Vinícius Esperança Motoori, 247395;
Atividade 4, MS211, Turma C.
Etrega: 25/11/2023;

Exercício 1)

- Algoritmos:
  - Método de Euler Explícito:

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

↳ Onde $f(x,y) = y'$, y_i a iteração anterior e, y_{i+1} a iteração atual (valor que procuramos no momento); h é $x_1 - x_0$, deve ser maior que 0 e constante!

↳ Algoritmo de Euler Explícito:

```
Function eE (x0, y0, h, n)  ⟶ número de iterações
   Initialize vector x, y with n+1 Elements
   x[0] ← x0, y[0] ← y0
   For i ← 0 to n-1 do
      y[i+1] ← y[i] + h.f(xi, yi)
      x[i+1] ← x[i] + h
   End For
   Return x, y
End Function
```

Obs: Todos os códigos estão na pasta "ativ_4"

* Compilar com "gcc -o <nome do programa> <nome do programa.c>"

Continuação Exercício 1)

- Método de Crank-Nicolson:

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_{i+1}, y_{i+1}) + f(x_i, y_i)\right)$$

↳ Para fazer tal algoritmo, podemos reutilizar o de Euler Explícito e incrementá-lo com Euler Implícito, conforme feito no código em "ativ_4". Porém, para simplificar o pseudo-código, façamos da seguinte maneira:

↳ Algoritmo de Crank-Nicolson:

```
Function CN(x0, y0, h, n)
    Initialize vector X, Y with n+1 Elements
    X[0] <- x0 , Y[0] <- y0
    For i <- 0 to n-1 do
        Y[i+1] <- Y[i] + h/2 (f(X[i] + h, Y[i+1]) + f(X[i], Y[i]))
        X[i+1] <- X[i] + h
    End For
    Return X, Y
End Function
```

* Código em "ativ_4"!

Compilar com: " gcc -o <nome> <nome.c>

↱ Do programa

Continuação Exercício 1)

↳ Desenvolvendo a função a ser utilizada para iteração:

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}\left(f(x_i+h, y_{i+1}) + f(x_i, y_i)\right) \\ y' = f(x,y) = 1-2y \end{cases}$$

$$\hookrightarrow y_{i+1} = y_i + \frac{h}{2}\left(1-2y_{i+1} + 1-2y_i\right)$$

$$\hookrightarrow y_{i+1} = y_i - h y_i - h y_{i+1} + h$$

$$\hookrightarrow (1+h)\,y_{i+1} = (1-h)\,y_i + h$$

$$\hookrightarrow y_{i+1} = \frac{(1-h)\,y_i + h}{1+h}$$

-> Função usada no código!

- método de Euler aperfeiçoado:

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, y_i + h f(x_i, y_i))\right)$$

↳ algoritmo de Euler aperfeiçoado:

```
Function eA(x0, y0, h, n)
    Initialize Vector x, y with n+1 Elements
    x[0] ← x0 , y[0] ← y0
    For i ← 0 to n-1 do
        y[i+1] ← y[i] + h/2 ( f(x[i], y[i]) + f(x[i+1], y[i] + h f(x[i], y[i]))
        x[i+1] ← xi + h
    End For
End Function
```

* código em "atim_4"

Compilar com " gcc -o <name> <name.c>

Continuação Exercício 1)

Como $f(x,y)$ não depende de x, temos que

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i,y_i) + f(x_{i+1}, hf(x_i,y_i) + y_i\right)$$

depende somente de y_i e, por consequência, usar-se-á essa expressão diretamente no código.

$$y_{i+1} = y_i + \frac{h}{2}\left(1-2y_i + 1-2\left(h(1-2y_i) + y_i\right)\right)$$

• -Escolhendo h para que Euler Explícito convirja

Para que o método de Euler Explícito convirja precisamos que a condição de estabilidade seja satisfeita, ou seja, que valha:

$$\left|1 + h\frac{\partial f}{\partial y}(x,y)\right| < 1$$

Então, desenvolvendo:

$$|1 + h(-2)| < 1$$

Como $h > 0 \rightarrow 0 < h < 1$

Então, usaremos $h = \frac{1}{2} = 0,5$ para o método de Euler Explícito;

Continuação Exercício 1)

- Quanto ao método de Crank-Nicolson, sabemos que ele converge para qualquer valor de h, uma vez que é um método implícito!

↳ Logo, por simplicidade, usaremos $h = 0,5$ também.

- Por fim, tratando do método de Euler, sabemos que é um método explícito, então, como fizemos para encontrar o intervalo de valores que h pode assumir, já sabemos que $0 < h < 1$, logo, usaremos $h = 0,5$ também, por simplicidade e para facilitar a comparação entre métodos.

• Usando os códigos escritos para cada um dos métodos, tem-se:

- Euler Explícito: $Y(3) = 0,5000$
- Crank-Nicolson: $Y(3) = 0,5001$
- Euler aperfeiçoado: $Y(3) = 0,5020$

Continuação Exercício 1)

* Sobre códigos e resultados:

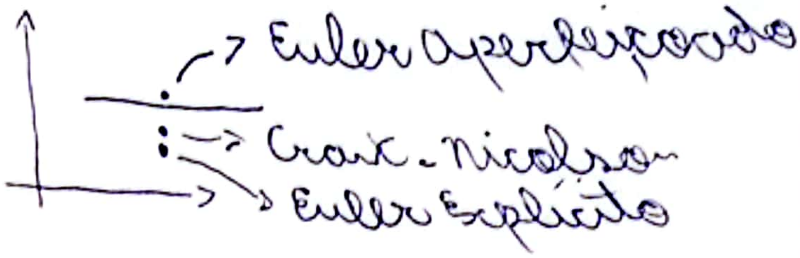
- Inicialmente, vale ressaltar que, para além do resultado de Y(3), para cada um dos métodos, foram obtidos os pontos (X,Y) de cada passo e, esses pontos podem ser vistos nas imagens em "ativ_4", cujos nomes começam com "dados". Além disso, todos os códigos enviados, se não forem alterados, dão os mesmos resultados de suas respectivas imagens de dados. Estes códigos podem ser executados usando "gcc -o <Nome> <Nome.c>" e, em seguida "./<Nome>", onde "<Nome>" deve ser trocado pelo nome do programa! Por exemplo: "gcc -o eulerAperfeicoado eulerAperfeicoado.c" ; "./eulerAperfeicoado".

• Discussão dos resultados:

- Para analisar de maneira satisfatória os resultados, tracei a curva ⓐ no aplicativo "geogebra" e sinalizei os pontos (3, Y(3)) dados pelos três métodos. Desse modo, o resultado foi salvo em "ativ_4" na imagem "comparacao_metodos_ponto_x_igual_3" e, foi

Continuação Exercício 1)

... algo como:



Esse resultado reafirmou o que esperava, os métodos comparados por sua precisão podem ser ordenados do seguinte modo:

Euler Explícito < Crank-Nicolson < Euler Aperfeiçoado

Isso coincide com as minhas expectativas, uma vez que o método de Euler explícito é o mais simples, usando apenas Y_i e $f(x_i, Y_i)$ para obter o Y_{i+1}, enquanto que o de Crank-Nicolson faz uma média entre Euler Implícito e Explícito, o que significa, geometricamente, que ele fica entre os resultados desses dois métodos, sendo, portanto, mais preciso, conforme o desenho ilustrativo:

Continuação Exercício 1)

Além disso, era possível, também, esperar que o método de Euler aperfeiçoado fosse ainda mais preciso que o de Nicolson, pois ao usar uma média entre Euler Explícito e Implícito, o método de Nicolson acaba tentando aproximar pelo meio entre os pontos gerados por tais métodos, enquanto que o método de Euler aperfeiçoado funciona quase como uma previsão da próxima iteração. Isso porque, ao fazer a média entre $f(x_i, y_i)$ e $f(x_{i+1}, hf(x_i, y_i) + y_i)$, o método de Euler aperfeiçoado se dá o luxo de usar o que seria uma iteração do método de Euler $(y_i + hf(x_i, y_i))$ para antecipar um resultado de y e usá-lo como forma de melhorar sua precisão para o próximo ponto y. (y_{i+1}).

Em suma, analisando não somente o ponto $(3, y)$ da curva obtida pelos métodos, mas todos os pontos, pode-se concluir, em concordância com a teoria e minhas expectativas, que o melhor método é o de Euler aperfeiçoado, seguido pelo de Crank-Nicolson e, finalmente pelo Euler Explícito.

Continuação Exercício 1)

- Para encontrarmos uma solução para o problema ⓑ, façamos:

$$\begin{cases} 2y'' - y' - 3y + 2 = 0 \\ y(2) = 1 \\ y(4) = -1 \end{cases} \longrightarrow h = \frac{4-2}{n} = \frac{2}{n}$$

, como sabemos os valores em $x=2$ e $x=4$, temos de determinar apenas as aproximações internas a este intervalo;

Aproximando y'': $D_2 y(x_i) = \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$

aproximando y': $D_0 y(x_i) = \dfrac{y_{i+1} - y_{i-1}}{2h}$

↳ Temos o seguinte método de Diferenças finitas:

$$2 \cdot \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - \frac{y_{i+1} - y_{i-1}}{2h} - 3y_i + 2 = 0$$

Para o primeiro passo, teríamos:

$$2 \cdot \frac{y_2 - 2y_1 + y_0}{h^2} - \frac{y_2 - y_0}{2h} - 3y_1 + 2 = 0$$

↳ multiplicando por h^2:

$$2(y_2 - 2y_1 + y_0) - \frac{h}{2}(y_2 - y_0) - 3h^2 y_1 + 2h^2 = 0$$

Continuação Exercício 1)

... ↳ agrupando:

$y(2) = 1$

$$\left(2-\frac{h}{2}\right)y_2 - (2+3h^2)\,y_1 + \left(2-\frac{h}{2}\right)y_0 + 2h^2 = 0$$

↳ temos equações entre a inicial e a final do tipo:

$$\left(2-\frac{h}{2}\right)y_{i+1} - (2+3h^2)\,y_i + \left(2-\frac{h}{2}\right)y_{i-1} = -2h^2; \quad \forall i: 1<i<n-1$$

↳ mas, para a primeira, temos: $y_0 = 1 = y(2)$

$$\left(2-\frac{h}{2}\right)y_2 - (2+3h^2)\,y_1 = -2h^2 - \left(2-\frac{h}{2}\right)$$

E, para a última: $y_n = -1 = y(4)$

$$-(2+3h^2)\,y_{n-1} + \left(2-\frac{h}{2}\right)y_{n-2} = -2h^2 + \left(2-\frac{h}{2}\right)$$

↳ Assim, temos a seguinte matriz de coeficientes:

$$A = \begin{bmatrix} -(2+3h^2) & (2-\frac{h}{2}) & & & \\ (2-\frac{h}{2}) & -(2+3h^2) & (2-\frac{h}{2}) & & \\ & \ddots & \ddots & \ddots & \\ & & (2-\frac{h}{2}) & -(2+3h^2) & (2-\frac{h}{2}) \\ & & & (2-\frac{h}{2}) & -(2+3h^2) \end{bmatrix}$$

Logo, usaremos o método de Thomas para sistemas tridiagonais;

Continuação Exercício 1:

Temos: $L = \begin{pmatrix} \alpha_1 & c_1 & & & \\ & \alpha_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{n-2} & c_{n-2} \\ & & & & \alpha_{n-1} \end{pmatrix}$, $\alpha_1 = -(2+3h^2)$

$U = \begin{bmatrix} 1 & & & \\ \beta_2 & 1 & & \\ & \ddots & \ddots & \\ & & \beta_{n-1} & 1 \end{bmatrix}$, $\beta_i = \dfrac{(2-\frac{h}{2})}{\alpha_{i-1}}$

$\hookrightarrow \alpha_i = -(2+3h^2) - \beta_i\,(2-\frac{h}{2})$

Usando $n=5$, temos $h = 0{,}4$ e:

$\alpha_1 = -(2+3.0{,}16) = -2{,}48$

$\beta_1 = -\dfrac{\alpha_1 + (2+3.0{,}16)}{(2-0{,}2)} = 0$

$\beta_2 = \dfrac{(2-0{,}2)}{\alpha_1} = -0{,}7258$

$\alpha_2 = -(2+3.0{,}16) + 0{,}7258.(2-0{,}2) = -1{,}1736$

$\beta_3 = \dfrac{2-0{,}2}{-1{,}1736} = -1{,}5337$

$\alpha_3 = -2{,}48 + 1{,}5337(1{,}8) = 0{,}28$

$\beta_4 = \dfrac{1{,}8}{0{,}28} = 6{,}4286$

$\alpha_4 = -2{,}48 + 6{,}4286(1{,}8) = 9{,}0915$

$\beta_5 = 0{,}1981$

$\alpha_5 = -2{,}83658$

$\beta_6 = -0{,}6346$

$\alpha_6 = -1{,}3377$

Calculando valores de y:

Continuação Exercício 1)

$$b = \begin{pmatrix} -2h^2 - \left(2 - \frac{h}{2}\right) \\ \vdots \\ -2h^2 + \left(2 - \frac{h}{2}\right) \end{pmatrix} = \begin{pmatrix} -2,12 \\ \vdots \\ 1,48 \end{pmatrix}$$

Para salvarmos a progressão:

$$\frac{2\left(2 - \frac{h}{2}\right)}{5} = K = 0,72$$

$$\hookrightarrow b = \begin{pmatrix} -2,12 \\ -1,4 \\ -0,68 \\ -0,04 \\ 0,76 \\ 1,48 \end{pmatrix} \rightarrow \begin{array}{l} X_1 = -2,12 \\ X_2 = -1,4 - 0,7258 \cdot 2,12 = -2,938 \\ X_3 = -0,68 - 1,5337 \cdot 2,9386 = -5,1870 \\ X_4 = -0,04 + 5,1870 \cdot 6,4286 = 33,305 \\ X_5 = 0,76 - 0,1981 \cdot 33,305 = -5,8377 \\ X_6 = 1,48 - 0,6346 \cdot 5,8377 = -2,2246 \end{array}$$

$\hookrightarrow$ Por fim:

$$\bar{Y}_5 = \frac{-2,2246}{-1,3377} = 1,6630 \rightarrow \bar{Y}_0 =$$

$$\bar{Y}_4 = -\frac{5,8377 - 1,8 \cdot 1,6630}{-2,83658} = 3,11329$$

$$\bar{Y}_3 = \frac{33,305 - 1,8 \cdot 3,1133}{9,0915} = 3,0470$$

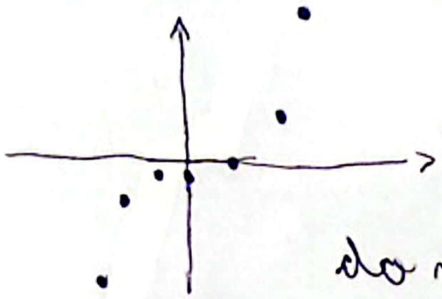
$$\bar{Y}_2 = \frac{-5,1870 - 1,8 \cdot 3,0470}{0,28} = -38,1129$$

$$\bar{Y}_1 = \frac{-2,9386 + 1,8 \cdot 38,1129}{-1,1736} = -55,9515$$

Exercício 2)

Colocando os pontos em um sistema de coordenadas cartesianas X Y, temos o que se encontra em "ativ_4", no arquivo "numem.png"

Essa imagem é algo como:



Logo, podemos perceber que a melhor forma de aproximar o comportamento do numem de dados por um $\varphi(x) = y$ do tipo linear é com as funções g_i compondo uma função do terceiro grau. No caso, seria prudente usarmos, o que se segue como função $g_i(x)$, então, teríamos:

$$g_1(x) = x^3, \quad g_2(x) = x^2, \quad g_3(x) = x, \quad g_4(x) = 1$$

Assim, temos: $y = \varphi(x) = C_1 x^3 + C_2 x^2 + C_3 x + C_4$.

Portanto, organizando a resposta:

Usa-se: $g_1(x) = x^3$, $g_2(x) = x^2$, $g_3(x) = x$, $g_4(x) = 1$;

$n = 4$;

Continuação Exercício 2)

Temos:

$$V = \{ v_0 = (1,1,1,1,1,1,1)^t ; \ v_1 = (-3,-2,-1,0,1,2,3)^t ;$$

$$v_2 = (9,4,1,0,1,4,9)^t ; \ v_3 = (-27,-8,-1,0,1,8,27)^t \}$$

$\langle v_0, v_0 \rangle = 7$ $\quad \langle v_1, v_1 \rangle = 28$ $\quad \langle v_2, v_3 \rangle = 0$

$\langle v_0, v_1 \rangle = 0$ $\quad \langle v_1, v_2 \rangle = 0$ $\quad \langle v_3, v_3 \rangle = 1588$

$\langle v_0, v_2 \rangle = 28$ $\quad \langle v_1, v_3 \rangle = 196$

$\langle v_0, v_3 \rangle = 0$ $\quad \langle v_2, v_2 \rangle = 196$

$$\begin{bmatrix} 7 & 0 & 28 & 0 \\ 0 & 28 & 0 & 196 \\ 28 & 0 & 196 & 0 \\ 0 & 196 & 0 & 1588 \end{bmatrix} \begin{bmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \end{bmatrix} = \begin{bmatrix} -13{,}8 \\ 391{,}2 \\ -54 \\ 3171{,}6 \end{bmatrix} \rightarrow \begin{cases} 7c_4 + 28c_2 = -13{,}8 \\ 28c_3 + 196c_1 = 391{,}2 \\ 28c_4 + 196c_2 = -54 \\ 196c_3 + 1588c_1 = 3171{,}6 \end{cases}$$

Resolvendo:

$$c_4 = -\frac{71}{35}, \quad c_3 = -\frac{17}{252}, \quad c_2 = \frac{1}{70}, \quad c_1 = \frac{361}{180}$$

$$\hookrightarrow \boxed{Y = \varphi = \frac{361}{180}x^3 + \frac{1}{70}x^2 - \frac{17}{252}x - \frac{71}{35}}$$