

# Relatorio

Lourenço Henrique Moinheiro Martins Sborz Bogo - 11208005

## Parte 1 do EP: Método do Ponto Fixo

Para a primeira parte do EP, eu escolhi a função de ponto fixo  $g(x) = x - \frac{f(x)}{f'(x)}$ . Como essa é a função do Método de Newton, se usarmos um ponto inicial "próximo" o suficiente da raiz, dentro de um certo intervalo, iremos achá-la.

Para achar cada uma das raízes foi usada precisão de  $10^{-8}$  e os seguintes pontos iniciais:

$$\begin{cases} X_0 = 1.5 \rightarrow 1.487962 \\ X_0 = -0.5 \rightarrow -0.539835 \\ X_0 = 2.5 \rightarrow 2.617867 \end{cases}$$

Essa parte do EP é implementada em apenas um arquivo, o fixed.c, que contém tudo que foi pedido e está devidamente comentado.

## Detalhes de Implementação:

- Optei também por usar ponteiros de funções pois isso facilitaria o resto da implementação.

## Parte 2 do EP: Método de Newton

Fiz essa parte do EP de uma maneira um pouco diferente da que foi pedida. Fiz uma pequena interface para que seja mais fácil utilizar o EP sem ter que mexer no código fonte. Ele pede primeiro para que seja passado quantos pixels serão calculados em cada direção, depois, pede os pontos u e l, respectivamente, e por último são oferecidas 6 funções para plotar.

O programa usa precisão de  $10^{-8}$  para aplicar o método de newton e  $10^{-5}$  para diferenciar as raízes. A segunda precisão maior pois caso isso não seja feito, acharíamos várias vezes a mesma raiz, em consequência das aproximações de double feitas pela computador.

Escolhi algumas funções que estavam em artigos que li sobre fractais de Newton, coma a minha principal fonte sendo a wikipedia.

As funções implementadas no EP são:

- $x^4 - 1$
- $x^2 + 1$
- $x^3 - 1$
- $x^3 - x$
- $x^8 + 15x^4 - 16$
- $\sin(x) - 1$

Os resultados dos experimentos podem ser vistos ao rodar o programa para uma das funções. O programa irá gerar um output chamado grafico.png, que é a imagem das bacias de convergência da função escolhida. O outro modo de utilizar o EP é rodar o script rodado.sh, que irá criar 7 bacias de convergência de uma vez e deixar suas imagens na pasta (paciência ao rodar esse script, ele demora um pouco).

**Implementação:** O EP é dividido em vários arquivos, cada um deles com suas respectivas funções a seguir:

**newton.c** É o arquivo principal, que calcula todos os pontos, os guarda em um arquivo chamado out.txt e depois chama o script printa.sh. O arquivo é dividido em várias funções, todas devidamente comentadas.

**printa.sh** É o arquivo que chama o gnuplot e plota os gráficos a partir do arquivo out.txt, deletando todos os arquivos auxiliares após.

**grafico.png** É a imagem da bacia de convergência.

### **Detalhes de Implementação:**

- Decidi usar uma matriz de ponteiros de funções para armazenar as funções matemáticas e suas derivadas, pois isso facilitaria o resto da implementação.
- Decidi usar 30 como limite de iterações pois como a convergência é quadrática (rápida), é muito provável que isso seja suficiente para achar a raiz. Colocar mais iterações deixaria o programa mais lerdo, e poderia causar problemas ao achar as raízes, como por exemplo, achar algumas que não existem.

- Decidi economizar o máximo possível no número de prints durante a execução do programa pois isso deixaria o programa muito mais lerdo também.
- O script rodado.sh calcula 1000x1000 pontos, pois isso é o suficiente para conseguir uma ótima resolução do fractal.
- Optei por deixar as imagens dentro do tar, pois é possível que quem for rodar o EP não tenha algum interpretador de bash, impedindo o uso dos scripts.