800000, 2.249975;

0.900000, 2.307227;

1.000000, 2.368541;

cujo gráfico, comparado com o resultado analítico, é apresentado na figura 3. Percebe-se que, para esta metodologia, o gráfico do método numérico sobrepõe-se ao gráfico do método analítico.

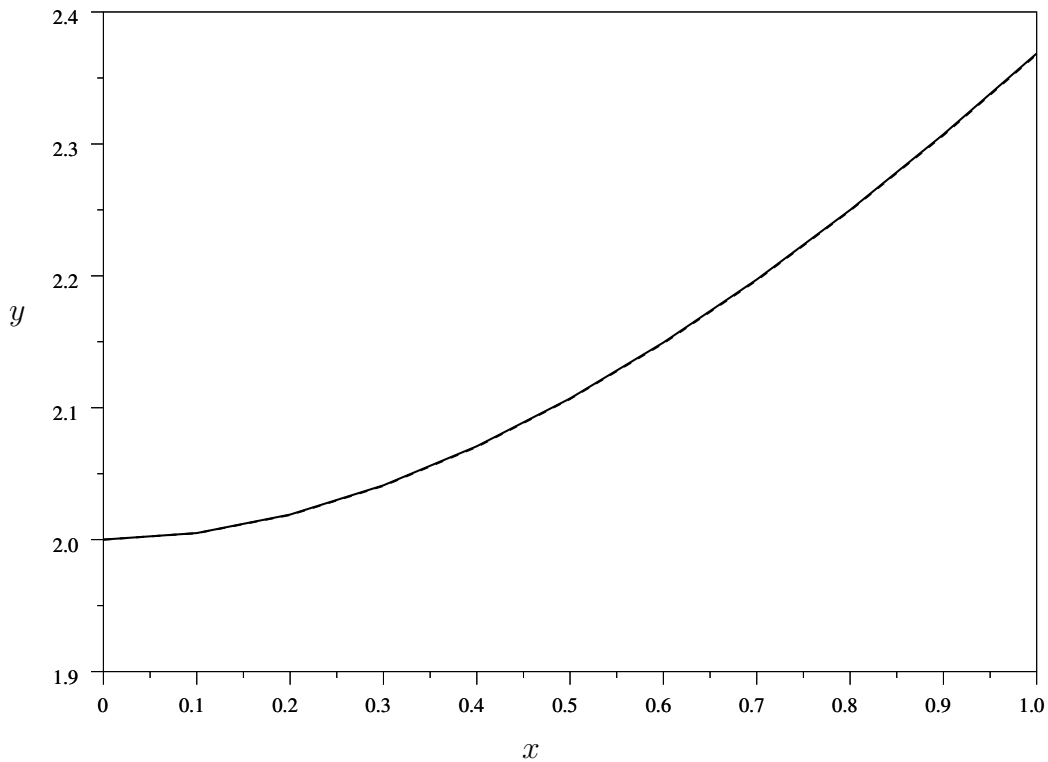


Figura 3: Comparação entre o resultado analítico (linha tracejada) e o Método de Euler definido pela Série de Taylor com três termos (linha sólida).

A grande desvantagem é que em aplicações práticas não há o conhecimento da forma analítica de $f(x, y)$. Assim, não é possível utilizar derivadas de ordem mais elevada nem estimar o erro local de truncamento para tais casos. A opção então é utilizar métodos que tenham uma boa precisão mas que não necessite das derivadas de ordem superior. A próxima seção apresentará estes métodos.

3 Métodos de Runge-Kutta

Mesmo utilizando recursos para diminuir o erro, o Método de Euler não é utilizado na prática. Em seu lugar aplica-se recursos que possuem uma precisão satisfatória e dispensem as derivadas de ordem superior de $f(x, y)$. Estes recursos são chamados de Métodos de Runge-Kutta.

Há vários métodos que são diferenciados por sua ordem. O mais simples é o Runge-Kutta de primeira ordem que concorda com o método de Euler, ou seja, é o próprio método de Euler.

Há também os Métodos de Runge-Kutta de segunda ordem. Eles podem assumir diferentes formatos. Os dois principais são:

$$\begin{cases} y_{j+1} = y_j + \frac{h}{2} \cdot (K_1 + K_2) \\ K_1 = f(x_j, y_j) \\ K_2 = f(x_j + h, y_j + h \cdot K_1) \end{cases}$$

com $j = 0, 1, 2, \dots, m - 1$. Este método é chamado de *Euler Melhorado*. O segundo

método é:

$$\begin{cases} y_{j+1} &= y_j + h \cdot K_2 \\ K_1 &= f(x_j, y_j) \\ K_2 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2} \cdot K_1) \end{cases}$$

também com $j = 0, 1, 2, \dots, m - 1$, mas é chamado de *Euler Modificado*. O modo como estas fórmulas foram obtidas encontra-se no livro texto [1].

Os métodos de Runge-Kutta de segunda ordem possuem erro local de truncamento igual a:

$$e_j = \frac{h^3}{3!} \cdot y'''(\xi)$$

sendo $x_{j-1} < \xi < x_j$. Nota-se que este erro é menor que o apresentado pelo Método de Euler mais simples e igual ao erro que o método de Euler possui se for utilizado a fórmula obtida pela Série de Taylor até o terceiro termo.

Dentre os métodos de Runge-Kutta, o mais popular é o de 4ª ordem, que é bastante difundido nas rotinas de cálculo computacional [1]. A sua formulação geral é:

$$\begin{cases} y_{j+1} &= y_j + \frac{h}{6} \cdot (K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 &= f(x_j, y_j) \\ K_2 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2}K_1) \\ K_3 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2}K_2) \\ K_4 &= f(x_j + h, y_j + h \cdot K_3) \end{cases}$$

com erro local de truncamento igual a:

$$e_j = \frac{h^5}{5!} \cdot y^{(v)}(\xi)$$

e $x_{j-1} < \xi < x_j$.

Exemplo 2 Resolver o exemplo anterior utilizando o Método de Runge-Kutta de quarta ordem.

O exemplo anterior diz que a equação diferencial ordinária com problema de valor inicial é dada por:

$$\begin{cases} y' = x - y + 2 \\ y(0) = 2 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.1$.

Destas informações, é possível identificar que $x_0 = 0$, $y_0 = 2$ e que a quantidade de subintervalos é igual a 10. Assim, o índice j varia de 0 até 9. Para $j = 0$:

$$\begin{aligned} K_1 &= f(x_0, y_0) \\ &= x_0 - y_0 + 2 \\ &= 0 - 2 + 2 \\ &= 0 \end{aligned}$$

$$\begin{aligned}
K_2 &= f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}K_1\right) \\
&= f\left(0 + \frac{0.1}{2}, 2 + \frac{0.1}{2} \cdot 0\right) \\
&= f(0.05, 2) \\
&= 0.05 - 2 + 2 \\
&= 0.05
\end{aligned}$$

$$\begin{aligned}
K_3 &= f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}K_2\right) \\
&= f\left(0 + \frac{0.1}{2}, 2 + \frac{0.1}{2} \cdot 0.05\right) \\
&= f(0.05, 2.0025) \\
&= 0.05 - 2.0025 + 2 \\
&= 0.0475
\end{aligned}$$

$$\begin{aligned}
K_4 &= f(x_0 + h, y_0 + h \cdot K_3) \\
&= f(0 + 0.1, 2 + 0.1 \cdot 0.0475) \\
&= f(0.1, 2.00475) \\
&= 0.1 - 2.00475 + 2 \\
&= 0.09525
\end{aligned}$$

O valor da aproximação é:

$$\begin{aligned}
y_1 &= y_0 + \frac{h}{6} \cdot (K_1 + 2K_2 + 2K_3 + K_4) \\
&= 2 + \frac{0.1}{6} \cdot (0 + 2 \cdot 0.05 + 2 \cdot 0.0475 + 0.09525) \\
&= 2.0048375
\end{aligned}$$

Este procedimento deve ser repetido até que $j = 9$. Como é um procedimento cansativo para se fazer manualmente, então será utilizado o programa escrito em C, que foi aplicado para este mesmo exemplo porém com o Método de Euler, com algumas modificações. Estas mudanças podem ser vistas no apêndice A.2. Os resultados apresentados foram:

Os valores de x e y são:

0.000000, 2.000000;
0.100000, 2.004838;
0.200000, 2.018731;
0.300000, 2.040818;
0.400000, 2.070320;
0.500000, 2.106531;
0.600000, 2.148812;
0.700000, 2.196585;
0.800000, 2.249329;
0.900000, 2.306570;
1.000000, 2.367880;

Se comparada com o resultado analítico vê-se que o resultado é bastante satisfatório. Na figura 4 as curvas das soluções exata e numérica se sobrepõem.

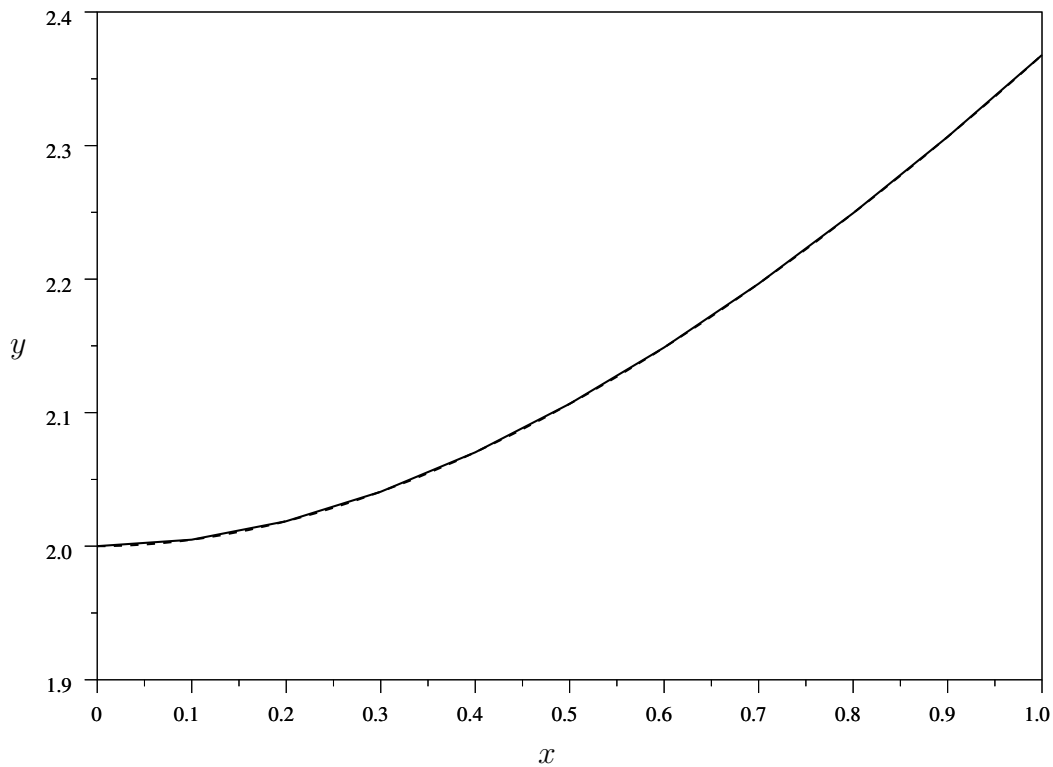


Figura 4: Comparação dos resultados obtidos analiticamente (linha tracejada) e da solução numérica por Runge-Kutta de 4ª ordem (linha sólida).

4 Métodos de Adams-Bashforth

Os métodos de Adams-Bashforth são baseados em integração numérica. Eles fazem parte dos métodos de passo múltiplo, ou seja, são métodos que precisam de mais de um valor calculado anteriormente para determinar a aproximação y_{j+1} (diferentemente dos Métodos de Euler e Runge-Kutta, que precisam somente de um valor calculado anteriormente). Um método é dito ser de passo k se ele precisar de k resultados anteriores.

Por definição, uma solução exata de uma E.D.O. de primeira ordem com problema de valor inicial é:

$$y(x+q) = y(x) + \int_x^{x+q} f(\beta, y(\beta)) \cdot d\beta$$

para quaisquer pontos x e $x+q$ no intervalo $[a, b]$. O termo β é uma variável de integração. Os métodos de Adams-Bashforth baseiam-se na idéia de substituir a função $f(x, y(x))$, que é desconhecida, por um polinômio interpolador que assuma valores $f_j = f(x_j, y_j)$ num conjunto de pontos x_j , onde y_j já foi obtido ou está sendo calculado. Assim, calcula-se a sua integral e aceita-se o seu valor como incremento para calcular y_{j+1} , através de:

$$y_{j+1} = y_j + \int_{x_j}^{x_{j+1}} f(x, y) \cdot dx \quad (3)$$

4.1 Adams-Bashforth de Passo Dois

Por ser de passo dois, este método necessita de dois valores calculados anteriormente. Sendo uma Equação Diferencial Ordinária (E.D.O.) com Problema de Valor Inicial (P.V.I.), tem-se o valor do primeiro ponto (x_0, y_0) . Através de algum método de passo simples, calcula-se a aproximação y_1 , obtendo o segundo ponto (x_1, y_1) .

Tendo agora estes dois valores, é possível utilizar o Método de passo dois para encontrar y_2 . Assim, utilizando a equação 3:

$$y_2 = y_1 + \int_{x_1}^{x_2} f(x, y) \cdot dx \quad (4)$$

Como os valores $f(x_0, y_0) = f_0$ e $f(x_1, y_1) = f_1$ são conhecidos então pode-se aproximar a função $f(x, y)$ pelo polinômio interpolador de primeiro grau $P_1(x)$. Desse modo:

$$P_1(x) = b_0 p_0(x) + b_1 p_1(x)$$

onde:

$$\begin{aligned} p_0(x) &= x - x_1 \\ p_1(x) &= x - x_0 \\ b_0 &= \frac{f_0}{x_0 - x_1} \\ b_1 &= \frac{f_1}{x_1 - x_0} \end{aligned}$$

Substituindo este polinômio na equação 4 e lembrando que $x_1 - x_0 = h$, então:

$$y_2 = y_1 - \frac{f_0}{h} \int_{x_1}^{x_2} p_0(x) \cdot dx + \frac{f_1}{h} \int_{x_1}^{x_2} p_1(x) \cdot dx$$

Integrando esta equação e rearranjando os termos, o resultado fica:

$$y_2 = y_1 + \frac{h}{2} \cdot (3f_1 - f_0)$$

Para calcular y_3 adota-se o mesmo procedimento, porém lembrando que somente devem ser utilizados os pontos (x_1, y_1) e (x_2, y_2) :

$$y_3 = y_2 + \frac{h}{2} \cdot (3f_2 - f_1)$$

Como este procedimento também é utilizado para $j = 3, 4, \dots, m - 1$, então pode-se utilizar a seguinte regra:

$$y_{j+1} = y_j + \frac{h}{2} \cdot (3f_j - f_{j-1})$$

com $j = 1, 2, 3, \dots, m - 1$. Salienta-se ainda que y_1 deve ser calculado por Euler ou Runge-Kutta.

4.2 Adams-Bashforth de Passo Quatro

Tem os mesmos princípios do método de passo dois, porém utiliza quatro pontos previamente calculados: (x_0, y_0) , (x_1, y_1) , (x_2, y_2) e (x_3, y_3) . Por utilizar estes quatro pontos, o polinômio interpolador deve ser de terceiro grau. Assim:

$$y_{j+1} = y_j + \int_{x_j}^{x_{j+1}} P_3(x) \cdot dx$$

com:

$$P_3(x) = b_0p_0(x) + b_1p_1(x) + b_2p_2(x) + b_3p_3(x)$$

que integrando resulta em:

$$y_{j+1} = y_j + \frac{h}{24} \cdot (55f_j - 59f_{j-1} + 37f_{j-2} - 9f_{j-3})$$

sendo $j = 3, 4, 5, \dots, m - 1$. Os termos y_1 , y_2 e y_3 devem ser calculados por métodos de passo simples ou por Adams-Bashforth de passo dois.

4.3 Método de Adams-Bashforth-Moulton de Passo Quatro

Este método também é chamado de *Preditor-Corretor*. Neste caso tem-se cinco pontos conhecidos mas apenas quatro são utilizados: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) e (x_4, y_4) . O que se deseja então é melhorar a precisão da aproximação y_4 . Assim, utilizando-se um polinômio interpolador de terceira ordem para aproximar a função $f(x, y)$, tem-se:

$$y_4 = y_3 + \int_{x_3}^{x_4} P_3(x) \cdot dx$$

cujo resultado é:

$$y_4 = y_3 + \frac{h}{24} \cdot (9f_4 + 19f_3 - 5f_2 + f_1) \quad (5)$$

É possível perceber uma incoerência nesta equação pois para calcular a aproximação y_4 necessita-se de $f_4 = f(x_4, y_4)$, que também necessita de y_4 . Este problema pode ser contornado se a equação 5 for usada apenas para correção, enquanto o valor de y_4 foi obtido através de uma predição. Portanto, é possível adotar a seguinte notação:

$$y_4^C = y_3 + \frac{h}{24} \cdot (9f_4^P + 19f_3 - 5f_2 + f_1)$$

sendo $f_4^P = f(x_4, y_4^P)$. Este valor predito pode ser obtido por Adams-Bashforth de passo quatro:

$$y_4^P = y_3 + \frac{h}{24} \cdot (55f_3 - 59f_2 + 37f_1 - 9f_0)$$

Escrevendo de uma forma generalista:

$$\begin{aligned} y_{j+1}^P &= y_j + \frac{h}{24} \cdot (55f_j - 59f_{j-1} + 37f_{j-2} - 9f_{j-3}) \\ f_{j+1}^P &= f(x_{j+1}, y_{j+1}^P) \\ y_{j+1}^C &= y_j + \frac{h}{24} \cdot (9f_{j+1}^P + 19f_j - 5f_{j-1} + f_{j-2}) \end{aligned}$$

com $j = 3, 4, \dots, m - 1$.

Para resolver um problema de valor inicial com o par Preditor-Corretor tem-se que seguir três fases:

1. Calculam-se y_1 , y_2 e y_3 por um método de passo simples de quarta ordem;
2. Calcula-se o valor predito y_4^P ;
3. Corrige-se este valor predito:
 - (a) Calcula-se f_4^P ;
 - (b) Calcula-se y_4^C ;
 - (c) Atualiza-se y_4^P , isto é, $y_4^P \leftarrow y_4^C$;
 - (d) Calcula-se novamente f_4^P com o valor atualizado;
 - (e) Calcula-se novamente y_4^C com o valor atualizado.

A aproximação y_4 será este último valor calculado. Para as aproximações seguintes (y_5, y_6, \dots, y_m) deve-se utilizar os mesmos procedimentos das fases 2 e 3.

Exemplo 3 Obter as aproximações para a equação diferencial ordinária com problema de valor inicial:

$$\begin{cases} y'(x) = \frac{x - 2xy - 1}{x^2} \\ y(1) = 0 \end{cases}$$

na malha de $[1, 2]$ com espaçamento igual a 0.01.

O número de subintervalos a ser calculado é:

$$m = \frac{2 - 1}{0.01} = 100$$

Dessa forma, utilizando o programa listado no apêndice A.3, tem-se as seguintes aproximações:

```

1.000000, 0.000000;
1.010000, 0.000049;
1.020000, 0.000192;
1.030000, 0.000424;
1.040000, 0.000740;
1.050000, 0.001134;
1.060000, 0.001602;
1.070000, 0.002140;
1.080000, 0.002743;
1.090000, 0.003409;
1.100000, 0.004132;
1.110000, 0.004910;
1.120000, 0.005740;
1.130000, 0.006618;
1.140000, 0.007541;
1.150000, 0.008507;
1.160000, 0.009512;
1.170000, 0.010556;
1.180000, 0.011635;
1.190000, 0.012746;

```

1.200000, 0.013889;
1.210000, 0.015060;
1.220000, 0.016259;
1.230000, 0.017483;
1.240000, 0.018730;
1.250000, 0.020000;
1.260000, 0.021290;
1.270000, 0.022599;
1.280000, 0.023926;
1.290000, 0.025269;
1.300000, 0.026627;
1.310000, 0.028000;
1.320000, 0.029385;
1.330000, 0.030782;
1.340000, 0.032190;
1.350000, 0.033608;
1.360000, 0.035035;
1.370000, 0.036470;
1.380000, 0.037912;
1.390000, 0.039361;
1.400000, 0.040816;
1.410000, 0.042277;
1.420000, 0.043741;
1.430000, 0.045210;
1.440000, 0.046682;
1.450000, 0.048157;
1.460000, 0.049634;
1.470000, 0.051113;
1.480000, 0.052593;
1.490000, 0.054074;
1.500000, 0.055556;
1.510000, 0.057037;
1.520000, 0.058518;
1.529999, 0.059998;
1.539999, 0.061477;
1.549999, 0.062955;
1.559999, 0.064431;
1.569999, 0.065905;
1.579999, 0.067377;
1.589999, 0.068846;
1.599999, 0.070312;
1.609999, 0.071776;
1.619999, 0.073236;
1.629999, 0.074692;
1.639999, 0.076145;
1.649999, 0.077594;
1.659999, 0.079039;
1.669999, 0.080480;

```

1.679999, 0.081916;
1.689999, 0.083348;
1.699999, 0.084775;
1.709999, 0.086197;
1.719999, 0.087615;
1.729999, 0.089027;
1.739999, 0.090435;
1.749999, 0.091837;
1.759999, 0.093233;
1.769999, 0.094625;
1.779999, 0.096011;
1.789999, 0.097391;
1.799999, 0.098765;
1.809999, 0.100134;
1.819999, 0.101497;
1.829999, 0.102855;
1.839999, 0.104206;
1.849999, 0.105551;
1.859999, 0.106891;
1.869999, 0.108224;
1.879999, 0.109552;
1.889999, 0.110873;
1.899999, 0.112188;
1.909999, 0.113497;
1.919999, 0.114800;
1.929999, 0.116097;
1.939999, 0.117388;
1.949999, 0.118672;
1.959999, 0.119950;
1.969999, 0.121222;
1.979999, 0.122487;
1.989999, 0.123747;
1.999999, 0.125000;

```

Este resultado pode ser plotado no Scilab (figura 5).

Observação 1 Intervalo de variação do espaçamento h para garantir a estabilidade do método.

Método	Intervalo
Euler	(0, 2)
RK2	(0, 2)
RK4	(0, 2.78)
ABM4	(0, 0.9)

Observação 2 Uma equação diferencial ordinária com problema de valor inicial pode ser solucionada numericamente através de uma função própria do Scilab chamada `ode`. Para saber mais sobre ela, deve-se digitar na linha de comando do Scilab:

```
-->help ode
```

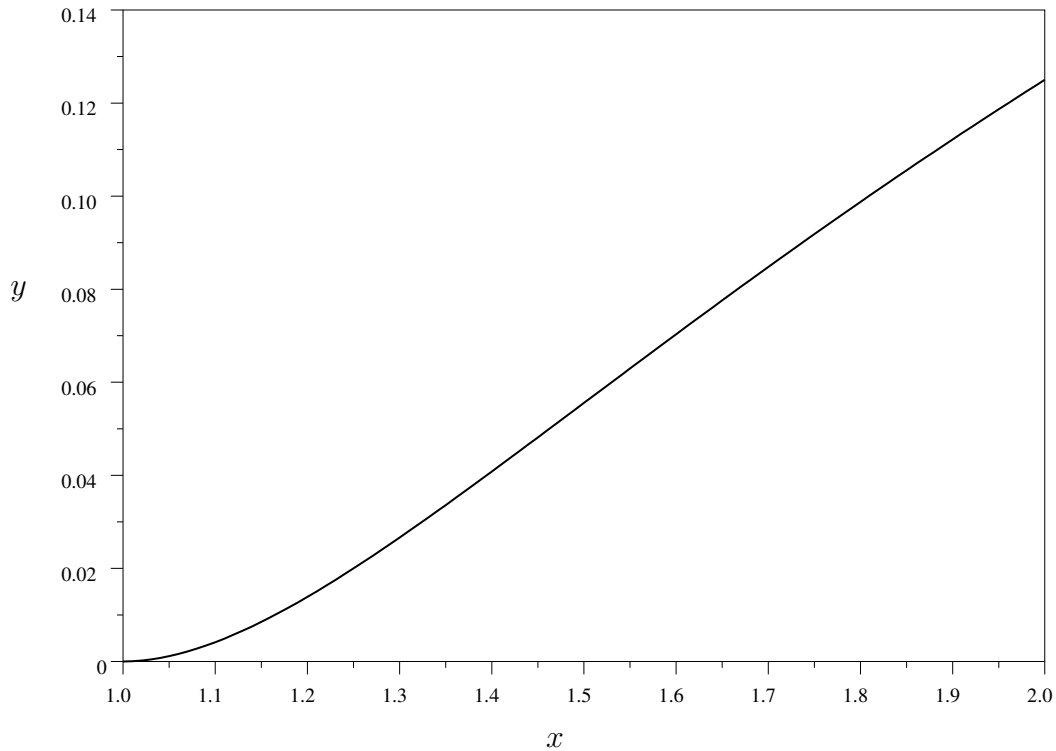



Figura 5: Resultado apresentado pelo método Preditor-Corretor.

5 Exercícios

1. Achar as aproximações, por Método de Euler, para os seguintes problemas de valor inicial:

$$(a) \begin{cases} y'(x) = x - y + 2 \\ y(0) = 2 \end{cases}$$

na malha $[0, 1]$ com $h = 0.05$ e $h = 0.01$ (sugestão: criar um programa em alguma linguagem de preferência e resolver este problema).

$$(b) \begin{cases} y'(x) = y - \frac{2x}{y} \\ y(0) = 1 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.2$.

$$(c) \begin{cases} y'(x) = \frac{1}{x} \\ y(1) = 0 \end{cases}$$

na malha de $[1, 2]$ com $h = 0.1$.

2. Achar as aproximações para os problemas a seguir utilizando os Métodos de Runge-Kutta.

$$(a) \begin{cases} y'(x) = x - y + 2 \\ y(0) = 2 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.1$, utilizando o Euler Melhorado.

$$(b) \quad \begin{cases} y'(x) = \frac{1}{x} \\ y(1) = 0 \end{cases}$$

na malha de $[1, 2]$ com $h = 0.1$, usando o Euler Modificado.

$$(c) \quad \begin{cases} y'(x) = y - \frac{2x}{y} \\ y(0) = 1 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.1$, usando o Runge-Kutta de quarta ordem.

$$(d) \quad \begin{cases} y'(x) = \frac{x - 2xy - 1}{x^2} \\ y(1) = 0 \end{cases}$$

na malha de $[1, 2]$ com $h = 0.1$, usando o Runge-Kutta de quarta ordem.

$$(e) \quad \begin{cases} y'(x) = \frac{y^2 - 1}{x^2 + 1} \\ y(0) = 1 \end{cases}$$

na malha de $[0, 1]$, usando o Euler Melhorado, com $h = 0.1$ e $h = 0.05$. Comparar os resultados com a solução exata:

$$y(x) = \frac{1 - x}{1 + x}$$

3. Obter as aproximações para os problemas a seguir utilizando os Métodos de Adams-Bashforth-Moulton:

$$(a) \quad \begin{cases} y'(x) = \frac{y^2 - 1}{x^2 + 1} \\ y(0) = 1 \end{cases}$$

na malha de $[0, 1]$ com $h = 0.1$, usando o Preditor-Corretor.

$$(b) \quad \begin{cases} y'(x) = -xy^2 \\ y(1) = 2 \end{cases}$$

na malha de $[1, 2]$ com $h = 0.1$, usando o Runge-Kutta de quarta ordem e o Preditor-Corretor. Compare os resultados plotando os gráficos.

4. Conhecidos os pares (x_0, y_0) , (x_1, y_1) e (x_2, y_2) , obter o Método Preditor-Corretor de passo dois.

Referências

- [1] *Cálculo Numérico (com aplicações)*; Leônidas C. Barroso, Magali M. A. Barroso, Frederico F. Campos, Márcio L. B. Carvalho, Miriam L. Maia; Editora Harbra; Segunda edição; 1987.
- [2] *Matemática para Economia e Administração*; Jean E. Weber; Editora Harbra; Segunda edição; 2001.
- [3] *Cálculo Numérico - Características Matemáticas e Computacionais dos Métodos Numéricos*; Décio Sperandio, João T. Mendes, Luiz H. Monken e Silva; Prentice-Hall; 2003.

A Listagem de Programas

A.1 Método de Euler

```
// Inclusão das bibliotecas do C (possuem funções necessárias ao
// programa)
#include <math.h>
#include <stdio.h>

// Declaração da função (derivada)
float dy(float, float);

int main()
{
    // Declaração das variáveis
    float x0, y0, h;
    int m, j;
    // Entrada de dados
    printf("Digite o valor do x inicial: \n");
    scanf("%f",&x0);
    printf("Digite o valor do y inicial: \n");
    scanf("%f",&y0);
    printf("Digite o valor do espacamento h: \n");
    scanf("%f",&h);
    printf("Digite o número de subintervalos m: \n");
    scanf("%d",&m);

    // Cálculo de x e y pelo Método de Euler
    float x[m+1], y[m+1];
    x[0] = x0;
    y[0] = y0;
    for(j=0;j<m;j++)
    {
        x[j+1] = x[j] + h;
        y[j+1] = y[j] + h*dy(x[j],y[j]);
    }
    // Exibição dos valores de x e y
    printf("Os valores de x e y sao:\n");
    for(j=0;j<=m;j++)
        printf("%f, %f; \n",x[j], y[j]);

    return(0);
}

// Escopo da função (derivada)
float dy(float x, float y)
{
    return(x-y+2);
}
```

A.2 Runge-Kutta de Quarta Ordem

```
#include <math.h>
#include <stdio.h>

float dy(float, float);

int main()
{
    float x0, y0, h;
    float K1, K2, K3, K4;
    int m, j;

    printf("Digite o valor do x inicial: \n");
    scanf("%f",&x0);
    printf("Digite o valor do y inicial: \n");
    scanf("%f",&y0);
    printf("Digite o valor do espacamento h: \n");
    scanf("%f",&h);
    printf("Digite o número de subintervalos m: \n");
    scanf("%d",&m);

    float x[m+1], y[m+1];
    x[0] = x0;
    y[0] = y0;
    for(j=0;j<m;j++)
    {
        K1 = dy(x[j], y[j]);
        K2 = dy(x[j]+h/2, y[j]+(h/2)*K1);
        K3 = dy(x[j]+h/2, y[j]+(h/2)*K2);
        K4 = dy(x[j]+h, y[j]+h*K3);
        x[j+1] = x[j] + h;
        y[j+1] = y[j] + (h/6)*(K1 + 2*K2 + 2*K3 + K4);
    }

    printf("Os valores de x e y sao:\n");
    for(j=0;j<=m;j++)
        printf("%f, %f; \n",x[j], y[j]);

    return(0);
}

float dy(float x, float y)
{
    return(x-y+2);
}
```

A.3 Preditor-Corretor de Passo Quatro

```

#include <math.h>
#include <stdio.h>

float dy(float, float);

int main()
{
    float x0, y0, h;
    float K1, K2, K3, K4;
    int m, j;
    FILE *arqv; // Definição de um ponteiro para um arquivo (para gravar)
                // os dados

    printf("Digite o valor do x inicial: \n");
    scanf("%f",&x0);
    printf("Digite o valor do y inicial: \n");
    scanf("%f",&y0);
    printf("Digite o valor do espacamento h: \n");
    scanf("%f",&h);
    printf("Digite o número de subintervalos m: \n");
    scanf("%d",&m);

    // Cálculo de x[0]->x[3] e y[0]->y[3] pelo método RK4
    float x[m+1], y[m+1];
    x[0] = x0;
    y[0] = y0;
    for(j=0;j<3;j++)
    {
        K1 = dy(x[j], y[j]);
        K2 = dy(x[j]+h/2, y[j]+(h/2)*K1);
        K3 = dy(x[j]+h/2, y[j]+(h/2)*K2);
        K4 = dy(x[j]+h, y[j]+h*K3);
        x[j+1] = x[j] + h;
        y[j+1] = y[j] + (h/6)*(K1 + 2*K2 + 2*K3 + K4);
    }

    // Cálculo dos outros elementos pelo Preditor-Corretor
    float yp, yc;
    for(j=3;j<m;j++)
    {
        x[j+1] = x[j] + h;
        yp = y[j] + (h/24)*(55*dy(x[j],y[j]) - 59*dy(x[j-1],y[j-1]) +
        37*dy(x[j-2],y[j-2]) - 9*dy(x[j-3],y[j-3]));
        yc = y[j] + (h/24)*(9*dy(x[j+1],yp) + 19*dy(x[j],y[j]) -
        5*dy(x[j-1],y[j-1]) + dy(x[j-2],y[j-2]));
        y[j+1] = y[j] + (h/24)*(9*dy(x[j+1],yc) + 19*dy(x[j],y[j]) -
        5*dy(x[j-1],y[j-1]) + dy(x[j-2],y[j-2]));
    }
}

```

```
    }

    // Gravação dos resultados em um arquivo do Scilab
    arqv = fopen("resultado.sci","w");
    printf("Os valores de x e y estão em 'resultado.sci'!\n");
    for(j=0;j<=m;j++)
    {
        fprintf(arqv,"%f, %f; \n",x[j],y[j]);
    }
    fclose(arqv);
    printf("Numero de elementos: %d\n",j);
    return(0);
}

float dy(float x, float y)
{
    return((x - 2*x*y -1)/powf(x,2));
}
```