

Projeto Calculadora de Polinômios em C

GABRIEL LAZARI TREVISANI

September 2024

1 Motivação

Esse projeto se originou da minha necessidade de relembrar a programar, já que esqueci muita coisa durante a greve. Ele utiliza as seguintes ferramentas: condicionais, loops, ponteiros, alocação dinâmica, struct, arquivo header/Makefile e recursão, não por coincidência as que eu queria voltar a dominar. A IDE foi o terminal do linux.

2 Objetivos

O objetivo é criar um programa que recebe uma string do usuário com o polinômio P , como $2 - x^2 + 4x$, e permite que ele faça as seguintes operações:

- calcular o valor de $P(x)$ para qualquer $x \in \mathbb{R}$, bem como aproximações lineares e quadráticas de $P(x)$ em 3 pontos próximos ao escolhido para fins didáticos
- calcular o valor da pseudo-área sobre a curva de $y = P(x)$ por meio do Teorema Fundamental do Cálculo, bem como aproximações usando o conceito da "Integral de Riemman" para 3 valores diferentes de N , novamente, para fins didáticos
- calcular as raízes, se houver, deste polinômio, através de um algoritmo recursivo que usa as raízes da derivada para achar pontos ideais para a aplicação do método de Newton

3 Detalhes do Projeto

3.1 A struct Polinômio

Cada polinômio possui um vetor char input, com tamanho estático N , e sinal, alocado dinamicamente, bem como um vetor inteiro v de coeficientes, float r de raízes, alocados junto com sinal. Também possui inteiros com o valor do grau de p , expoentes, coeficiente e o número de raízes reais.

3.2 A função `acha_grau` e `cria_vetor`

Utilizadas uma seguida da outra, a função `acha_grau` recebe como input o vetor `char` preenchido pelo usuário. Ela determina o grau do polinômio e aloca os outros 3 vetores com o tamanho do grau. Em seguida, zera todos os valores deles.

Depois, chama-se a função `cria_vetor`, cujo objetivo é preencher `v` e `sinal` tendo como base o `char` input, preenchido pelo usuário.

3.3 Integrando e Derivando

Criou-se uma função derivada, que recebe os polinômios `p` e `d` e preenche todos os campos possíveis de `d` com base em `p`, exceto raízes, que é mantido zerado.

Criou-se também uma função primitiva que retorna o valor float da primitiva em um ponto escolhido, assumindo constante $C = 0$. Ela é chamada pela função `area`, que retorna a pseudo-área entre dois pontos escolhidos pelo usuário.

Optou-se por essa abordagem porque não será necessário guardar o valor da primitiva, já que ela é utilizada apenas para calcular o valor da área. A função derivada, por outro lado, é utilizada fortemente no método de Newton e na função `acha_raiz`, que precisa das raízes de `d`. Como ela é recursiva, é necessário salvar os dados de `d` como foi feito em `p`.

3.4 Função `acha_raiz`

Definitivamente minha favorita, ela parte da ideia de que é possível encontrar as raízes se `p` for linear ou se soubermos as raízes da derivada, pois assim podemos encontrar pontos ideais para aplicar o método de Newton. Desse modo, caso o grau de `p` seja maior que 1, faz-se a recursão com a derivada de `p`. O ciclo de recursão termina quando a n -ésima derivada for linear. A função, então, vai usando as raízes das derivadas anteriores para calcular as raízes das derivadas seguintes, até retornar em `p`. Mais detalhes estão disponíveis em "Fundamentação Matemática", seção 4.3.

4 Limitações e comentários

- É necessário que os coeficientes estejam à esquerda de `x`, enquanto os expoentes devem estar à direita. Creio que não há problemas em relação a essa limitação.
- Não é necessário escrever `1*`, `^ 0` ou `^ 1`, e a ordem dos elementos do polinômio não importa
- O coeficiente são inteiros e deve ser ≤ 9 , e os expoentes devem pertencer aos inteiros. Poderia colocar um loop a mais em `cria_vetor` para ter coeficientes inteiros do tamanho que desejar.

- a função valor necessita dos vetores sinal e v. Poderia ter otimizado e colocado os coeficientes com os sinais corretos em v em cria_vetor

5 Considerações finais

Acredito que a parte de conferência de erros do código foi relativamente simples, só sendo um pouco trabalhosa na função "acha_raiz", na qual optei por deixar os printf's que usei no processo para que fosse possível receber um feedback da minha metodologia.

De modo geral, fiquei satisfeito com o resultado e lamento estar voltando as aulas tão cedo. Acho que o projeto merece as melhorias supracitadas na seção anterior. No entanto, sei que é melhor investir meu tempo em projetos que utilizem as ferramentas de ED1, os quais espero fazer futuramente.

Agradeço de coração a leitura desse documento até o final e ficaria muito feliz se pudesse me dar um feedback geral do projeto.