Universidade Federal da Paraíba
Discente: Jefferson Bezerra dos Santos
Docente: José Miguel Aroztegui Massera

Segunda lista de EDO

Observação:
Em algumas partes do código foram inseridas quebras de linha para que o código fosse inserido na folha A4. Caso seja necessário verificar o funcionamento dos códigos esses podem ser encontrados na pasta desse arquivo com a extensão *m.

1. Questão:
O código a seguir se refere a implementação do método de Taylor para q = 3.

```
###==================== PROGRAM ============================
# Program:taylor3.m
# Description: Solve ODE numeric by the Taylor method
# Date of Create: dom 20 out 2019 12:52:05
# Update in: ter 22 out 2019 15:08:15
# Author: Jefferson Bezerra dos Santos
#----------------------------------------------------------

### ============= Taylor method ============================
function [x,y] = taylor3(f,f1,f2,xo,yo,h,N)
x = xo:h:(xo + h*N);
y(1) = yo;
for i = 1:(length(x)-1)
     y(i+1) = y(i) + h*f(x(i),y(i)) + ((h^2)/2)*f1(x(i),y(i))
 + ((h^3)/6)*f2(x(i),y(i));
endfor
endfunction #------ End of function ----------------------
```

Universidade Federal da Paraíba
Discente: Jefferson Bezerra dos Santos
Docente: José Miguel Aroztegui Massera

2. Questão:

O código a seguir se refere a resolução do problema de valor inicial pelo método de Taylor para q = 3.

```
###==================== PROGRAM =============================
# Program: teste.m
# Description: Plot of solutions analytic and numeric
# Date of Create: dom 20 out 2019 13:41:49
# Update in: dom 20 out 2019 13:43:51
# Author:Jefferson Bezerra dos Santos
#------------------------------------------------------------

###============= Clear enviromment ===========================
clc
clear
#------------------------------------------------------------

###================ Defining auxiliary functions =============
f  = @(x,y) y - x + 2;#Order derivative 0
f1 = @(x,y) y - x + 1;#Order derivative 1
f2 = @(x,y) y - x + 1;#Order derivative 2
#------------------------------------------------------------

### ===== Taylor method, paramenters(f,f1,f2,x0,y0,h,N) ======
[x,y] = taylor3(f,f1,f2,0, 2,0.1,50);
#------------------------------------------------------------

###===== Solution analytic, paramenters(x0,y0,h,N) ===========
[x1,y1] = analitica(0,2,0.1,50);

### =========== Plotting ====================================
#color b for blue, '.' for point, + is type of symbol
plot(x,y,'b.+',x1,y1,'r.x');
title('Solution of ODE');
xlabel('Axis (x)'); ylabel('Axis (y)');
h = legend("Taylor method","Analytic solution","location","northwest");
set(h,"fontsize",11);
legend("boxoff");
grid on
print -depsc plot2.eps #Figure in color eps format
#------------------------------------------------------------
```

Universidade Federal da Paraíba
Discente: Jefferson Bezerra dos Santos
Docente: José Miguel Aroztegui Massera

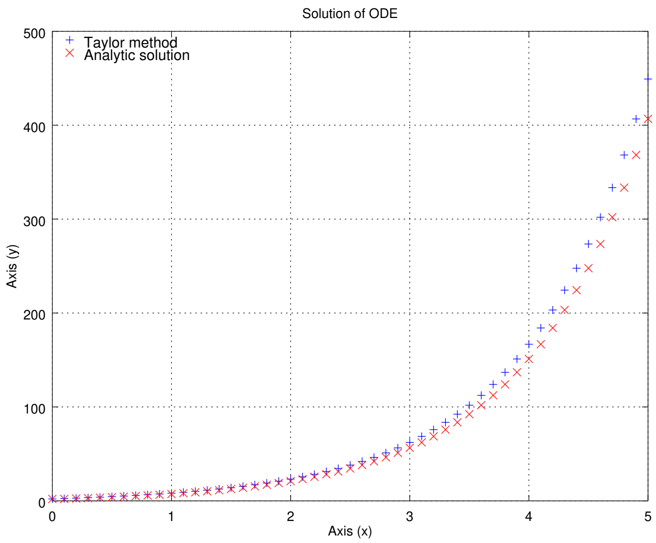


Figura 1: Comparação solução analitica e o método de Taylor q = 3.

3. Questão:

O código a seguir se refere a implementação do método de Euler.

```
###==================== PROGRAM ============================
# Program: euler.m
# Date of Create:Solve ODE the Euler method
# Update in: ter 22 out 2019 15:05:48
# Author:Jefferson Bezerra dos Santos
#----------------------------------------------------------

### =============== Euler method ===========================
function [x,y] = euler(f,xo,yo,h,N)
x = xo:h:(xo + h*N);
y(1) = yo;
for i = 1:(length(x)-1)
     y(i+1) = y(i) + h*f(x(i),y(i));
endfor
endfunction #--- End of function --------------------------
```

Universidade Federal da Paraíba
Discente: Jefferson Bezerra dos Santos
Docente: José Miguel Aroztegui Massera

O código a seguir se refere a resolução do problema de valor inicial pelo método de Euler.

```
###==================== PROGRAM =============================
# Program: testeEuler.m
# Description: Plot of solutions analytic and numeric
# Date of Create: dom 20 out 2019 13:41:49
# Update in: dom 20 out 2019 13:43:51
# Author:Jefferson Bezerra dos Santos
#-------------------------------------------------------------

###============= Clear envirомment ===========================
clc
clear
#-------------------------------------------------------------

###================ Defining auxiliary functions =============
f  = @(x,y) y - x + 2;#Order derivative 0
#-------------------------------------------------------------

### ===== Euler method, paramenters(f,x0,y0,h,N) =============
[x,y] = euler(f,0, 2,0.1,50);
#-------------------------------------------------------------

###===== Solution analytic, paramenters(x0,y0,h,N) ===========
[x1,y1] = analitica(0,2,0.1,50);

### =========== Plotting =====================================
#color b for blue, '.' for point, + is type of symbol
plot(x,y,'b.+',x1,y1,'r.x');
title('Solution of ODE');
xlabel('Axis (x)');
ylabel('Axis (y)');
h = legend("Euler method","Analytic solution","location","northwest");
set(h,"fontsize",11);
legend("boxoff");
grid on
print -depsc plot2.eps #Figure in color eps format
#-------------------------------------------------------------
```

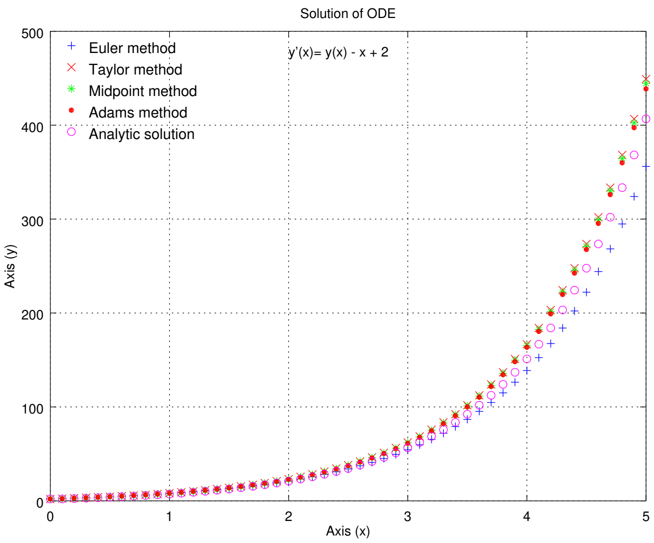


Figura 2: Solução do problema de valor inicial pelo método de Euler.

Segue o código modificado de teste.m com os métodos de Euler e Taylor.

```
###===================== PROGRAM =============================
# Program: testeEulerTaylor.m
# Description: Plot of solutions analytic and numeric
# Date of Create: dom 20 out 2019 13:41:49
# Update in: dom 20 out 2019 13:43:51
# Author:Jefferson Bezerra dos Santos
#-------------------------------------------------------------

###============= Clear enviromment ===========================
clc
clear
#-------------------------------------------------------------

###================ Defining auxiliary functions =============
f  = @(x,y) y - x + 2;#Order derivative 0
f1 = @(x,y) y - x + 1;#Order derivative 1
f2 = @(x,y) y - x + 1;#Order derivative 2
#-------------------------------------------------------------

### ===== Euler method, paramenters(f,x0,y0,h,N) =============
[x,y] = euler(f,0, 2,0.1,50);
```

```
#--------------------------------------------------------------

###===== Solution analytic, paramenters(x0,y0,h,N) ===========
[x1,y1] = analitica(0,2,0.1,50);

###===== Solution analytic, paramenters(x0,y0,h,N) ===========
[x2,y2] = taylor3(f,f1,f2,0,2,0.1,50);


### =========== Plotting ======================================
#color b for blue, '.' for point, + is type of symbol
plot(x,y,'b.+',x1,y1,'m.o',x2,y2,'r.x');
title('Solution of ODE');
xlabel('Axis (x)');
ylabel('Axis (y)');
h = legend("Euler method","Analytic solution","Taylor method",
"location","northwest");
set(h,"fontsize",11);
legend("boxoff");
grid on
print -depsc plot3.eps #Figure in color eps format
#--------------------------------------------------------------
```

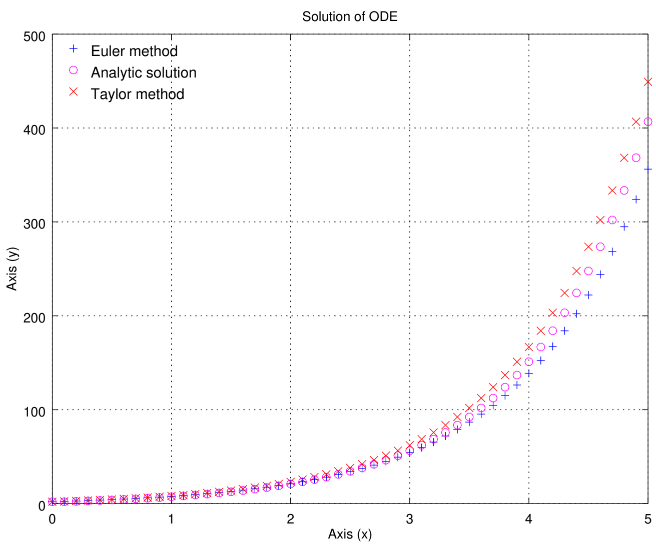


Figura 3: Solução do problema de valor inicial por Euler e Taylor.

4. Questão:
A implementação do método de Adams Bashforth.

```
###==================== PROGRAM ============================
# Program: adams_bashforth.m
# Date of Create:Solve ODE the Euler method
# Update in: dom 20 out 2019 12:56:45 -03
# Author:Jefferson Bezerra dos Santos
#------------------------------------------------------------

### =============== adams bashfort =========================
function [x,y] = adams_bashforth(f,xo,yo,h,N)
x = xo:h:(xo + h*N);
y(1) = yo;
    y(2) = yo + h*f(x(1),y(1));
for i = 1:(length(x)-2)
     y(i+2) = y(i+1) + (h/2)*((-f(x(i),y(i))) + 3*(f(x(i+1),y(i+1))));
endfor
endfunction #--- End of function ---------------------------
```

A implementação do método do ponto médio.

```
###==================== PROGRAM ============================
# Program: ponto_medio.m
# Date of Create:Solve ODE the Euler method
# Update in: ter 22 out 2019 15:05:04
# Author:Jefferson Bezerra dos Santos
#------------------------------------------------------------

### =============== Mitpoin method==========================
function [x,y] = ponto_medio(f,xo,yo,h,N)
x = xo:h:(xo + h*N);
y(1) = yo;
    y(2) = yo + h*f(x(1),y(1));
for i = 1:(length(x)-2)
     y(i+2) = y(i) + 2*h*f(x(i+1),y(i+1));
endfor
endfunction #--- End of function ---------------------------
```

5. Questão:
Segue o código para a questão 5.

```
###===================== PROGRAM =============================
# Program: teste3.m
# Description: Plot of solutions analytic and numeric
# Date of Create: seg 21 out 2019 12:29:08
# Update in: ter 22 out 2019 15:06:37
# Author:Jefferson Bezerra dos Santos
#-------------------------------------------------------------

###============= Clear enviromment ===========================
clc
clear
#-------------------------------------------------------------

###================ Defining auxiliary functions =============
f = @(x,y) y/(1 + 3*x);#Order derivative 0
f1 = @(x,y) -2*y/((1 + 3*x)^2);#Order derivative 1
f2 = @(x,y) 10*y/((1 + 3*x)^3);;#Order derivative 2
#-------------------------------------------------------------

###=== Euler method, paramenters(f,x0,y0,h,N) ===============
[x,y] = euler(f,0, 2,1.0,10);
#-------------------------------------------------------------

### ===== Taylor method, paramenters(f,f1,f2,x0,y0,h,N) ======
[x1,y1] = taylor3(f,f1,f2,0,2,1.0,10);
#-------------------------------------------------------------

###=== Midpoint method, paramenters(f,x0,y0,h,N) =============
[x2,y2] = ponto_medio(f,0, 2,1.0,10);
#-------------------------------------------------------------

###=== Adams method, paramenters(f,x0,y0,h,N) ================
[x3,y3] = adams_bashforth(f,0, 2,1.0,10);
#-------------------------------------------------------------

###===== Solution analytic, paramenters(x0,y0,h,N) ===========
[x4,y4] = analitica2(0, 2,1.0,10);
#-------------------------------------------------------------
```

```
### =========== Plotting =====================================
#color b for blue, '.' for point, + is type of symbol
plot(x,y,'b.+',x1,y1,'r.x',x2,y2,'g.*',x3,y3,'y.r',x4,y4,'m.o');
title('Solution of ODE');
xlabel('Axis (x)');ylabel('Axis (y)');
legend("Euler method","Taylor method","Mitpoint method",
"Adams method","Analytic solution","location","northwest");
legend("boxoff");
text(4,15.5,"y'(x)= y(x)/(1 + 3x)");#Text in plot
text(4,15.0,"Analytic solution is:");
text(4,14.5,'{y(x) = 2(1 + 3x)^{1/3}}');
text(4,14.0,"for y(0) = 2");
grid on
print -depsc plot4.eps #Figure in color eps format
#-------------------------------------------------------------
```

Resultado da plotagem o passo 1.0 é não uma boa escolha.

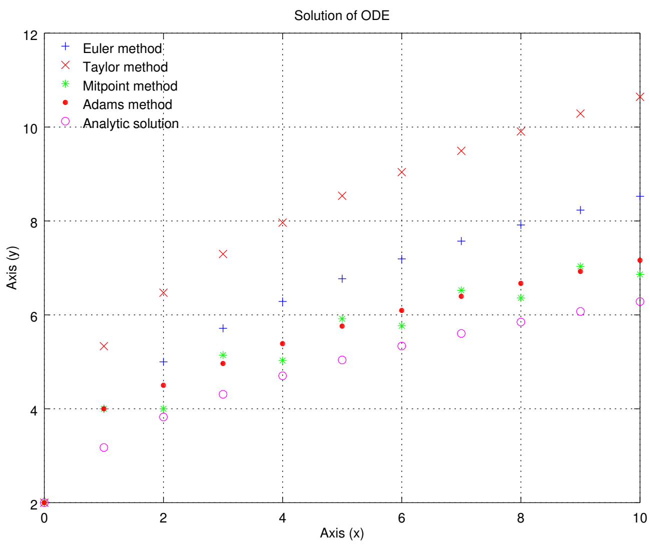
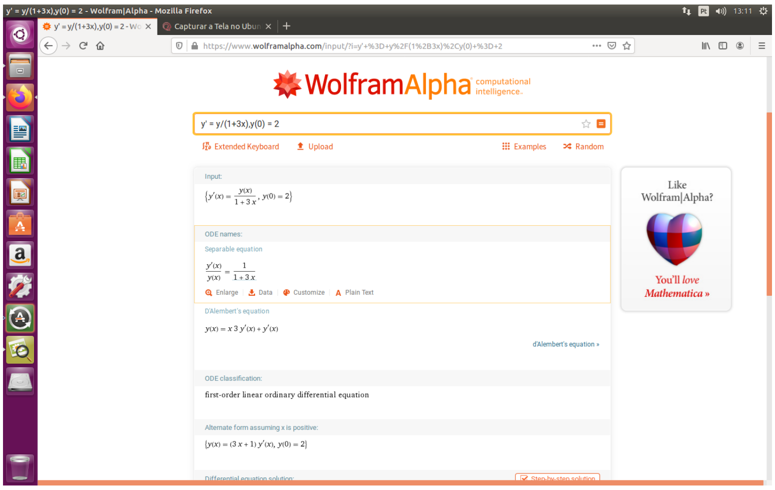
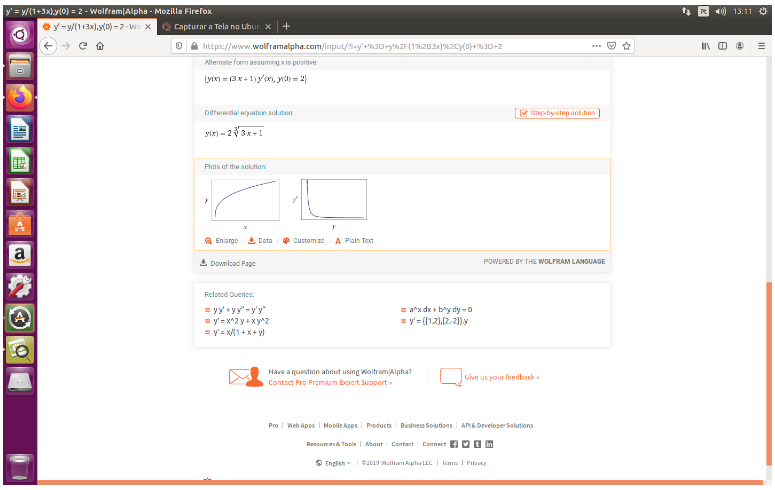


Figura 4: Comparação entre os métodos numéricos e a solução analítica.

Universidade Federal da Paraíba
Discente: Jefferson Bezerra dos Santos
Docente: José Miguel Aroztegui Massera

6. Questão:





7. Questão:

a) Temos que o *método do ponto médio* é dado por,

$$y_{n+2} = y_n + 2hf_{n+1}$$
ou
$$-y_n + 0y_{n+1} + y_{n+2} = h(0f_n + 2f_{n+1} + 0f_{n+2}).$$

Logo,

$$\alpha_0 = -1 \qquad \beta_0 = 0$$
$$\alpha_1 = 0 \qquad \beta_1 = 2$$
$$\alpha_2 = 1 \qquad \beta_2 = 0$$

segue que,

$$\begin{aligned}
c_0 &= \alpha_0 + \alpha_1 + \alpha_2 && \Rightarrow c_0 = -1 + 0 + 1 && \Rightarrow c_0 = 0\\
c_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) && \Rightarrow c_1 = 0 + 2 - (0 + 2 + 0) && \Rightarrow c_1 = 0\\
c_2 &= \frac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \frac{1}{1!}(\beta_1 + 2\beta_2) && \Rightarrow c_2 = \frac{1}{2}(0 + 4) - (2 + 0) && \Rightarrow c_2 = 0\\
c_3 &= \frac{1}{3!}(\alpha_1 + 2^3\alpha_2) - \frac{1}{2!}(\beta_1 + 2^2\beta_2) && \Rightarrow c_3 = \frac{1}{6}(0 + 8) - \frac{1}{2}(2 + 0) && \Rightarrow c_3 = \frac{1}{3}
\end{aligned}$$

portanto, o *método do ponto médio* possui ordem $q = 2$ com erro dado por $c_3 = \frac{1}{3}$.

b) Temos que o *método de Simpson* é dado por,

$$y_{n+2} = y_n + \frac{h}{3}[f_n + 4f_{n+1} + f_{n+2}]$$
ou
$$-y_n + 0y_{n+1} + y_{n+2} = h[\frac{1}{3}f_n + \frac{4}{3}f_{n+1} + \frac{1}{3}f_{n+2}]$$

Logo,

$$\alpha_0 = -1 \qquad \beta_0 = \frac{1}{3}$$
$$\alpha_1 = 0 \qquad \beta_1 = \frac{4}{3}$$
$$\alpha_2 = 1 \qquad \beta_2 = \frac{1}{3}$$

segue que,

$$
\begin{aligned}
c_0 &= \alpha_0 + \alpha_1 + \alpha_2 && \Rightarrow c_0 = -1 + 0 + 1 && \Rightarrow c_0 = 0 \\
c_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) && \Rightarrow c_1 = 0 + 2 - (\frac{1}{3} + \frac{4}{3} + \frac{1}{3}) && \Rightarrow c_1 = 0 \\
c_2 &= \frac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \frac{1}{1!}(\beta_1 + 2^1\beta_2) && \Rightarrow c_2 = \frac{1}{2}(0+4) - (\frac{4}{3} + \frac{2}{3}) && \Rightarrow c_2 = 0 \\
c_3 &= \frac{1}{3!}(\alpha_1 + 2^3\alpha_2) - \frac{1}{2!}(\beta_1 + 2^2\beta_2) && \Rightarrow c_3 = \frac{1}{6}(0+8) - \frac{1}{2}(\frac{4}{3} + \frac{4}{3}) && \Rightarrow c_3 = 0 \\
c_4 &= \frac{1}{4!}(\alpha_1 + 2^4\alpha_2) - \frac{1}{3!}(\beta_1 + 2^3\beta_2) && \Rightarrow c_4 = \frac{1}{24}(0+16) - \frac{1}{6}(\frac{4}{3} + \frac{8}{3}) && \Rightarrow c_4 = 0 \\
c_5 &= \frac{1}{5!}(\alpha_1 + 2^5\alpha_2) - \frac{1}{4!}(\beta_1 + 2^4\beta_2) && \Rightarrow c_5 = \frac{1}{120}(0+32) - \frac{1}{24}(\frac{4}{3} + \frac{16}{3}) && \Rightarrow c_5 = -\frac{1}{90}
\end{aligned}
$$

portanto, a ordem do *método de Simpson* é q = 4 e o erro é $c_5 = -\frac{1}{90}$.

c) Temos que o *método de Adams-Moulton* é dado por,

$$y_{n+2} = y_{n+1} + \frac{h}{12}[-f_n + 8f_{n+1} + 5f_{n+2}]$$

ou

$$0y_n - y_{n+1} + y_{n+2} = h[-\frac{1}{12}f_n + \frac{8}{12}f_{n+1} + \frac{5}{12}f_{n+2}]$$

Logo,

$$
\begin{aligned}
\alpha_0 &= 0 & \beta_0 &= -\frac{1}{12} \\
\alpha_1 &= -1 & \beta_1 &= \frac{8}{12} \\
\alpha_2 &= 1 & \beta_2 &= \frac{5}{12}
\end{aligned}
$$

segue que,

$$
\begin{aligned}
c_0 &= \alpha_0 + \alpha_1 + \alpha_2 && \Rightarrow c_0 = 0 - 1 + 1 && \Rightarrow c_0 = 0 \\
c_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) && \Rightarrow c_1 = -1 + 2 - (-\frac{1}{12} + \frac{8}{12} + \frac{5}{12}) && \Rightarrow c_1 = 0 \\
c_2 &= \frac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \frac{1}{1!}(\beta_1 + 2^1\beta_2) && \Rightarrow c_2 = \frac{1}{2}(-1+4) - (\frac{8}{12} + \frac{10}{12}) && \Rightarrow c_2 = 0 \\
c_3 &= \frac{1}{3!}(\alpha_1 + 2^3\alpha_2) - \frac{1}{2!}(\beta_1 + 2^2\beta_2) && \Rightarrow c_3 = \frac{1}{6}(-1+8) - \frac{1}{2}(\frac{8}{12} + \frac{20}{12}) && \Rightarrow c_3 = 0 \\
c_4 &= \frac{1}{4!}(\alpha_1 + 2^4\alpha_2) - \frac{1}{3!}(\beta_1 + 2^3\beta_2) && \Rightarrow c_4 = \frac{1}{24}(-1+16) - \frac{1}{6}(\frac{8}{12} + \frac{40}{12}) && \Rightarrow c_4 = -\frac{1}{24}
\end{aligned}
$$

portanto, a ordem do *método de Adams-Moulton* é q = 3 e o erro é $c_4 = -\frac{1}{24}$.

d) Temos que o *método de Adams-Bashforth* é dado por,

$$y_{n+2} = y_{n+1} + \frac{h}{2}[-f_n + 3f_{n+1}]$$
ou
$$0y_n - y_{n+1} + y_{n+2} = h[-\frac{1}{2}f_n + \frac{3}{2}f_{n+1}]$$

Logo,

$$\begin{aligned} \alpha_0 &= 0 & \beta_0 &= -\frac{1}{2} \\ \alpha_1 &= -1 & \beta_3 &= \frac{3}{2} \\ \alpha_2 &= 1 & \beta_2 &= 0 \end{aligned}$$

segue que,

$$\begin{aligned} c_0 &= \alpha_0 + \alpha_1 + \alpha_2 & &\Rightarrow c_0 = 0 - 1 + 1 & &\Rightarrow c_0 = 0 \\ c_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) & &\Rightarrow c_1 = -1 + 2 - (-\frac{1}{2} + \frac{3}{2} + 0) & &\Rightarrow c_1 = 0 \\ c_2 &= \frac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \frac{1}{1!}(\beta_1 + 2^1\beta_2) & &\Rightarrow c_2 = \frac{1}{2}(-1 + 4) - (\frac{3}{2} + 0) & &\Rightarrow c_2 = 0 \\ c_3 &= \frac{1}{3!}(\alpha_1 + 2^3\alpha_2) - \frac{1}{2!}(\beta_1 + 2^2\beta_2) & &\Rightarrow c_3 = \frac{1}{6}(-1 + 8) - \frac{1}{2}(\frac{3}{2} + 0) & &\Rightarrow c_2 = \frac{5}{12} \end{aligned}$$

portanto, a ordem do *método de Adams-Bashforth* é q = 2 e o erro é $c_4 = -\frac{5}{12}$.

8. Questão:
a) Vamos analizar a estabilidade do método dado por,

$$y_{n+2} - y_n = 2hf_{n+1}$$
ou
$$-y_n + 0y_{n+1} + y_{n+2} = h(f_n + 2f_{n+1} + 0f_{n+2}$$

logo,

$$\begin{aligned} \alpha_0 &= -1 & \beta_0 &= 0 \\ \alpha_1 &= \ \ 0 & \beta_1 &= 2 \\ \alpha_2 &= \ \ 1 & \beta_2 &= 0 \end{aligned}$$

segue que,

$$\begin{aligned}
p(\varepsilon) &= \sum_{j=0}^{k} \alpha_j \varepsilon^j \\
p(\varepsilon) &= \alpha_0 \varepsilon^0 + \alpha_1 \varepsilon^1 + \alpha_2 \varepsilon^2 \\
p(\varepsilon) &= \varepsilon^2 - 1 \Rightarrow \varepsilon = \pm 1
\end{aligned}$$

nota-se que as raízes são simples e possuem o modulo menor ou igual 1. Portanto, o método é estável.

Vamos analisar a consistência do método,

$$\begin{array}{lll}
c_0 = \alpha_0 + \alpha_1 + \alpha_2 & \Rightarrow c_0 = -1 + 0 + 1 & \Rightarrow c_0 = 0 \\
c_1 = \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) & \Rightarrow c_1 = 0 + 2 - (0 + 2 + 0) & \Rightarrow c_1 = 0 \\
c_2 = \dfrac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \dfrac{1}{1!}(\beta_1 + 2^1\beta_2) & \Rightarrow c_2 = \dfrac{1}{2}(0 + 4) - (2 + 0) & \Rightarrow c_2 = 0
\end{array}$$

nota-se que como $c_2 = 0$ o método possue ordem $q \geq 1$. Portanto é consistente, por fim como é estável e consistente, logo, é convergente.

b) Vamos analisar a estabilidade do método dado por,

$$y_{n+2} - y_{n+1} = \frac{h}{3}(-2f_n + 3f_{n+1})$$

ou

$$0y_n - y_{n+1} + 0y_{n+1} = h(-\frac{2}{3} + f_{n+1} + 0f_{n+2})$$

logo,

$$\begin{array}{ll}
\alpha_0 = \ \ 0 & \beta_0 = -\dfrac{2}{3} \\
\alpha_1 = -1 & \beta_1 = \ \ 1 \\
\alpha_2 = \ \ 1 & \beta_2 = \ \ 0
\end{array}$$

segue que,

$$\begin{aligned}
p(\varepsilon) &= \sum_{j=0}^{k} \alpha_j \varepsilon^j \\
p(\varepsilon) &= \alpha_0 \varepsilon^0 + \alpha_1 \varepsilon^1 + \alpha_2 \varepsilon^2 \\
p(\varepsilon) &= \varepsilon^2 - \varepsilon \Rightarrow p(\varepsilon) = \varepsilon(\varepsilon - 1) \Rightarrow \varepsilon = 0 \, \text{ou} \, 1
\end{aligned}$$

nota-se que todas as raízes possuem módulo menor ou igual a 1, além disso a raíz que tem módulo igual a 1 é simples, portanto, o método é estável.

Vamos analisar a consistência do método,

$$c_0 = \alpha_0 + \alpha_1 + \alpha_2 \quad \Rightarrow c_0 = 0 - 1 + 1 \quad \Rightarrow c_0 = 0$$
$$c_1 = \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) \quad \Rightarrow c_1 = -1 + 2 - (-\frac{2}{3} + 1 + 0) \quad \Rightarrow c_1 = \frac{2}{3}$$

ordem do método é $q = 0$, portanto o método não é consistente. Logo, não é convergente.

c)

$$y_{n+2} - y_{n+1} = \frac{h}{12}(-f_n + 8f_{n+1} + 4f_{n+2})$$

ou

$$0y_n - y_{n+1} + 0y_{n+1} = h(-\frac{1}{12}f_n + \frac{8}{12}f_{n+1} + \frac{4}{12}f_{n+2})$$

logo,

$$\alpha_0 = 0 \qquad \beta_0 = -\frac{1}{12}$$
$$\alpha_1 = -1 \qquad \beta_1 = \frac{8}{12}$$
$$\alpha_2 = 1 \qquad \beta_2 = \frac{4}{12}$$

Vamos analisar a estabilidade do método, segue que,

$$p(\varepsilon) = \sum_{j=0}^{k} \alpha_j \varepsilon^j$$
$$p(\varepsilon) = \alpha_0\varepsilon^0 + \alpha_1\varepsilon^1 + \alpha_2\varepsilon^2$$
$$p(\varepsilon) = \varepsilon^2 - \varepsilon \Rightarrow p(\varepsilon) = \varepsilon(\varepsilon - 1) \Rightarrow \varepsilon = 0 \,\text{ou}\, 1$$

portanto é estável. Vamos analisar a consistência do método,

$$c_0 = \alpha_0 + \alpha_1 + \alpha_2 \quad \Rightarrow c_0 = 0 - 1 + 1 \quad \Rightarrow c_0 = 0$$
$$c_1 = \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) \quad \Rightarrow c_1 = -1 + 1 - (-\frac{1}{12} + \frac{8}{12} + \frac{4}{12}) \quad \Rightarrow c_1 = \frac{11}{12}$$

ordem do método é $q = 0$, portanto o método não é consistente. Logo, não é convergente.

d)

$$y_{n+2} - y_n = \frac{h}{3}(-f_n + 4f_{n+1} + f_{n+2})$$

ou

$$-1y_n + 0y_{n+1} + y_{n+1} = h(\frac{1}{3}f_n + \frac{4}{3}f_{n+1} + \frac{1}{3}f_{n+2})$$

Vamos analisar a estabilidade do método, logo,

$$\begin{aligned} \alpha_0 &= -1 & \beta_0 &= \frac{1}{3} \\ \alpha_1 &= \ \ 0 & \beta_1 &= \ \ \frac{4}{3} \\ \alpha_2 &= \ \ 1 & \beta_2 &= \ \ \frac{1}{3} \end{aligned}$$

segue que

$$\begin{aligned} p(\varepsilon) &= \sum_{j=0}^{k} \alpha_j \varepsilon^j \\ p(\varepsilon) &= \alpha_0 \varepsilon^0 + \alpha_1 \varepsilon^1 + \alpha_2 \varepsilon^2 \\ p(\varepsilon) &= \varepsilon^2 - 1 \Rightarrow \varepsilon = \pm 1 \end{aligned}$$

portanto é estável.
Vamos analisar a consistência do método,

$$\begin{aligned} c_0 &= \alpha_0 + \alpha_1 + \alpha_2 & &\Rightarrow c_0 = -1 + 0 + 1 & &\Rightarrow c_0 = \ \ 0 \\ c_1 &= \alpha_1 + 2\alpha_2 - (\beta_0 + \beta_1 + \beta_2) & &\Rightarrow c_1 = 0 + 2 - (\frac{1}{3} + \frac{4}{3} + \frac{1}{3}) & &\Rightarrow c_1 = 0 \\ c_2 &= \frac{1}{2!}(\alpha_1 + 2^2\alpha_2) - \frac{1}{1!}(\beta_1 + 2^1\beta_2) & &\Rightarrow c_2 = \frac{1}{2}(0+2) - (\frac{4}{3} + \frac{2}{3}) & &\Rightarrow c_2 = -1 \end{aligned}$$

ordem do método é $q = 1$, portanto o método é consistente. Como é estável e consistente, portanto é convergente.

9. Questão:

Segue a implementação do método do previsor-corretor:

```
###==================== PROGRAM ============================
# Program:predCor9.m
# Description: Solve ODE numeric by the Previsor Corretor
# Date of Create: qui 31 out 2019 14:19:13
# Update in:  qui 31 out 2019 14:19:24
# Author:Jefferson Bezerra dos Santos
#-----------------------------------------------------------
### =========================== Clear enviromment ============================
clear
clc
#--------------------------------------------------------------------------------
```

```
###============ Metodo do Previsor Corretor =================
#================ Parametros ================================
tol = <+defina a tol+>;
h = <+ defina o passo+>;
xo = <+ defina o xo+>;
xf = <+ defina o xf+>;
y(1) = <+ defina o y0+>;
x = xo:h:xf;# Vetor x
f = @(x,y) y - x + 2;# Derivada
y(2) = y(1)+ h*f(x(1),y(1));# Aproxima de yn+1 por Euler
#------------------ Fim dos parametros ----------------------
###=============== Funcao anilitica ==========================
anali = @(x) <+ defina a funcao analitica+>;
#-----------------------------------------------------------

###================== Iniciando o loop =======================
i = 0;
while ((++i) < (length(x)-1))
        ym = y(i);#ym variavel auxiliar
        yn = y(i+1);#yn variavel auxiliar
    while (1)
        ymj = yn + (h/2)*((f(x(i),ym)) + 3*f(x(i+1),yn));#Previsor, adams
        aux = f((x(i+1)),ymj);# f
        ynj = ym + (h/3)*(f(x(i),ym) + 4*(f(x(i+1),yn)) + aux);#Corretor,simp
        if(abs(ynj-ymj) < tol)# Verificando tolerancia
            y(i+2) = ynj;
            break;
        else
            ym = yn;# Atualizando para a proxima iteracao
            yn = ynj;# Atualizando para a proxima iteracao
        endif
    endwhile
endwhile
#------------------- fim do laco ----------------------------
###============ Preenchimento do vetor y solucao analitica====
i = 0;
while ((++i) <= (length(x)))
    y1(i) = anali(x(i));
endwhile
#-----------------------------------------------------------
```

```
###================= Aplicando a funcao =====================
[x2,y2] = adamsbs (xo,xf,yo,h,f);
#--------------------------------------------------------------

###==================== Plotagem ============================
plot(x,y,'b.+',x,y1);
legend("previsor-corretor", "analitica", "adams");
#-------------------------------------------------------------
```

10. Questão:
Segue o código do método previsor-corretor, este é uma aplicação da questão 9, portanto, consulte o código da questão 9 para uma explicação detalhada.

```
###==================== PROGRAM ============================
# Program:prevCor9.m
# Description: Solve ODE numeric by the Previsor Corretor
# Date of Create: qui 31 out 2019 14:19:13
# Update in:  qui 31 out 2019 14:19:24
# Author:Jefferson Bezerra dos Santos
#------------------------------------------------------------
### ============================ Clear envirომment ============================
clear
clc
#--------------------------------------------------------------------------------

###========================== Parametros ======================================
tol = 5e-1;
h = 0.01;
xo = 0.0;
xf = 0.3;
yo = 2;
y(1) = 2;
x = xo:h:xf;
f = @(x,y) y - x + 2;
y(2) = y(1)+ h*f(x(1),y(1));
#--------------------------------------------------------------------------------

###==================== Function Adasm Bashforth ===============================
function [x,y] = adamsbs (xo,xf,yo,h,f)
    x = xo:h:xf;
    y(1)= yo;
    y(2) = y(1)+ h*f(x(1),y(1));
```

```
    i = 0;
    while ((++i) < (length(x)-1))
        y(i+2) = y(i) + (h/2)*((f(x(i),y(i))) + 3*f(x(i+1),y(i+1)));
    endwhile
endfunction
#---------------------- End function -----------------------------

### ================== Solution analytic =========================
anali = @(x) 3*exp(x) + x - 1;
#------------------------------------------------------------------

#loop
i = 0;
while ((++i) < (length(x)-1))
        ym = y(i); # variavel auxiliar
        yn = y(i+1);#variavel auxiliar
    while (1)
        ymj = yn + (h/2)*((f(x(i),ym)) + 3*f(x(i+1),yn));#preditor adasm bashfor
        aux = f((x(i+1)),ymj);# funcao f
        ynj = ym + (h/3)*(f(x(i),ym) + 4*(f(x(i+1),yn)) + aux);#Corretor simpson
        if(abs(ynj-ymj) < tol)# Verificando a tolerancia foi atingida
            y(i+2) = ynj;
            break;
        else
            ym = yn;#Atualizando para proxima iteracao
            yn = ynj;#Atuaizando para proxima iteracao
        endif
    endwhile
endwhile

#loop para prencher solucao analitica
i = 0;
while ((++i) <= (length(x)))
    y1(i) = anali(x(i));
endwhile

###============= Adams bashfort =============================
[x2,y2] = adamsbs (xo,xf,yo,h,f);
#-----------------------------------------------------------

###================= Plotagem ===============================
```

```
plot(x,y,'b.+',x,y1,'r.x',x2,y2,'g.o');
title('Solution of ODE');
xlabel('Axis (x)');ylabel('Axis (y)');
legend("previsor-corretor", "analitica", "adams");
legend("boxoff");
grid on
print -depsc plot6.eps #Figure in color eps format
#-----------------------------------------------------------
```