E02 — ANN0001 — CCI

Aluno: ENDREW RAFAEL TREPTOW HANG Submeter até: 23/08/2019 23:59hs

Q1 Usando o método iterativo de Newton, com estimativa inicial $p_1 = 1.776$, para aproximar uma raiz da função $f(x) = x^3 - 4x - 1$, encontre p_5.

a) 2.1166054 *b*) 2.1169058 *c*) 2.1165586 *d*) 2.1166435 *e*) 2.1164676 2.1149075

Q2 Usando o método iterativo das Secantes, com estimativas iniciais $p_1 = -0.592$ e $p_2 = 0.663$, para aproximar uma raiz da função $f(x) = x^3 - 4x - 1$, encontre p_5.

a) −0.2530432 *b*) −0.2521511 *c*) −0.2523453 *d*) −0.2524095 −0.2541034 *f*) −0.2522431

Q3 Usando o método iterativo da Posição Falsa, com estimativas iniciais $a_1 = -0.801$ e $b_1 = 0.774$, para aproximar uma raiz da função $f(x) = x^3 - 4x - 1$, encontre p_5.

a) −0.2526359 *b*) −0.252157 *c*) −0.2523866 *d*) −0.2527929 −0.2541016 *f*) −0.2521024

Ex 01)

```
1.776
x1 = 1.7760000
x2 = 2.2340633
x3 = 2.1234253
x4 = 2.1149560
x5 = 2.1149075
```

Ex 02)

```
-0.592
0.663
x1 = -0.5920000, x2 = 0.6630000
x2 = 0.6630000, x3 = -0.2698523
x3 = -0.2698523, x4 = -0.2535539
x4 = -0.2535539, x5 = -0.2541035
```

Ex 03)

```
-.801
.774
-0.30087302
-0.25112930
-0.25413141
-0.25410171
-0.25410169
```

Codigo 1)

```
#include <stdio.h>
#include <math.h>

double func(double x){
return pow(x, 3) - 4*x -1;
}

double derfunc(double x){
return 3 * pow(x, 2) - 4;
}

int main(){
double p1, res;
scanf("%lf", &p1);
res = p1;
for (size_t i = 1; i <= 5; i++)
{
printf("x%lu = %.7lf\n", i, res);
res = res - func(res) / derfunc(res);
}
}
```

Codigo 2)

```
#include <stdio.h>
#include <math.h>

double func(double x){
return pow(x, 3) - 4*x -1;
}

int main(){
double x[6];
scanf("%lf%lf", &x[0], &x[1]);
for (size_t i = 1; i < 5; i++)
{
printf("x%lu = %.7lf, x%lu = %.7lf\n", (i), x[i - 1], (i + 1), x[i]);
x[i + 1] = -x[i] * func(x[i - 1]) + x[i - 1] * func(x[i]);
x[i + 1] /= (func(x[i]) - func(x[i - 1]));
}
}
```

Codigo 3)

```
#include <stdio.h>
#include <math.h>

double func(double x){
return pow(x, 3) - 4*x -1;
}

int main(){
double x, y;
double xn;
scanf("%lf%lf", &x, &y);
for(int i=0;i<5;i++){
xn = x*func(y) - y*func(x);
xn /= (func(y) - func(x));
if(func(xn)*func(x) < 0)
y = xn;
else
x = xn;
printf("%.8lf\n", xn);
}
}
```