UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO - INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
MS211A - CÁLCULO NUMÉRICO
PROF. JOÃO FREDERICO DA COSTA AZEVEDO MEYER

PROVA 2

BRYAN WOLFF RA: 214095

CAMPINAS
JULHO DE 2021

Observação: *Todos os gráficos gerados e as questões discutidas foram feitos juntamente com a aluna Wesna Simone Bulla de Araujo RA: 225843. Além disso, os scripts utilizados estão anexados ao final da prova.*

Questão 1 - *O vetor abaixo contém os últimos 66 valores para as mortes diárias em todo o mundo nos últimos 66 dias (retirados de https://www.worldometers.info/coronavirus/, do gráfico "Deaths per Day"). Faça uma previsão do número de mortes diárias no 80º dia. Justifique sua escolha da função a ser ajustada. Critique avaliativamente seu resultado: porque você tende a acreditar nele?*

y=[15812,14673,13855,10766,11519,14705,15233,14594,14453,13513,10927,11543,14505, 14587,14435,14047,13289,10606,11515,14846,13705,14000,13595,11635,10484,9519,12844 ,12740,11950,12013,10946,8544,8934,11071,11301,11101,10439,9366,7765,7948,10297,102 23,10197,9438,8659,6888,6798,9243,9508,8753,9057,8707,6780,6439,8931,9094,8655,8782 ,7635,6345,6307,7892,8784,8318,8545,7214];

Para realizar a previsão do número de mortes diárias, foi utilizado o *script* criado pelo doscente "*aulaQM0TN.m*" *n*o qual utiliza o método dos quadrados mínimos lineares. Vale destacar que a lógica programacional que foi adaptada para este exercício está disponível no Anexo A. A escolha da função a ser ajustada trata-se de uma função exponencial, pois ao plotar os dados de dispersão do vetor y, é notável que os números, decaem a uma aproximação exponencial. Dessa forma, utilizando a mesma função exponencial criada pelo docente no código, foi possível obter os seguintes resultados demonstrados na Figura 1.

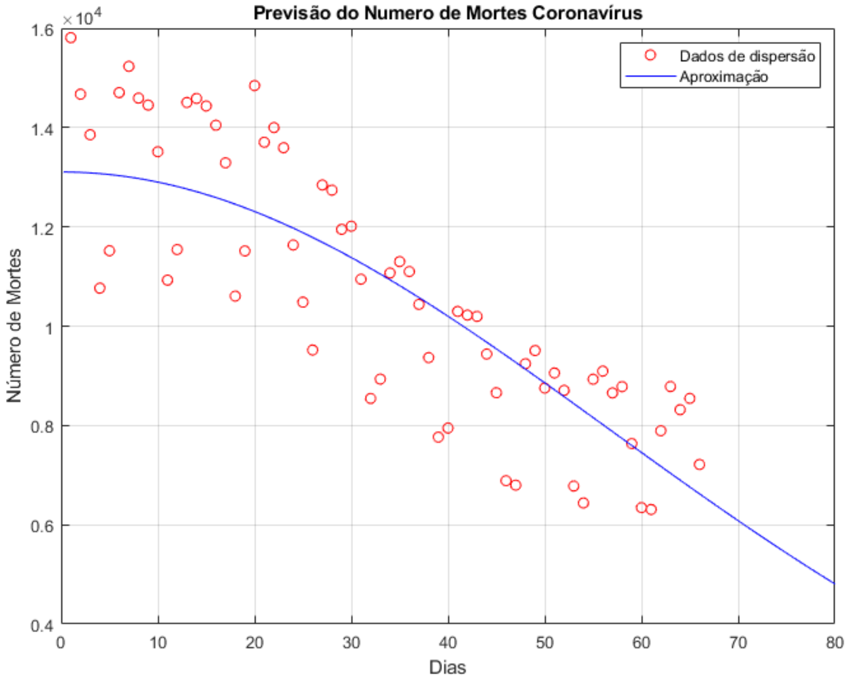

Figura 1: Previsão do número de mortes causadas pelo Coronavírus.

A partir da imagem é notável que a função encontrada aproxima bem os resultados analíticos. Pela aproximação calculada pelo *script*, obtemos que o número de mortes previstas para o 80° dia é 4807 mortes.

Questão 2 - *Para se obter o valor numérico da integral:* $\int_0^{\frac{\pi}{2}} (|\text{sen}(x) * \cos(x)|^{3/2})$ *. Qual seria o método mais adequado para obter tal resultado com uma ótima precisão? Observe, por favor, que o termo "ótima precisão" é bastante subjetivo. É isso mesmo. Justifique sua escolha de fórmula, use algum software para calcular essa integral e avalie o resultado obtido. Qual método seria melhor se não fosse necessária uma sensível precisão e sim rapidez de cálculo – e por quê?*

O método utilizado para obter o resultado dessa integral com uma boa aproximação é o algoritmo de Romberg, no qual permite que se calcule com grande precisão a integral com um esforço computacional relativamente pequeno. O algoritmo utilizado para o cálculo desta integral foi implementado pelo Felipe Longo e está disponível na página da disciplina como "*romberg.m*" . Além disso, o código adaptado para a questão também está disponível no Anexo B.

As vantagens de se utilizar a integração de Romberg se dá devido ao fato de que é necessário um menor esforço computacional quando em comparação com alguns outros métodos. Além disso, o algoritmo é de fácil implementação, e o usuário pode escolher a precisão com base no parâmetro de passo. Para o tamanho de passo *n*, a ordem de convergência é de pelo menos n^6 enquanto usando a regra de Simpson é n^4.

Ao executar o *script* com a implementação do método de integração de Romberg, obtemos que o valor da integral no intervalo [a,b] é dado por: 0.30901246. Vale destacar que foi dividido o intervalo [a,b] em 2^n para *n=10*, o que indica uma ótima precisão para o exercício proposto.

Caso não fosse necessária uma precisão sensível e ótima, priorizando apenas a rapidez de cálculo, o método indicado seria o método de Simpson, pois possui um tempo médio de processamento bem menor que o método de Romberg, pois este é outra fórmula de Newton-Cotes para aproximar integrais em um intervalo com nós igualmente espaçados. Assim, pode-se concluir que o método de Simpson requer um tempo de processamento menor que o método de Romberg em integral aproximada, sendo melhor para este caso.

Questão 3 - *Considere a edo dada por* $\frac{dp}{dt}(t) = 0{,}025.\,p(t).\left[1 - \frac{p(t)}{2000+e^{\frac{-t}{12}}}\right]$ *sabendo que p(0)=121. Deseja-se obter a solução no intervalo [0, 75]. Desenvolva esta aproximação numérica, justificando suas escolhas. Se der, forneça um gráfico da aproximação obtida.*

Para obter a solução da edo no intervalo proposto, utilizou-se do *script "ODE45.m"* disponibilizado pelo Felipe Longo na página da disciplina. Além disso, o código adaptado para a questão também está disponível no Anexo C. Ao executar o *script,* obtemos uma aproximação numérica a partir do solver *ODE45*, que utiliza combinação dos métodos de Runge-Kutta de quarta e quinta ordem. O resultado da aproximação obtida está na Figura 2.

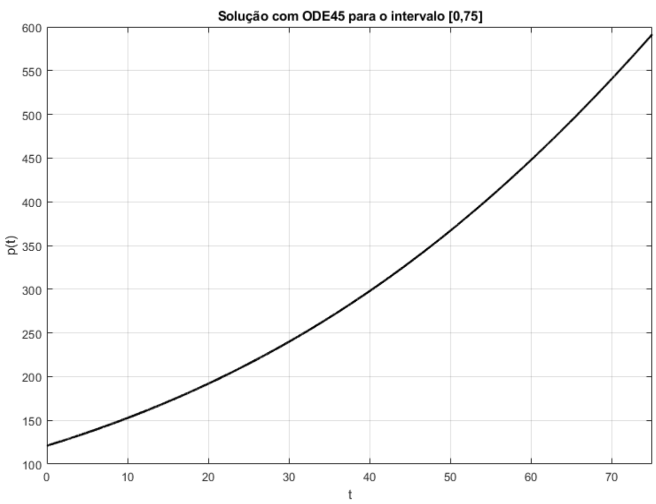

Figura 2: Solução da EDO com ODE45 para o intervalo [0,75].

Questão 4 - *Dos métodos apresentados para aproximar numericamente o valor de uma integral de uma função de que não se conhece a primitiva, há algum método que você acredita que nunca vai usar? Justifique cuidadosamente suas opiniões.*

Dos métodos apresentados para aproximar numericamente o valor de uma integral de uma função, um método que eu evitaria utilizar é o Método dos Trapézios, pelo fato da estimativa de erro da solução numérica ser geralmente maior do que os outros métodos estudados. O erro é maior pois quando a função apresenta um gradiente decrescente, a regra do trapézio sempre dá uma subestimativa da solução, enquanto para um gradiente crescente ela dá uma superestimativa..

Questão 5 - *Na resolução de um sistema do tipo m.x = b, qual ou quais métodos você tentaria – e em que ordem? Justifique cuidadosamente suas escolhas. Como é que se poderia avaliar se o resultado está satisfatório?*

Dentre os métodos estudados eu escolheria em ordem o Método iterativo de Newton-Raphson Modificado, Método iterativo de Gauss-Seidel e o Método Direto de Eliminação de Gauss com pivoteamento parcial.

O método iterativo de Newton é considerado por muitos autores o melhor método para encontrar sucessivas melhores aproximações da solução, além do método funcionar para grande parte dos casos e ainda também para sistemas lineares e não lineares. Além de eficiente, apresenta um número reduzido de iterações e com uma boa precisão para encontrar a solução aproximada do sistema. Além disso, o Método de Newton Modificado tem a vantagem de calcular uma única vez a matriz Jacobiana e pode convergir para uma solução quando o ponto inicial está longe desta. E o número de iterações são independentes do tamanho do sistema.

O método de Gauss-Seidel, é um ótimo método para a resolução de sistemas lineares, pois os cálculos por iterações são simples e, portanto, a sua implementação é mais fácil. Dessa forma, o tempo gasto por iteração é menor do que o Método de Newton, e por isso o armazenamento necessário na memória do computador é relativamente menor, podendo ser aplicados em sistemas menores.

Já o Método de Eliminação de Gauss é o algoritmo de solução mais fundamental, que consiste em aplicar operações elementares sucessivas em um sistema linear (escalonamento), transformando num sistema mais fácil de resolvê-lo. A técnica de pivô parcial é usada para evitar erros de arredondamento. Além disso, pode ser usado onde o número de equações e o número de incógnitas não são iguais.

Questão 6 - *Considere uma equação diferencial de segunda ordem (na verdade, um PVC) dada por y''(t) + 2,12*y'(t) + 0.078*y(t) = f(t) para t ∈ [0, 5] e sendo y(0)=2 e y(5)=2 e sendo f(t) dada por:*

$$\begin{cases} 0, \text{se } x \in [0,1), \\ 2, \text{se } x \in [1, 3{,}5] \\ 0, \text{se } x \in (3{,}5,\ 5]. \end{cases}$$

Para a resolução da equação diferencial de segunda ordem, utilizou-se como base a implementação de um problema de valor de contorno simples disponibilizado pelo docente na página da disciplina pelo nome *"PVCSimples.m". O script* simula um varal e com 2 pesos diferentes colocados em pontos diferentes desse varal. O script foi adaptado para os cálculos matemáticos propostos pelo docente Marcos Valle na [Aula 23](), e está disponível no [Anexo D]().

Ao executar o *script*, podemos obter a solução do sistema de segunda ordem *y''(t) + 2,12*y'(t) + 0.078*y(t) = f(t)* para t ∈ [0, 5] e sendo *y(0)=2* e *y(5)=2.* Esta solução está disponível na Figura 3.

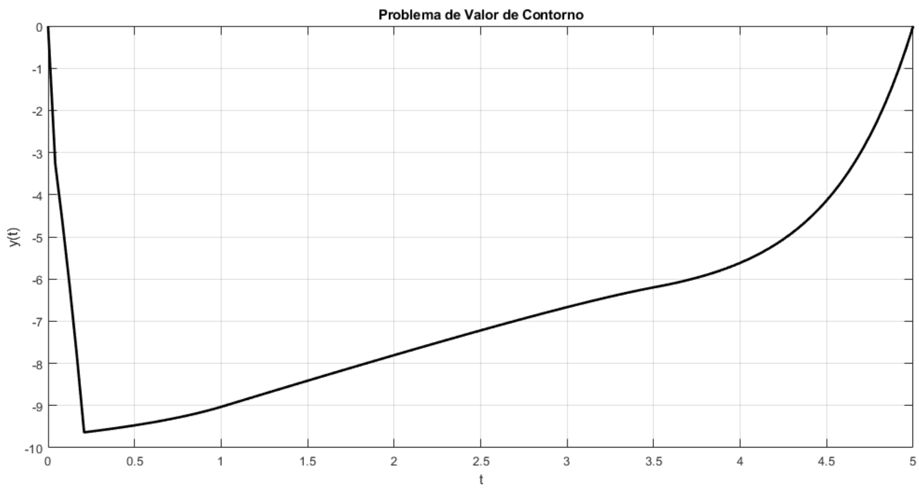

Figura 3: Solução do PVC para *2,12*y'(t) + 0.078*y(t) = f(t) e* t ∈ [0, 5].

Questão 7 - *Para calcular y(0.63), use a tabela com valores de x e de y dada abaixo, fornecendo uma estimativa de erro, justificando suas decisões.*

x	0.06813	0.17032	0.32361	0.55355	0.89845	1.41581	2.19184
y	121.9718	123.4432	125.6814	129.1098	134.4159	142.7555	156.1629

Para calcular o ponto *y(0.63),* foi utilizado como base o *script* para Interpolação de Newton disponibilizado pelo Felipe Longo na página da disciplina pelo nome *"newton_interpolation.m"*. O *script* foi adaptado para o exercício utilizando-se dos cálculos propostos pelo docente Marcos Valle na Aula 16 para a estimativa do erro. Além disso, o *script* adaptado para a questão proposta está disponível no Anexo E.

Ao executar o programa, obtemos que o ponto de interpolação é dado por $y(0.63) = 130.26883996$e que a estimativa para o erro de interpolação é de: 0.00045644. Portanto, dado o ponto em questão e a estimativa de erro da solução obtida indica uma boa aproximação da solução encontrada.

Questão 8 - *Comente o que lhe pareceu útil num futuro exercício profissional daquilo que foi visto em MS211 e os motivos dessa indicação. Faça o mesmo com o que lhe pareceu menos útil com o porquê dessa opinião.*

Em geral, tudo o que aprendemos até agora pode ser utilizado posteriormente em exercícios profissionais. Os enfatizados Métodos Numéricos, seja para encontrar zeros de funções, soluções de EDO, sistemas lineares, integrais de funções difíceis, quaisquer que sejam seu objetivo, os materiais estudados dos métodos numéricos nos introduz à um universo cheio de possibilidades no qual é possível formular e resolver problemas matemáticos complexos de diversas áreas da ciências exatas usando operações aritméticas menos complexas a partir de aplicações de algoritmos simples.

Em resumo, eu não acho que eu possa indicar o que eu não achei útil para a vida profissional, devido ao fato de que, mesmo que um método apresenta desvantagens em um certo ponto, existem outros cujo este ponto seja uma vantagem, como por exemplo o método iterativo de Newton para resolver sistemas lineares é considerado por muitos autores o melhor método para encontrar sucessivas melhores aproximações da solução, porém no método de Gauss-Seidel, o tempo gasto por iteração é menor do que o Método de Newton, e por isso o armazenamento necessário na memória do computador é relativamente menor, podendo ser aplicados em sistemas menores. Portanto, pode ser que um método possa não ser útil em uma situação, na qual outros são.

Questão 9 - *Comente algum tópico que, para você, não foi abordado de modo adequado na disciplina de MS211.*

Acredito que a resolução de sistemas baseados em equações diferenciais foi um tópico passado um pouco rápido demais.

ANEXO A: *SCRIPT* DA ATIVIDADE 1

```
% Método dos quadrados mínimos lineares
% exemplo prático usando exponencial
% y = a*exp(b*x²)
% fi1=1   e   fi2=x²
clc
clear all
% Dados da CoViD-19 no Brasil: casos diários.média de 7 dias
% Fonte: view-source:https://www.worldometers.info/coronavirus/country/brazil/
%
% 49 últimos dias
dias=[1:66];
% Casos correspondentes
casos=[15812,14673,13855,10766,11519,14705,15233,14594,14453,13513,10927,11543,14505,14587,14435,14047,13289,10606,11515,14
846,13705,14000,13595,11635,10484,9519,12844,12740,11950,12013,10946,8544,8934,11071,11301,11101,10439,9366,7765,7948,10297,
10223,10197,9438,8659,6888,6798,9243,9508,8753,9057,8707,6780,6439,8931,9094,8655,8782,7635,6345,6307,7892,8784,8318,8545,72
14];
% visualização

N=length(dias); a11=N;
a12=sum(dias.*dias); a21=a12; a22=sum(dias.^4);
Y=log(casos);
m=[a11 a12; a21 a22]; b=[sum(Y);sum((dias.*dias).*Y)];
c1=m\b;
x=0.25*[1:320];
% Transformar c1 em d1 em d1=exp(c1(2)) e d2=c1(2)
d1=exp(c1(1)); d2=c1(2);
for i=1:320
  y1(i)=d1*exp(d2*(x(i)*x(i)));
end

figure
p1 = plot(dias,casos,'ro'); hold on;
p2 = plot(x,y1,'b'); hold off;
h = [p1(1);p2(1)];
legend(h, 'Dados de dispersão','Aproximação');
grid;
xlabel('Dias');
ylabel('Número de Mortes');
title('Previsão do Número de Mortes Coronavírus');
fprintf("O número de mortes previstas para o 80° dia é %i", y1(320));
```

ANEXO B: SCRIPT DA ATIVIDADE 2

```
clc
clear all
format long

% Função para ser integrada
f=@(t) (abs(sin(t)*cos(t))^(3/2));

%intervalo de integração [a,b]
a=0; b=pi/2;

% aumentar n para aumentar a precisão.
% dívide o intervalo [a,b] em 2^n
n=10;
h=b-a;

R=zeros(n,n);
R(1,1)=(f(a)+f(b))*h/2;

for k=2:n
   for i=1:2^(k-2)
      R(k,1)=R(k,1)+f(a+(2*i-1)*h/(2^(k-1)));
   end
   R(k,1)=R(k-1,1)*0.5 + R(k,1)*h/(2^(k-1));
end

for k=2:n
   for j=2:k
      R(k,j)=R(k,j-1)+(R(k,j-1)-R(k-1,j-1))/(4^(j-1) -1);
   end
end

IntRomberg = R(n,n);
fprintf("O valor da integral no intervalo [a,b] é dado por: %.8f", IntRomberg);
```

ANEXO C: SCRIPT DA ATIVIDADE 3

```
clc; close all; clear;

%EDO
f=@(t,y) 0.025*y*(1-(y/(2000+(exp(-t/12)))));

%intervalos e condições iniciais
t0=0; tf=75;
y0=121;

% para construção do gráfico
h=0.01;
t=t0:h:tf;
%

sol=ode45(f,[t0 tf],y0);
y=deval(sol,t);

plot(t,y,'b-','LineWidth',2)
title('Solução com ODE45 para o intervalo [0,75]')
grid;
xlabel('t');
ylabel('p(t)');
xlim([0 75]);
```

ANEXO D: SCRIPT DA ATIVIDADE 6

```
clc; clear all;
%Resolvendo um sistema linear da forma Av+B
% Equação dada por
% y''(x) + a*y(x)' b*y(x) = f(x)
% Domínio
L=5;
% Parâmetros
a=2.12; b=0.078;
% Dados de Discretização
N=120; deltx=L/N; dx=deltx^2;
% Preparação da matriz e do vetor
n=N-1;
%Costrução da Matriz A
for i=1:n
  A(i,i) = 2 + b*dx;
end
for i=1:n-1
  A(i,i+1) = -1 + ((a*deltx)/2); %Diagonal Superior
  A(i+1,i) = -1 - ((a*deltx)/2); %Diagonal Inferior
end
%condições de contorno
yi=2;    yf=2;
%construção da matriz B
B = [];
time = deltx;
pos = 1;
while true
    B(end + 1) = -dx*f(time);
    if pos == n
        break
    end
    time = time + deltx
    pos = pos+1
end
B(1)= B(1) - (yi*(-1 - ((a*deltx)/2)));
B(5)= B(5) - (yf*(-1 + ((a*deltx)/2)));
B = B'
% Resolução do sistema
y = A\B;
%gráfico
x=[0:deltx:L];
varal=-[0 ; y; 0];
plot(x,varal,'r');
grid;
xlabel('t');
ylabel('y(t)');
```

```
title("Problema de Valor de Contorno");

function t = f(x)
    if x >= 0 && x < 1
        t = 0;
    end
    if x >= 1 && x <=3.5
        t = 2;
    end
    if x > 3.5 && x <= 5
        t = 0;
    end
end
```

ANEXO E: SCRIPT DA ATIVIDADE 7

```
clc;
clear all;

%Matrizes de linhas
% x and y are two Row Matrices and p is point of interpolation
x = [0.06813 0.17032 0.32361 0.55355 0.89845 1.41581 2.19184];
y = [121.9718 123.4432 125.6814 129.1098 134.4159 142.7555 156.1629];
p = 0.63;



%chamando a função de interpolação de newton para o ponto 0.63
pointInterpolation = newtonInterpolation(x,y,p);
fprintf("O ponto de inerpolação y(0.63) é dado por: %0.8f\n", pointInterpolation )

%calculando a estimativa de erro
x0=x(4);
x1=x(5);
x2=x(6);
erro = estimativaErro(x,y,p,x0,x1,x2);
fprintf("Estimativa para o erro de interpolação: %.8f\n", erro)

%função da estimativa do erro para ordem2
%x,y matriz de linhas
%p é o ponto de interpolação
%x0,x1,x2 pontos da matriz x próximos de p
function erro = estimativaErro(x,y,p,x0,x1,x2)
    ordem0 = y;
    ordem1 = [];
    ordem2 = [];
    ordem3 = [];

    i = 1;
    %calculando ordem 1
```

```
    while true
        ordem1(end+1)= (abs(ordem0(i)-ordem0(i+1)))/(abs(x(i)-x(i+1)));
        i = i + 1;
        if i == length(y);
            break
        end
    end

    %calculando ordem 2
    i=1;
    while true
        ordem2(end+1) = (abs(ordem1(i)-ordem1(i+1)))/(abs(x(i)-x(i+2)));
        i = i + 1;
        if i == length(ordem1);
            break
        end
    end
    %calculando ordem 3
    i=1;
    while true
        ordem3(end+1) = (abs(ordem2(i)-ordem2(i+1)))/(abs(x(i)-x(i+3)));
        i = i + 1;
        if i == length(ordem2);
            break
        end
    end
    erro = abs(max(ordem3)*(p-x0)*(p-x1)*(p-x2));
end

function fp = newtonInterpolation(x,y,p)
n = length(x);
a(1) = y(1);
for k = 1 : n - 1
   d(k, 1) = (y(k+1) - y(k))/(x(k+1) - x(k));
end
for j = 2 : n - 1
   for k = 1 : n - j
      d(k, j) = (d(k+1, j - 1) - d(k, j - 1))/(x(k+j) - x(k));
   end
end
d;
for j = 2 : n
   a(j) = d(1, j-1);
end
Df(1) = 1;
c(1) = a(1);
for j = 2 : n
   Df(j)=(p - x(j-1)) .* Df(j-1);
   c(j) = a(j) .* Df(j);
end
```

```
fp=sum(c);
end
```