Lucca Ferreira Paiva 240229

Questao 1)

A)

$$1)\ \begin{cases} -(1-0{,}09(5-x))c''(x) - (0{,}2-0{,}01(5-x))c'(x) + 0{,}05c(x) = f(x) \\ c(0)=0 \quad c'(5)=0 \end{cases}$$

$$f(x) = 0{,}025$$

$-\alpha = -(1-0{,}09(5-x)) \rightarrow$ difusibilidade

$v = -(0{,}2-0{,}01(5-x)) \rightarrow$ velocidade (p/ baixo)

$\mu = 0{,}05 \rightarrow 5\%$ decaimento

Método Numérico

$$c_i'' = \frac{c_{i-1} - 2c_i + c_{i+1}}{h^2} \qquad c_i' = \frac{c_{i+1} - c_{i-1}}{2h}$$

$$-\alpha\left(\frac{c_{i-1} - 2c_i + c_{i+1}}{h^2}\right) + v\left(\frac{c_{i+1} - c_{i-1}}{2h}\right) + \mu c_i = f_i \qquad h = \frac{x_{max}}{n}$$

$[0, 5]$

$c(0) = c(x_0) = 0 = c_0$

$$\left(\frac{-\alpha}{h^2} - \frac{v}{2h}\right)c_{i-1} + \left(\frac{2\alpha}{h^2} + \mu\right)c_i + \left(\frac{-\alpha}{h^2} + \frac{v}{2h}\right)c_{i+1} = f_i \qquad i = 1, 2, 3, \ldots$$

$\vec{c} = [c_0, c_1, c_2, \ldots, c_{n-1}, c_n]$

$i = 0 \rightarrow c_0 = 0$

$$\left(\frac{-\alpha}{h^2} - \frac{v}{2h}\right)c_0 \rightarrow 0 + \left(\frac{2\alpha}{h} + \mu\right)c_1 + \left(\frac{-\alpha}{h^2} + \frac{v}{2h}\right)c_2 = f$$

$i = 1, 2, \ldots, n-1$

$$\left(\frac{-\alpha}{h^2} - \frac{v}{h}\right)c_{i-1} + \left(\frac{2\alpha}{h^2} + \mu\right)c_i + \left(\frac{-\alpha}{h^2} + \frac{v}{2h}\right)c_{i+1} = f$$

$$i = n \left(\frac{-\alpha}{h^2} - \frac{v}{h}\right)c_{n-1} + \left(\frac{2\alpha}{h^2} + \mu\right)c_n + \left[\frac{-\alpha}{h^2} + \frac{v}{2h}\right]c_{n-1} = f$$

$i = n$

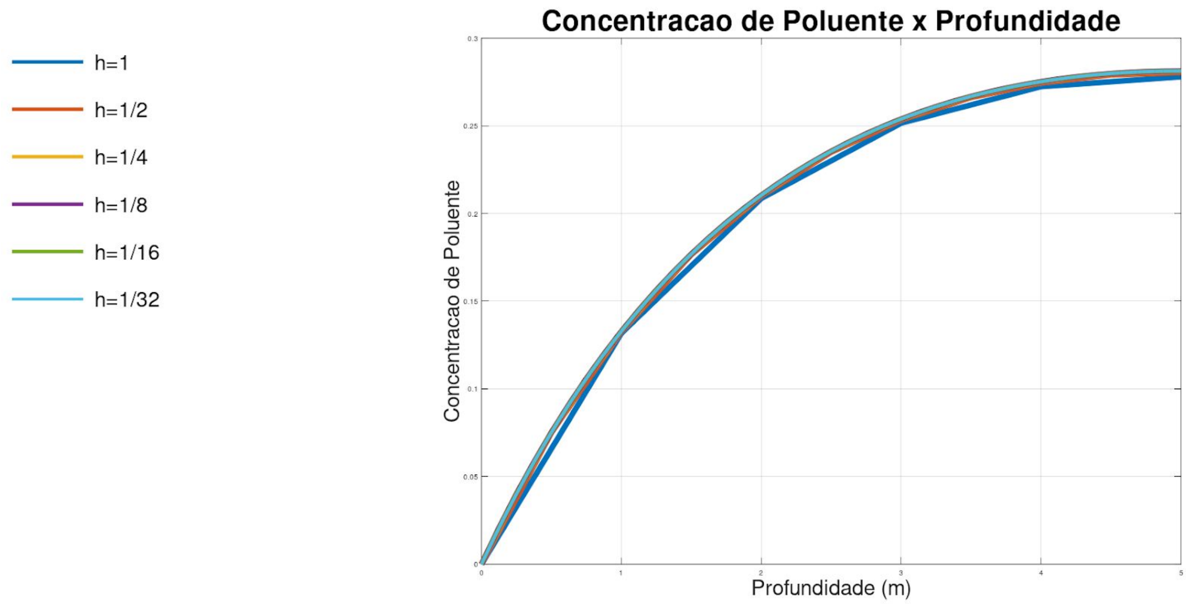
$$\left[\frac{-2\alpha}{h^2}\right] c_{n-1} + \left[\frac{2\alpha}{h^2} + \mu\right] c_n = f$$

$$\begin{cases} (2\alpha + \mu h^2)\, c_1 + \left(-\alpha + v\frac{h}{2}\right) c_2 = fh^2 \\ (-\alpha - vh/2)\, c_1 + (2\alpha + \mu h^2)\, c_2 + (-\alpha + vh/2)\, c_3 = fh^2 \\ \vdots \\ (-2\alpha)\, c_{n-1} + (2\alpha + \mu h^2)\, c_n = fh^2 \end{cases}$$

$$\begin{bmatrix} 2\alpha + \mu h^2 & -\alpha + vh/2 & 0 & 0 & \cdots & 0 & 0 \\ -\alpha - vh/2 & 2\alpha + \mu h^2 & -\alpha + vh/2 & 0 & \cdots & 0 & 0 \\ 0 & -\alpha - vh/2 & 2\alpha + \mu h^2 & -\alpha + vh/2 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ & & \vdots & & \ddots & \vdots & \\ 0 & 0 & \cdots & 0 & -\alpha - vh/2 & 2\alpha + \mu h^2 & -\alpha + v\frac{h}{2} \\ 0 & 0 & \cdots & 0 & 0 & -2\alpha & 2\alpha + \mu h^2 \end{bmatrix} \cdot$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} fh^2 \\ fh^2 \\ fh^2 \\ \vdots \\ \\ fh^2 \\ fh^2 \end{bmatrix}$$

B)



Concentracao de Poluentes [0, 0.3] x Profundidade [0, 5]

C)

É possível verificar claramente a curva para h = 1, isso porque ela é mais “dura”, ou seja, como o h e muito grande, as transições entre os pontos ficam muito óbvias. Porém, para h menores do que ½, ja é dificil identificar esses pontos, e ar curvas ja ficam bem mais suaves. Para h = 1/32 os pontos já são imperceptíveis.

```
h = 1;
[x1, y1] = Sis2(h);

h = 1/2;
[x2, y2] = Sis2(h);

h = 1/4;
[x3, y3] = Sis2(h);

h = 1/8;
[x4, y4] = Sis2(h);

h = 1/16;
[x5, y5] = Sis2(h);

h = 1/32;
[x6, y6] = Sis2(h);
```

```
function [xs, ys] = Sis2 (h)
  n = 5/h;

  x = linspace(h, 5, n);

  A = -(1 - 0.09*(5 - x));
  B = -(0.2 - 0.01*(5 - x));
  C = 0.05;
  f = 0.025*ones(n, 1);

  mat = zeros(n , n);
  mat(1,1) = h^2*C(1)-2*A(1);
  mat(1,2) = A(1)+h*B(1)/2;
  for k = 2:n-1
    mat(k,k-1) = A(k) - h*B(k)/2;
    mat(k,k) = h^2*C - 2*A(k);
    mat(k,k+1) = A(k) + h*B(k)/2;
  end
  mat(n,n-1) = 2*A(n);
  mat(n,n) = h^2*C-2*A(n);
  g = h^2*f;

  xs = [0, x];
  ys = [0; mat\g];
endfunction
```

Questao 2)

Polinômio de grau 7

```
x = [0 6 14 21 28 35 49 56]';


n = 8;

y = [2546 2347 2219 2261 2294 2821 3118 2899]';

P = polyfit(x,y,length(x));

T = TabelaDifDiv(x, y);

pre = output_precision(10);

([x,T]);

P2 = @(k) (2546 + -33.1666666667*(k - 0) + 0.9861111111*(k - 0)*(k - 6)  + 0.0355820106*(k - 0)*(k - 6)*(k - 14) +
-0.0040798074*(k - 0)*(k - 6)*(k - 14)*(k - 21) + 0.0004098557*(k - 0)*(k - 6)*(k - 14)*(k - 21)*(k - 28) +
-0.0000195348*(k - 0)*(k - 6)*(k - 14)*(k - 21)*(k - 28)*(k - 35) +
0.0000007157*(k - 0)*(k - 6)*(k - 14)*(k - 21)*(k - 28)*(k - 35)*(k - 49));

Erro = @(k)  (k - 0)*(k - 6)*(k - 14)*(k - 21)*(k - 28)*(k - 35)*(k - 49);


r1 = polyval(P, 42);
r2 = P2(42);

E = abs(Erro(42));
```

Tabela dos Coeficientes

coeficientes

X	Aproximacao de Ordem 0 Y0	Aproximacao de Ordem 1	Aproximacao de Ordem 2	Aproximacao de Ordem 3	Aproximacao de Ordem 4	Aproximacao de Ordem 5	Aproximacao de Ordem 6	Aproximacao de Ordem 7
0.0000000000	2546.0000000000	-33.1666666667	0.9861111111	0.0355820106	-0.0040798074	0.0004098557	-0.0000195348	0.0000007157
6.0000000000	2347.0000000000	-21.3333333333	1.7333333333	-0.0786525974	0.0102651422	-0.0005473502	0.0000205462	0.0000000000
12.0000000000	2219.0000000000	4.6666666667	0.0029761905	0.2190365277	-0.0132709161	0.0004799623	0.0000000000	0.0000000000
21.0000000000	2261.0000000000	4.7142857143	5.0408163265	-0.2719873664	0.0078474247	0.0000000000	0.0000000000	0.0000000000
28.0000000000	2294.0000000000	75.2857142857	-2.5748299320	0.0026724976	0.0000000000	0.0000000000	0.0000000000	0.0000000000
35.0000000000	2821.0000000000	21.2142857143	-2.5000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000
49.0000000000	3118.0000000000	-31.2857142857	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000
56.0000000000	2899.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000

Polinômio de Grau 3

```
a = [28 35 49 56]';

n = 4;

b = [2294 2821 3118 2899]';

R = polyfit(a,b,length(a));

T2 = TabelaDifDiv(a, b);

pre = output_precision(10);

([a,T2]);

R2 = @(k)  2294 + 75.2857142857*(k - 28) + -2.5748299320*(k - 28)*(k - 35) + 0.0026724976*(k - 28)*(k - 35)*(k - 49);

Erro3 = @(k) 0.0026724976*(k - 28)*(k - 35)*(k - 49);

r3 = polyval(R, 42);
r4 = R2(42);
```

Tabela dos Coeficientes

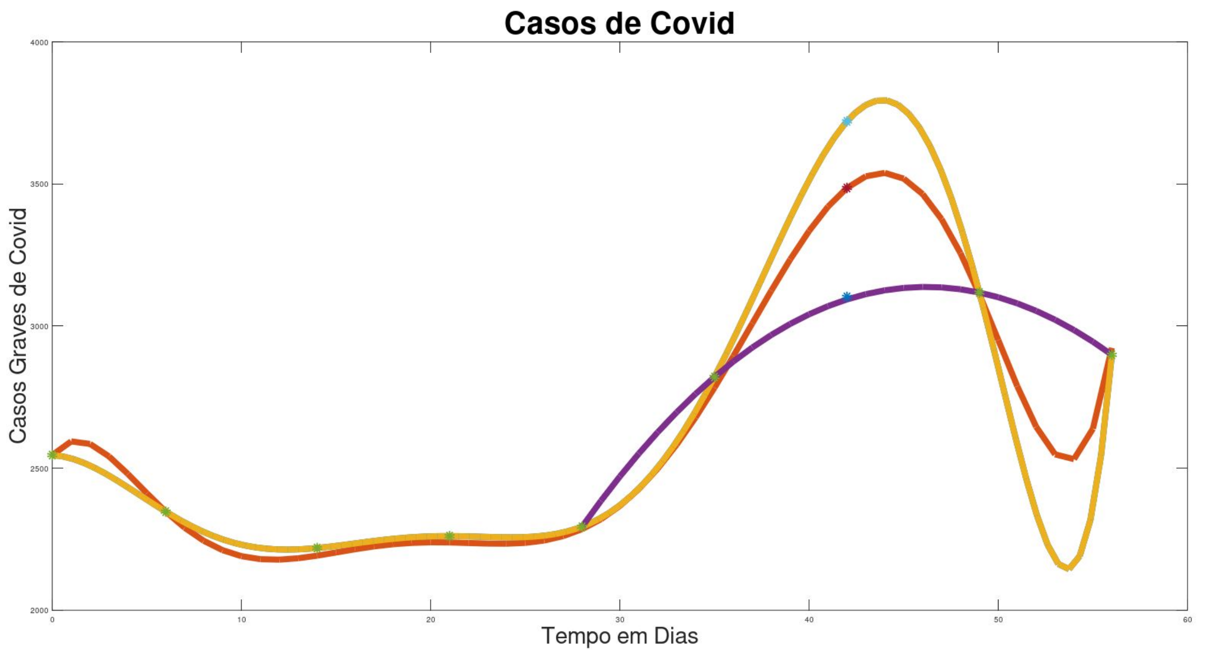
coeficintes

x	Ordem 0	Ordem 1	Ordem 2	Ordem 3
28.0000000000	2294.0000000000	75.2857142857	-2.5748299320	0.0026724976
35.0000000000	2821.0000000000	21.2142857143	-2.5000000000	0.0000000000
49.0000000000	3118.0000000000	-31.2857142857	0.0000000000	0.0000000000
56.0000000000	2899.0000000000	0.0000000000	0.0000000000	0.0000000000

Quanto ao número de casos

```
Erro 1: 609892416.000000
Erro 2: 1.833333
Numero de casos dado por P1: 3720.575339
Numero de casos dado por P2: 3093.833333
Numero de casos dado por F1: 3486.090067
Numero de casos dado por F2: 3102.637569
```

Em que, P1 representa o polinômio de grau 7 encontrado pela forma de Newton (Amarelo), e F1 o polinômio de grau 7 encontrado pelo octave (Laranja). De forma similar, P2 representa o polinômio de grau 3 encontrado pela forma de Newton (Roxo), e F2 o polinômio de grau 3 encontrado pelo octave. Portanto, o número de casos no dia 15/11 foi de aproximadamente 3094 pessoas



Casos [0, 4000] x Tempo [0, 60]

- Curva Laranja: Polinômio gerado pelo octave de grau 7
- Curva Amarelo: Polinômio gerado pela Forma de Newton de grau 7
- Curva Roxa: Polinômio gerado pelo método de Newton de grau 3 (Ps: coincide com o polinômio gerado pelo octave)
- Pontos verdes: Pontos base dados
- Pontos azuis e vermelho: Número de casos de covid para cada polinômio

Pelo gráfico de número de casos graves pelo tempo em dias podemos ver claramente essa diferença. (Obs talvez não dê pra ler os números) Em que o tempo representa o número de dias decorridos desde 4/10, e portanto, 15/11 o nosso dia a ser estimado corresponde ao dia 42.

Nele podemos ver claramente a diferença entre esses gráficos. Primeiro, vale reparar a diferença entre os dois de grau maiores, embora o gerado pela função nativa do octave acabe se aproximando da curva roxa, que apresenta um erro menor, vale notar que ele não passa pelos pontos dados, o que o Amarelo faz. Isso provavelmente se deve a utilização de algum método com mínimos quadrados ou algo similar.

Além disso, vale notar como a curva roxa, de grau 3, assim como a amarela, passam claramente pelos pontos base, mesmo que no intervalo de intersecção o comportamento das duas seja bem diferente.

O erro de ambas as funções foi calculado pela fórmula para a interpolação pelo máximo do valor absoluto das diferenças divididas de ordem n+1, ou seja, o produtório de x - xk, para k de 0 até n + 1, multiplicado pelo maior valor dessa ordem. Realizando isso, obtivemos o erro do polinômio de grau 7 é de aproximadamente E1 = 609892416, enquanto que o do de grau 3 é de E2 = 1.833. Portanto, o de menor grau apresenta um erro muito inferior ao de maior grau, isso porque este último tem que englobar mais pontos, e estes pontos estão cada vez mais afastados do valor de interesse, que é a marca de 42 dias. Por isso, neste caso é melhor optar por uma opção de menor ordem.

Portanto, o método escolhido pela forma de Newton se deve pela facilidade de calcular o erro da aproximação, mas também devido ao fato de que basta calcular a tabela de ordem uma vez. Repare que a linha que dá origem ao polinômio de grau 3 já está presente na tabela para o de grau 7, e corresponde a quinta linha. Sendo assim, calculando a tabela uma vez já podemos ver a dimensão do erro, e com base no número de pontos que achamos necessários utilizar, podemos determinar o grau do nosso polinômio interpolador.

Portanto, o melhor polinômio interpolador é dado por:

$$p(x) = 2294 + 75.2857142857(x - 28) - 2.5748299320(x - 28)(x - 35) + 0.0026724976(x - 28)(x - 35)(k - 49)$$

Cujo erro em x dado por:

$$E(x) = 0.0026724976*(k - 28)*(k - 35)*(k - 49)$$