PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

CURSO DE ENGENHARIA DA COMPUTAÇÃO

AED VI – DIFERENCIAÇÃO NÚMERICA

Goiânia

2018

LUCAS MACEDO DA SILVA

AED VI – DIFERENCIAÇÃO NÚMERICA

Trabalho apresentado por Lucas Macedo da Silva e, para avaliação e aplicação dos conceitos aprendidos em atividade extra disciplinar.

Goiânia

2018

1 ENUNCIADO DO PROBLEMA

A aplicação dos conceitos de diferenciação numérica se deu na resolução do problema descrito na figura a seguir.

Figura 1 – Enunciado do Problema.

- Utiliza a regra

$$f'(x) \approx \frac{1}{h}[(x+h) - f(x)]$$

- Para aproximar a primeira derivada da função $f(x) = senx$ em $x = 0.5$.
- Implemente o pseudocódigo e explique o que acontece quando um grande o número de iterações é executado.

Fonte: Atividade extra disciplinar 6 (2018)

O pseudocódigo mencionado se encontra na figura 2, a seguir.

Figura 2 – Pseudocódigo

- O pseudcódigo

```
n = 50
x = 0.5
h = 1
emax = 0
para i = 1 até n faca
    h = 0.25 * h
    y = [sin(x + h) − sin(x)]/h
    erro = |cos(x) − y|
    escreva(i, h, y, erro)
    se erro > emax
        então emax = erro
              imax = i
    fim se
fim para
escreva(imax, emax)
```

Fonte: Atividade extra disciplinar 6 (2018)

2 CÓDIGO DESENVOLVIDO

A partir do enunciado presente na figura 1, o pseudocódigo da figura 2, foi implementado no *software* Octave, o código denominado “dif_numerica” a seguir.

```
function dif_numerica(n)


 %n = Numero de iteracoes
 x = 0.5;
 h = 1;

 imax = 0; %Numero da iteracao em que o erro foi maximo
 emax = 0; %Erro maximo

 for i = 1: n

   h = 0.25*h; %Passo

   y = (sin(x + h) - sin(x))/h; %Regra

   erro = abs (cos(x) - y); %Erro
   % Escrita dos valores
   i
   h
   y
   erro
   printf('\n')
   if erro> emax
     emax = erro
     imax = i;
   endif

 endfor
```

```
  % Escrita dos valores
  imax
  emax
endfunction
```

3 RESULTADOS OBTIDOS

A execução do código no ambiente de desenvolvimento e a análise da saída implicou que quanto maior o número de iterações, mais próximo da derivada naquele ponto está o erro. Isso ocorre devido ao passo h, ser tão pequeno que y se torna 0 e o erro se torna a própria derivada aplicada naquele ponto.

Por exemplo ao executar o código para n = 10, no ponto x = 0.5, definido no enunciado do problema, obtemos:

Figura 3 – Saída do código para n = 10

```
i =  10
h =   9.53674316406250e-007
y =  0.877582333283499
erro =   2.28606873875492e-007

imax =  1
emax =  0.0687296762138483
```

Fonte: O autor (2018)

O valor correto da derivada no ponto é:

Figura 4 – Valor correto da derivada

```
>> cos(0.5)
ans =  0.877582561890373
```

Fonte: O autor (2018)

Pela análise dos valores percebe-se que até a sexta casa decimal os valores foram iguais (y presente na figura 3 e *ans* presente na figura 4), porém a partir da mesma eles divergem. Com isso ao aumentarmos o número de iterações, aumentamos também a precisão do código, ou seja, o valor calculado pelo método numérico se aproximado do valor calculado aplicando-se um método mais analítico.

Porém após a 27 º iteração o valor de y se torna 0, e o valor de emax (erro máximo) se torna o próprio valor da derivada naquele ponto, o que faz sentido visto que y = 0. Com isso até a 14 º iteração é possível perceber que o valor calculado pela regra e o valor real estão próximos até a 8 º casa decimal, sendo possível perceber nas figuras 5 a seguir e na figura 4 que contém o valor de *cos(0.5)*.

Figura 5 – Valor calculado para n = 14

```
i =  14
h =   3.72529029846191e-009
y =  0.877582564949989
erro =   3.05961656010822e-009

imax =  1
emax =  0.0687296762138483
```

Fonte: O autor (2018)

Executando o código com n = 50, obtém-se:

Figura 6 – Valor calculado para n = 50

```
i =  50
h =   7.88860905221012e-031
y = 0
erro =  0.877582561890373

imax =  27
emax =  0.877582561890373
```

Fonte: O autor (2018)

A partir da análise das saídas anteriores percebe-se que o código, gera resultados diferentes até a 27 º iteração que é quando y se iguala a zero. Portanto pode-se afirmar que quanto maior o número de iterações a serem executadas pelo algoritmo, mas próximo de 0 o valor de y ficará, até o momento em que o mesmo se tornará zero, e o valor de emax, se tornará o próprio valor da derivada.

A analise também permite afirmar que até a 14 º iteração o código gera uma boa aproximação entre o valor real e o calculado, sendo, portanto, uma boa estimativa para quantidade de iterações a serem utilizadas na execução do código. Não gerando assim, um momento em que o código apenas atualiza o valor do passo h e não gera um valor mais aproximado ainda para a derivada.