Instituto Superior Técnico

Matemática Computacional

Projeto de Matemática Computacional

1º Semestre 2018/2019

MEBiom

Ana Matoso, nº 89787

Madalena Cerqueira, nº 89819

Maria Fernandes, nº 89822

Rita Almeida, nº 90180

Índice

1. Grupo I

1.1.

Ver ficheiro ***[metsec2.m].*** A função *metsec2* recebe uma função, dois números reais a e b um número positivo e “e” um número natural n.

Devolve uma aproximação do zero da função dada segundo o método da secante com iteradas iniciais a e b com critérios de paragem o erro “e” ou o número de iteradas máximo, n.

1.2.

Para calcular o primeiro instante em que a carga elétrica é reduzida para metade do seu valor inicial, recorreu-se ao método da secante implementado na alínea anterior. Foram calculadas iteradas até obter uma aproximação com erro absoluto inferior a 10^{-8}. Criou-se uma função $f(t)$ a partir da funçao $q(t)$ que tem como expressão:

$$f(t) = q(t) - \frac{q(0)}{2}$$

Como q é C^{∞}, f também o é e, portanto, também é C^2.

Para se usar o método da secante verificou-se primeiro se o método era convergente. As condições de convergência são, para um certo intervalo $[a, b]$:

1. $f(a)f(b) < 0$;
2. $f'(t) \neq 0, \forall\, t \in [a, b]$;
3. $f''(t) \leq 0 \ \vee\ f''(t) \geq 0, \forall\, t\ \in [a, b]$;
4. $\left|\frac{f(a)}{f\prime(a)}\right| < b - a \ \wedge\ \left|\frac{f(b)}{f\prime(b)}\right| < b - a$.

De facto, com as iteradas dadas no enunciado, a função não verifica as condições todas, no entanto, assumindo que o método é convergente calculou-se três iteradas. Ora, se considerarmos $[a, b] = [x_3, x_2]$, sendo x_2 a segunda iterada calculada e x_3 a terceira ($x_3 = 0.016667744020192$ e $x_2 = 0.025557930265620$), a função, nesse intervalo já verifica todas as condições de convergência:

1. ver ***[relatoriomc.m]***, secção 2 - justificação de convergência(variável cond1).

2. ver gráfico 1 abaixo criado com o código de ***[relatoriomc.m]***, secção 2 - justificação de convergência

3. ver gráfico 1 abaixo criado com o código de ***[relatoriomc.m]***, secção 2 - justificação de convergência

4. ver ***[relatoriomc.m]***, secção 2 - justificação de convergência.(variável cond4).

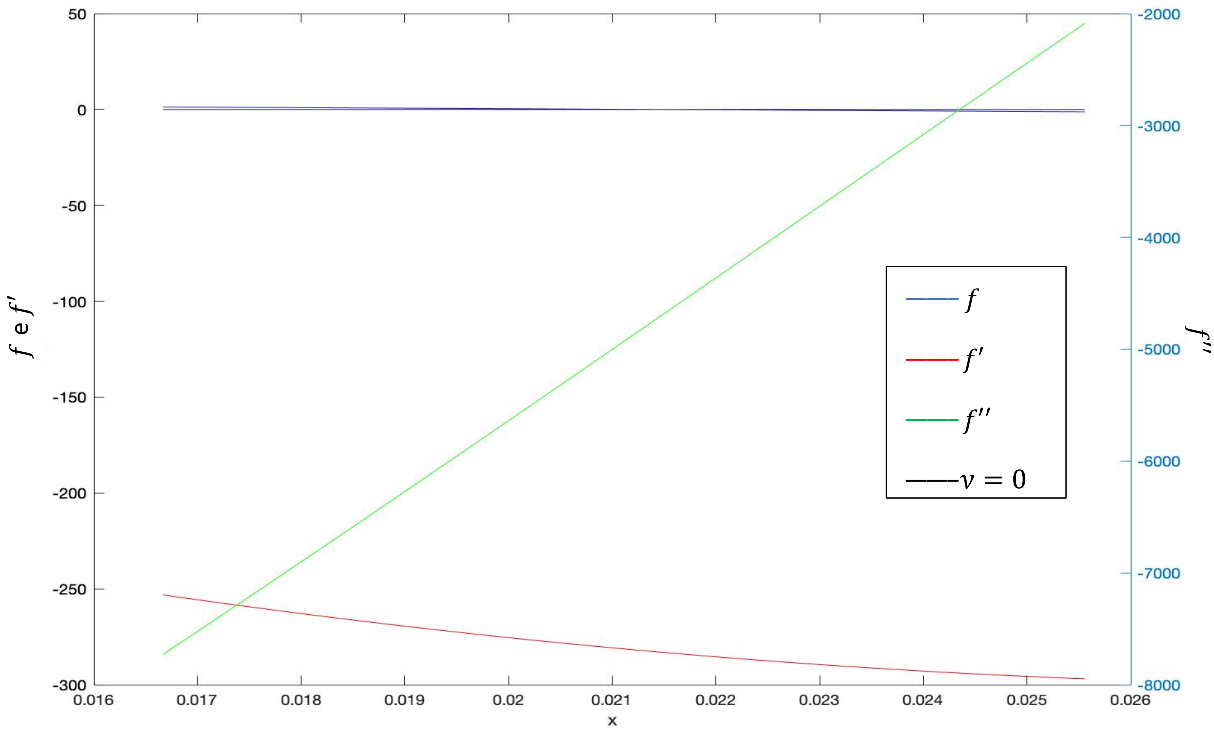

***Gráfico 1** - Função f e suas derivadas*

Assim conclui-se que pelo menos a partir destas duas iteradas o método é convergente.

Para o cálculo das iteradas utilizou-se o código do ficheiro ***[metsec2.m].*** Com L, R, C, t_0 e t_{-1} constantes dadas no enunciado, e e o valor do erro absoluto que a aproximação tem de satisfazer.

$q(t)$ é a função que exprime a carga elétrica do circuito em função do tempo e f é a função que permite calcular o instante em que a carga elétrica é reduzida a metade do seu valor inicial.

Escolheu-se 100 como o valor máximo de iteradas de forma a ter a certeza que a condição de paragem fosse exclusivamente o erro absoluto e não o número máximo de iteradas. De facto, como a tabela só tem 8 linhas em vez de 100 (o que aconteceria se o critério de paragem fosse o número máximo de iteradas), o critério de paragem foi unicamente o erro absoluto. O instante pretendido é aquele em que a função f se anula.

Assim basta introduzir f, t_0, t_{-1}, e e 100 como *input* no código do ficheiro ***[metsec.m]***, secção 1 - Grupo 1, para ter como *output* os dados da tabela seguinte.

Tabela 1 *- Iteradas e respetivo erro absoluto*

Número de iteradas	**Aproximação do zero de f**	**Erro absoluto**
1	0.079530012367804	0.079530012367804
2	0.025557930265620	0.053972082102184
3	0.016667744020192	0.008890186245428
4	0.021279983313024	0.004612239292832
5	0.021459510601779	0.00017952728875528893
6	0.021450739342393	$8.771259386249758 \times 10^{-6}$
7	0.021450751909140	$1.256674663635549 \times 10^{-8}$
8	0.021450751910056	$9.165099235097784 \times 10^{-13}$

Assim uma aproximação com erro inferior a 10^{-8} é $\bar{t} \approx 0.021450751910056$.

2. Grupo II

2.1.

Pretende-se demonstrar que: $B^TWBc = B^TWy$.

Partindo da seguinte equação:

$$Q(c) = \sum_{k=1}^{n} w_k(y_k - g(x_k))^2$$

$$Q(c) = \sum_{k=1}^{n} w_k(y_k - \sum_{i=1}^{m} c_i\varphi_i(x_k))^2$$

$$Q(c) = \sum_{k=1}^{n} w_k(y_k^2 - 2y_k \sum_{i=1}^{m} c_i\varphi_i(x_k) + (\sum_{i=1}^{m} c_i\varphi_i(x_k))^2)$$

$$Q(c) = \sum_{k=1}^{n} w_k(y_k^2 - 2y_k \sum_{i=1}^{m} c_i\varphi_i(x_k) + (\sum_{i=1}^{m} c_i\varphi_i(x_k))^2)$$

$$Q(c) = \sum_{k=1}^{n} w_k y_k^2 - 2\underbrace{\sum_{k=1}^{n} y_k w_k \sum_{i=1}^{m} c_i\varphi_i(x_k)}_{A} + \underbrace{\sum_{k=1}^{n} w_k \left(\sum_{i=1}^{m} c_i\varphi_i(x_k)\right)^2}_{B}$$

$$A = \sum_{k=1}^{n} y_k w_k \sum_{i=1}^{m} c_i\varphi_i(x_k) = \sum_{k=1}^{n}\sum_{i=1}^{m} y_k w_k c_i\varphi_i(x_k) = \sum_{i=1}^{m} c_i \sum_{k=1}^{n} \varphi_i(x_k) y_k w_k$$

$$B = \sum_{k=1}^{n} w_k \left(\sum_{i=1}^{m} c_i\varphi_i(x_k)\right)^2 = \sum_{k=1}^{n} w_k \left(\sum_{i=1}^{m} c_i\varphi_i(x_k)\right)\left(\sum_{i=1}^{m} c_i\varphi_i(x_k)\right) =$$

$$= \sum_{k=1}^{n} w_k \left(\sum_{j,i=1}^{m} c_i c_j \varphi_i(x_k)\varphi_j(x_k)\right) = \sum_{j,i=1}^{m} c_i c_j \left(\sum_{k=1}^{n} \varphi_i(x_k)\varphi_j(x_k)\, w_k\right)$$

Criamos, portanto, a seguinte notação:

$$(y_k, y_k, w_k) = \sum_{k=1}^{n} w_k y_k^2$$

$$(y_k, \varphi_i, w_k) = \sum_{k=1}^{n} \varphi_i(x_k) y_k w_k$$

$$(\varphi_j, \varphi_i, w_k) = \sum_{k=1}^{n} \varphi_i(x_k) \varphi_j(x_k) w_k$$

E assim conseguimos escrever Q da seguinte forma:

$$Q(c_1, c_2, \dots, c_n) = (y, y) - 2\sum_{i=1}^{m} c_i (y_k, \varphi_i, w_k) + \sum_{i,j=1}^{m} c_i c_j (\varphi_i, \varphi_j, w_k)$$

Derivando a função Q componente a componente e igualando a zero, de modo a achar o mínimo, obtém-se o seguinte sistema linear:

$$\begin{cases} -2(y_1, \varphi_1, w_1) + 2\sum_{j=1}^{m} c_j (\varphi_1, \varphi_j, w_1) = 0 \\ -2(y_2, \varphi_2, w_2) + 2\sum_{j=1}^{m} c_j (\varphi_2, \varphi_j, w_2) = 0 \\ \dots \\ -2(y_m, \varphi_m, w_n) + 2\sum_{j=1}^{m} c_j (\varphi_m, \varphi_j, w_n) = 0 \end{cases}$$

Assim, o sistema anterior corresponde a um produto interno com pesos. Convertendo o sistema linear para um sistema normal, obtém-se:

$$\begin{bmatrix} (\varphi_1, \varphi_1, w_1) & \cdots & (\varphi_1, \varphi_m, w_1) \\ \vdots & \ddots & \vdots \\ (\varphi_m, \varphi_1, w_n) & \cdots & (\varphi_m, \varphi_m, w_n) \end{bmatrix} \times \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} (y_1, \varphi_1, w_1) \\ \vdots \\ (y_n, \varphi_m, w_n) \end{bmatrix}$$

O sistema normal acima corresponde à equação:

$$B^T W B c = B^T W y$$

Sendo:

$$B = \begin{bmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \dots & \varphi_m(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_m(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_n) & \varphi_2(x_n) & \dots & \varphi_m(x_m) \end{bmatrix}$$

$$B^T = \begin{bmatrix} \varphi_1(x_1) & \varphi_1(x_2) & \dots & \varphi_1(x_m) \\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_m) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_m(x_1) & \varphi_m(x_2) & \dots & \varphi_m(x_m) \end{bmatrix}$$

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix}$$

2.2.

Ver ficheiro ***[metminquadpond3.m].*** A função *metminquadpond3* recebe uma lista de funções de base, uma lista com as abcissas dos pontos a ajustar, uma lista com as ordenadas dos pontos a ajustar, e uma lista de pesos.

Devolve a aproximação segundo o método dos mínimos quadrados ponderados dos pontos a ajustar com as funções de base dadas e com os pesos dados.

2.3.

Para se conseguir construir a estrada pretendida, com base na tabela dada, utilizou-se o código do ficheiro **[relatoriomc.m]**, secção 3- Grupo 2, Alínea 1. Deu-se como *input* as listas que correspondem às coordenadas de cada cidade, a lista de habitantes de cada cidade, que vai corresponder ao peso que cada cidade vai ter, e as seguintes funções de base:

$$\phi_1(x) = 1 \qquad \phi_2(x) = x \qquad \phi_3(x) = x^2 \qquad \phi_4(x) = x^3$$

A expressão obtida através da aplicação do método foi:

$$y = 0.99097 + 0.30471x + 0.25749x^2 - 0.012799x^3$$

O traçado da estrada obtido com base no critério dos mínimos quadrados ponderados é o seguinte:

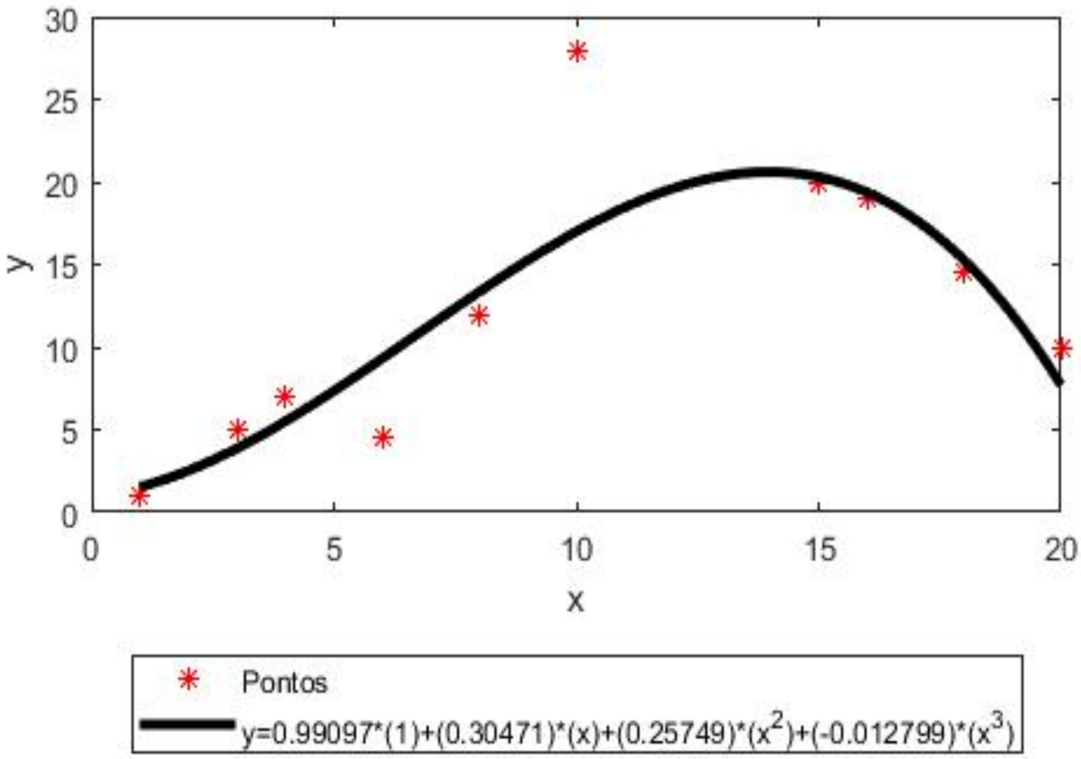

Gráfico 2 *- Curva cúbica obtida através da aplicação do método dos mínimos quadrados ponderados aos pontos em estudo*

3. Grupo III

3.1.

Ver ficheiro ***[Simpson.m]***. A função *Simpson* recebe uma função f, dois números reais a e b, e um número natural n.

Devolve uma aproximação de $\int_a^b f(x)\,dx$ se dividirmos o intervalo $[a,b]$ em n subintervalos.

3.2.

Sendo $f(x) = x^{20}e^{x-1}$, como $f \in C^{\infty}([0,1])$, podemos utilizar o método de Simpson para aproximar o integral $\int_0^1 x^{20}e^{x-1}\,dx$.

A fórmula do erro para o método de Simpson é a seguinte:

$$|E_N^S(f)| \leq \frac{(b-a)^5}{180\ \times\ N^4} \times max_{x\in[a,b]}|f^{(4)}(x)|$$

Logo para o cálculo do n que vai dar pelo menos 12 algarismos significativos faz-se:

$$|E_N^S(f)| \leq 0.5\ \times\ 10^{-12}$$

$$\frac{1}{180\ \times\ N^4} \times max_{x\in[0,1]}f^{(4)}(x)\ \leq\ 0.5\ \times\ 10^{-12}$$

$$f(x) =\ x^{20}e^{x-1}$$

$$f'(x) =\ 20x^{19}e^{x-1} + x^{20}e^{x-1}$$

$$f''(x) =\ 380x^{18}e^{x-1} + 20x^{19}e^{x-1} + 20x^{19}e^{x-1} +\ x^{20}e^{x-1}$$

$$f''(x) =\ 380x^{18}e^{x-1} + 40x^{19}e^{x-1} +\ x^{20}e^{x-1}$$

$$f'''(x) =\ 6\,840x^{17}e^{x-1} + 380x^{18}e^{x-1} + 760x^{18}e^{x-1} +\ 40x^{19}e^{x-1} +\ 20x^{19}e^{x-1} + x^{20}e^{x-1}$$

$$f'''(x) =\ 6\,840x^{17}e^{x-1} + 1\,140x^{18}e^{x-1} +\ 60x^{19}e^{x-1} + x^{20}e^{x-1}$$

$$f^{(4)}(x) =\ 116\,280x^{16}e^{x-1} + 6\,840x^{17}e^{x-1} +\ 20\,520x^{17}e^{x-1} + 1\,140x^{18}e^{x-1} +\ 1\,140x^{18}e^{x-1} +\ 60x^{19}e^{x-1} + x^{20}e^{x-1} +\ 20x^{19}e^{x-1}$$

$$f^{(4)}(x) = 116280x^{16}e^{x-1} + 27\,360x^{17}e^{x-1} + 2\,280x^{18}e^{x-1} + 80x^{19}e^{x-1} + x^{20}e^{x-1}$$

Assim, como $f^{(4)}(x)$ é uma função crescente positiva:

$$max_{x\in[a,b]}\left|f^{(4)}(x)\right| = f^{(4)}(1) = 146\,001$$

$$\frac{1}{180 \times N^4} \times 146\,001 \leq 0.5 \times 10^{-12}$$

$$N \geq 6\,346.412991 \wedge N\, par$$

$$N = 6\,348$$

Considerou-se, portanto, como referência o integral calculado, com n=6348, através do método de Simpson que tem pelo menos 12 algarismos significativos, (variável I no código abaixo mencionado).

Para construir a tabela 2 e a tabela 3 abaixo utilizou-se o código do ficheiro ***[relatoriomc.m],*** secção 4 - Grupo 3, alínea 2 (tabela 2-variável resposta32, tabela 3- variável respostalin).

Tabela 2 *- Relação entre o valor do integral calculado com o método de Simpson com n variável e o seu erro absoluto*

n	S_n	$\|E_n\|$
10	0.048710762505035	0.003165878429190
20	0.045799729061692	$2.548449858462282 \times 10^{-4}$
40	0.045561934788246	$1.705071240053896 \times 10^{-5}$
80	0.045545968389068	$1.084313222557420 \times 10^{-6}$
160	0.045544952141210	$6.806536408110464 \times 10^{-8}$
320	0.045544888334545	$4.258699372317043 \times 10^{-9}$
640	0.045544884342061	$2.662154871546640 \times 10^{-10}$
1280	0.045544884092459	$1.661381449080679 \times 10^{-11}$
2560	0.045544884076858	$1.012592787397182 \times 10^{-12}$

Para calcular a ordem de convergência, sabe-se que se calcula com a equação abaixo em que p é a ordem de convergência:

$$E_h = C(h_k)^p \iff \ln(|E_h|) = \ln C + pln(h_k)$$

Procedeu-se, portanto, à linearização da equação calculando o logaritmo dos dados de forma a poder calcular o valor de p:

Tabela 3 *- Logaritmo de h_k e E_h respetivamente.*

$\boldsymbol{ln(h_k)}$	$\boldsymbol{ln(\|E_h\|)}$
-2.302585092994046	-5.755324717126796
-2.995732273553991	-8.274855096293384
-3.688879454113936	-10.979318572084821
-4.382026634673881	-13.734563745986305
-5.075173815233827	-16.502797356975424
-5.768320995793772	-19.274302034944892
-6.461468176353717	-22.046715033017232
-7.154615356913663	-24.820786568409640
-7.847762537473608	-27.618506958255090

Com estes dados foi possível, a partir da função do código do ficheiro ***[metminquad3]***, e dando como *input* as colunas da tabela acima, uma lista w que contem os pesos todos iguais a 1, e as funções de base:

$$\phi_1(x) = 1 \qquad \phi_2(x) = x$$

Assim determinou-se a seguinte equação em que p é o declive da reta:

$$y = 3.5503 + 3.9618x$$

Ora, isto vem verificar a ordem de precisão do método de Simpson visto que o valor estimado é 3.9618 e o real é 4.

3.3.

Para calcular os valores para I_k presentes na tabela 4 usou-se o código presente no ficheiro ***[relatoriomc.m],*** secção 5-Grupo 3, alínea 3 (variável Ik).

Tabela 4 *- Relação entre o valor de k e Ik*

k	$\mathbf{I_k}$
2	0.264241117657115
3	0.207276647028654
4	0.170893411885384
17	0.057191870597308
18	-0.029453670751536
19	1.559619744279189
20	-30.192394885583780
21	$6.350402925972594 \times 10^{2}$
25	$1.927850088325280 \times 10^{8}$
30	$-3.296762455608387 \times 10^{15}$
50	$-3.780092725853228 \times 10^{47}$
100	$-1.159928542966359 \times 10^{141}$

Utilizando os dois métodos para calcular o integral, observamos que os resultados obtidos são consideravelmente diferentes. No método de Simpson (S_n) obtivemos resultados semelhantes e próximos de 0,045. No entanto, com a fórmula de recorrência, obtemos um valor próximo de -30.

Esta diferença pode ser explicada pelo facto de os valores dos integrais calculados da forma recursiva dependerem do anterior e, portanto, como cada iterada tem um erro associado a ela, isso vai fazer com que cada vez haja um erro acumulado maior (amplificação do erro) e daí a discordância deste método de recorrência com o método de Simpson (mais preciso, visto ser de ordem 4).

3.4.

Pretende-se comparar os valores de números primos até um certo número x calculados com a função $Li(x) = \int_2^x \frac{dt}{\ln t}$ e com a função *primes* do Matlab. Como a função $Li(x) \in C^\infty([2,x])$, $\forall\, x \geq 2$, usou-se o código seguinte: ***[relatoriomc.m],*** <u>secção 6-Grupo 3, alínea 4</u> para construir a tabela 5 (variável comp).

Tabela 5 *- Contagem de números primos através da função Li(x) e da função primes do Matlab e erro relativo respetivo*

x	Função $Li(x)$	Função *primes* do Matlab	Erro relativo
10	5	4	0.25
100	29	25	0.16
1000	176	168	0.047619047619
10000	1245	1229	0.013018714402
100000000	5797913	5761455	0.000877903237984

Comparando os dois resultados, pode-se ver que à medida que o valor máximo até ao qual se conta os números primos aumenta, o erro relativo diminui. Ora, isto vem confirmar a hipótese do enunciado que afirma que quando x (sendo x o valor até ao qual se quer contar os números primos) tende para infinito, $\frac{Li(x)}{\pi(x)}$ tende para 1.

No entanto, verificou-se que a função *primes* do Matlab, para números x muito grandes (maiores que 10^9) é inutilizável, uma vez que seria preciso gastar imensa memória, e por isso dá erro.

Não obstante, a função $Li(x)$ consegue calcular uma estimativa, independentemente do valor de x e, como o erro decresce quanto maior for x, cada vez essa aproximação é mais precisa e, portanto, a função $Li(x)$ torna-se cada vez mais uma boa alternativa para o cálculo do número de números primos.

4. Grupo IV

4.1.

O programa encontra-se no anexo ***[pontomediovet.m]***. a função *pontomediovet* recebe um função $f(t, y(t))$, uma iterada inicial y_a, dois números reais a e b e um número natural n. Depois de corrida, devolve uma sucessão de iteradas calculadas a partir do método do ponto médio da solução do seguinte sistema:

$$\begin{cases} y'(t) = f\big(t, y(t)\big), t \in [a, b] \\ y(a) = y_a \end{cases}$$

4.2.

Para esboçar os gráficos correspondentes às quatro situações usou-se o código seguinte: ***[relatoriomc.m],*** <u>secção 7 - Grupo 4, alíneas a), b), c) e d)</u>.

Para cada gráfico deu-se como *input* os valores de $a_i, b_i, c_i, d_i \ (i = 1, 2, 3, 4)$, o intervalo $[a, b] = [0{,}365]$, o número de subintervalos, 365, i.e., cada iterada vai corresponder ao final de cada dia (iterada 0 corresponde ao início do primeiro dia), o vetor $y_0 = \begin{bmatrix} y_{1,0} \\ y_{2,0} \end{bmatrix}$, e a função $f_i(t, y) = \begin{bmatrix} a_i y_1 + b_i y_2 (1 - |y_2|) \\ d_i y_2 + c_i y_1 (1 - |y_1|) \end{bmatrix}$.

O *output* são os quatro gráficos seguintes, onde a linha vermelha corresponde à Julieta e a azul ao Romeu:

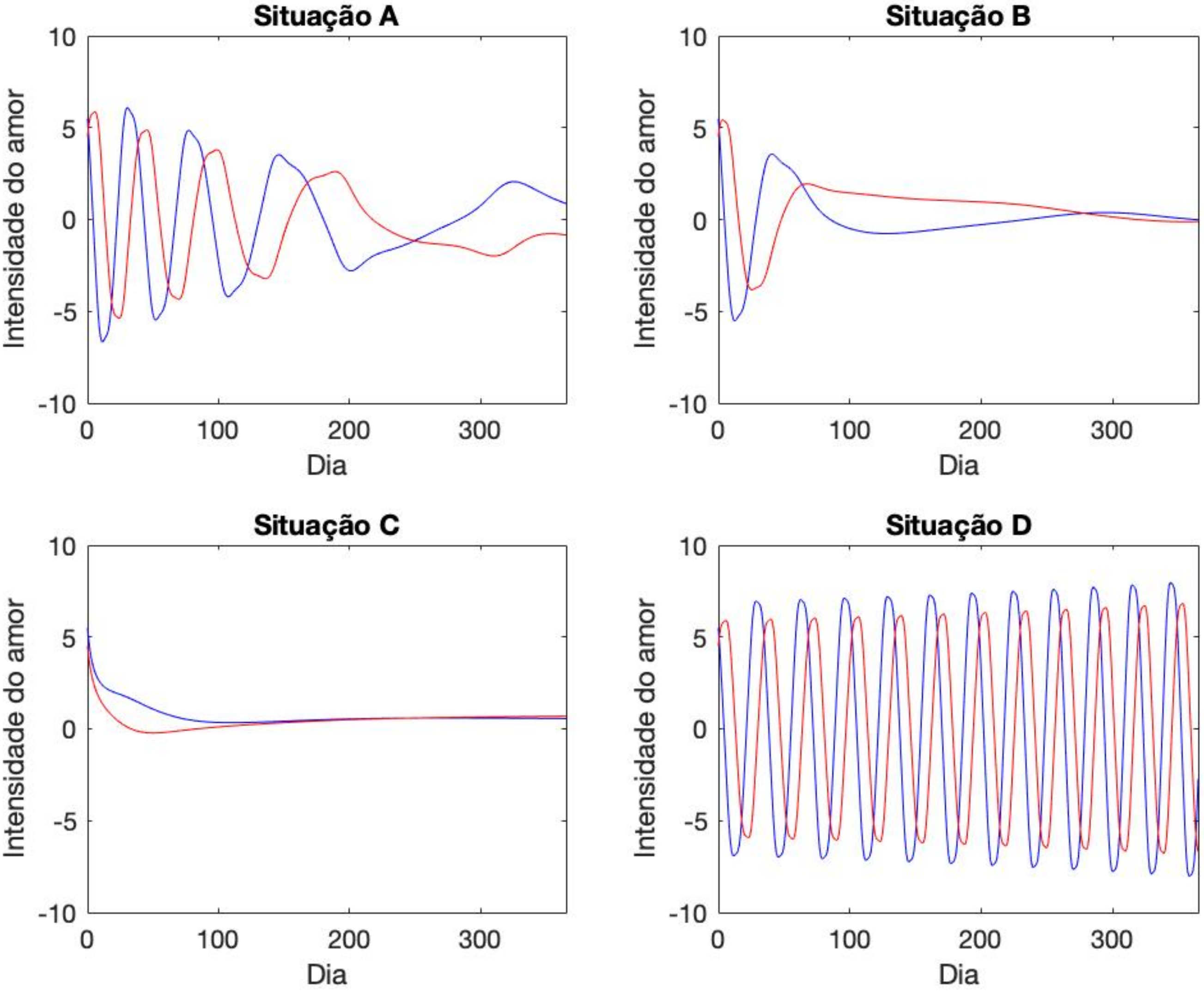

Gráfico 3 *- Gráficos das 4 situações pretendidas*