

MS211 - Cálculo Numérico - Turma J - Atividade 2

Luiz Henrique Costa Freitas RA: 202403

Gabriel de Freitas Garcia RA: 216179

Vanessa Vitória de Arruda Pachalki RA: 244956

8 de Outubro de 2020

Enunciado Atividade 2

Podemos utilizar os métodos para encontrar raízes de funções para estimar a raiz quadrada de um número. Considere a função $f(x) = x^2 - A$, onde A representa o número que queremos estimar a raiz quadrada. Vamos utilizar o método de Newton para tanto.

1. Escreva um pseudocódigo para o método de Newton e implemente como uma função do Octave.
2. Implemente a função $f(x)$ em Octave deixando A como um parâmetro de entrada.
3. Teste sua função para $A = 13$. Use $x_0 = 3$ como aproximação inicial e erro $\epsilon = 10^{-8}$.
4. Teste sua função para um valor de A inteiro à sua escolha, desde que não tenha raiz inteira. Use o mesmo ϵ do item anterior. Como x_0 use um inteiro que deve ser a raiz quadrada de algum número menor que A .
5. Repita esse experimento com o método das secantes. Utilize x_1 um inteiro que é a raiz quadrada de um número maior que A .
6. Compare o comportamento dos dois métodos nos exemplos acima.

1 Respostas

Para resolver o item 1, desenvolvemos o seguinte algoritmo:

```
1 f = minha_funcao(x)
2 f_linha = derivada(f)
3 x = x0
4 while(f(x) > erro):
5     x = x - f(x)/f_linha(x)
```

A partir deste pseudocódigo e assumindo as funções $f(x)$ e $f'(x)$, o valor inicial x_0 e o erro ϵ anteriormente definidas podemos representar este mesmo algoritmo em Octave:

```
1 function res = newton_loop(x,f,f_linha,erro)
2 k = 0;
3 while(abs(f(x)) > erro)
4     x = passo_newton(x,f,f_linha);
5     k++;
6     disp("Aproximação atual:");
7     disp(x);
8 endwhile
9 res = x;
10 disp("Número de passos:");
11 disp(k);
12 endfunction
```

```
1 function x_res = passo_newton(x,f,f_linha)
2     x_res = x - f(x)/f_linha(x);
3 endfunction
```

Para o item 2 definimos a função $f(x)$ desta maneira:

```
1 A = input("Digite um número inteiro:\n");
2 f = @(x) x.^2 - A;
```

E com esta definição de função obtivemos o seguinte código para calcular o Método de Newton para a equação dada:

```

1 # Jesus quer transformar sua vida
2 format long;
3 A = input("Digite um número inteiro:\n");
4 f = @(x) x^2 - A;
5 f_linha = @(x) 2*x;
6 x = input("Digite o valor inicial:\n");
7 erro = input("Digite o valor de erro: \n");
8 x = newton_loop(x,f,f_linha,erro);
9 disp("x final:");
10 disp(x);
11 disp("f(x) final: ");
12 disp(f(x));

```

Com este programa obtivemos as repostas para o item 3

```

1 >> lab02
2
3 Digite um número inteiro:
4 13
5 Digite o valor inicial:
6 3
7 Digite o valor de erro:
8 10^-8
9 Aproximação atual:
10 3.666666666666667
11 Aproximação atual:
12 3.606060606060606
13 Aproximação atual:
14 3.605551311433664
15 Aproximação atual:
16 3.605551275463990
17 Número de passos:
18 4
19 x final:
20 3.605551275463990
21 f(x) final:
22 1.776356839400250e-15

```

Semelhantemente obtivemos o resultado para o item 4:

```

1 >> lab02
2 Digite um número inteiro:
3 47
4 Digite o valor inicial:
5 6
6 Digite o valor de erro:
7 10^-8
8 Aproximação atual:
9 6.916666666666667
10 Aproximação atual:
11 6.855923694779117
12 Aproximação atual:
13 6.855654605682009
14 Aproximação atual:
15 6.855654600401045
16 Número de passos:
17 4
18 x final:
19 6.855654600401045
20 f(x) final:
21 7.105427357601002e-15

```