Grupo II

No grupo II a tabela com todos os resultados vai estar no fim, em conjunto com o código que contém a organização e exposição dos resultados na tabela.

O ficheiro que contém f(x), f.m, encontra-se se seguida.

function y = f(x)

y = atan(sin(x));

end

1. O método dos trapézios revela-se ser o seguinte:

, sendo e . No nosso caso, . Este equivale ao número de trapézios que o programa irá calcular, e é o inteiro calculado diretamente do erro, que o método produz. Neste caso, . O nosso K é calculado através da visualização direta do máximo do gráfico .

n = ceil((sqrt((K \* (deltaX^3))/(12\*maxerror))));

X = 0:pi/100:2\*pi;

Y = f(X);

dY = gradient(Y(:))./gradient(X(:));

dYY = gradient(dY(:))./gradient(X(:));

plot(X, dYY);

No programa vai ser calculado para cada presente na tabela. Para isso vai ser usado um ciclo for. Na tabela vai ser também mostrado o e calculado para cada x.

De seguida encontra-se o ficheiro A.m:

function ret = A(xs, maxerror)

format long;

K = 0.9;

a = 0;

results = [];

erros = [];

Ns = [];

for b = double(xs)

deltaX = b - a;

n = ceil((sqrt((K \* (deltaX^3))/(12\*maxerror))));

x = a:deltaX/n:b;

x = x(:);

y = f(x);

y = y(:);

result = (ones(1,n) .\* (deltaX/n)) \* (y(1:end-1, :) + y(2:end, :))/2;

results = [results result];

erroAbsolutoMenorQue = (K \* (deltaX^3))/(12\*(n^2));

erros = [erros erroAbsolutoMenorQue];

Ns = [Ns n];

end

ret = [double(results); erros ;Ns];

end

b)

O método referido no enunciado aparenta ser o seguinte:

. Quanto mais perto estiver de 0, maior a precisão do resultado com o real. Para calcular entre os pontos, podemos usar a função diff. Como os pontos inicial e final não contêm anteriores e posterior, respetivamente, usa-se a formula sem a necessidade de ter tais.

De seguida encontra-se o ficheiro B.m:

function ret = B(xs, gxs)

hs = diff(double(xs));

ret = ones(size(xs));

ret(1) = (gxs(2) - gxs(1) ) / hs(1);

ret(end) = (gxs(end) - gxs(end-1)) / hs(end);

ret(2:end-1) = (gxs(3:end) - gxs(1:end-2) )./(2\*hs(2:end));

end

c)

O teorema fundamental do cálculo descreve que se g(x) estiver definido em , e , então . Sendo e a = 0, .

De seguida encontra-se o ficheiro C.m:

function ret = C(xs)

fxs = f(xs);

dgdxExato = fxs;

syms x

dfdx = diff(f(x));

dg2dx2Exato = double(subs(dfdx, x, xs));

ret = [dgdxExato; dg2dx2Exato];

end

análise dos resultados)

De seguida encontra-se code.m, o ficheiro que constrói a tabela, e a própria.

clear;

K = 0.9;

maxerror = 0.001;

h = 0.00000000000000000001;

xs = sym([]);

for i = 1 : 11

xs(i) = (i-1) \* (pi/5);

end

strXs = string(xs);

xs = double(xs);

format long;

t1 = A(xs, maxerror);

t2 = B(xs, t1(1, :));

t3 = B(t1(1, :), t2(1, :));

t4 = C(xs);

T = array2table( ...

[t1 ; t2 ; t4(1, :); t3; t4(2, :)], 'VariableNames', strXs,'RowNames',{ ...

'g(x)',

'Erro Absoluto Menor Que',

'nº de trapézios',

'dg/dx calculado' ,

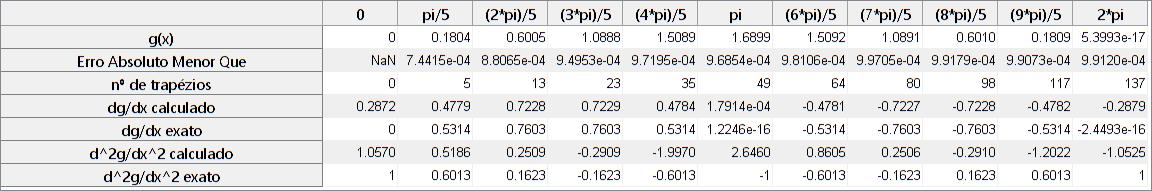
'dg/dx exato',

'd^2g/dx^2 calculado' ,

'd^2g/dx^2 exato'

});

uitable('Data',T{:,:},'ColumnName',T.Properties.VariableNames,...

 'RowName',T.Properties.RowNames,'Units', 'Normalized', 'Position',[0, 0, 1, 1]);

Os resultados da a) mostram que para um domínio maior entre e , maior será necessário o número de trapézios de forma a manter um erro baixo.

Os resultados da b), comparado com os resultados da c), mostram uma grande imprecisão. Isto deve-se ao (aqui sempre igual a ) estar muito longe de 0.