UNIVERSIDADE FEDERAL DO ESPIRITO SANTO

Trabalho de Algoritmos Numéricos

Relatório

Leonardo Khoury Picoli
Matheus Salomão

Vitória
Dezembro de 2018

UNIVERSIDADE FEDERAL DO ESPIRITO SANTO

Leonardo Khoury Picoli
Matheus Salomão

Trabalho de Algoritmos Numéricos

Relatório

Trabalho referente à disciplina de Algoritmos Numéricos do curso de Ciência de Computação do Departamento de Informática da Universidade Federal do Espírito Santo.

Vitória
Dezembro de 2018

Sumário

1 Introdução

Muitas vezes não é possível resolver uma função analíticamente, seja pelo grau de complexidade da mesma ou por não tê-la disponível, porém apenas com a amostragem de alguns pontos é possível contruir um polinômio que seja igual ou parecido com a função, e para isso podemos utilizar vários métodos, e então resolver o problema numericamente.

$$p(x) \simeq f(x) \tag{1}$$

onde f é uma função desconhecida, e $p(x)$ é o polinômio com pontos em comum à f.

Obtendo uma função $polyfunc1(x)$ que possui seu código fonte ofuscado(tornando assim essa função desconhecida), é possível apenas fazer uma amostragem dos pontos para então seguir com uma solução numérica.

x	x_0	x_1	$\dots$	x_n
y	y_0	y_1	$\cdots$	y_n

Quanto maior a quantidade de pontos, mais próximo à função ele fica, isso ficará evidente durante os módulos feitos no trabalho, além também do fato de que a convergência muda de método para método.

2 Objetivos

O objetivo deste trabalho é resolver numericamente problemas que envolvem funções unidimensionais:

$$\begin{aligned} f : R &\to R \\ x &\mapsto y. \end{aligned} \tag{2}$$

Para tanto, serão seguidos os módulos e as tarefas contidas nos mesmos, sempre exibindo na saída padrão, isto é, no terminal, os resultados e respostas necessários, e também plotando os gráficos, tornando a visualização dos resultados mais fácil e evidente.

3 Metodologia

O código do programa é subdivido em módulos assim como a especificação, onde em cada um são realizadas as tarefas do mesmo, como os resultados de um módulo podem ser necessários em módulos seguintes, também há a comunicação entre ele. Esses módulos são chamados por uma main.m, que executará todas as funções pedidas.

Para auxiliar na compreensão do código cada umas das funções será explicada separadamente.

3.1 Módulo 1: Função $polyfunc1(x)$: Interpolação

Considerando a função $polyfunc1()$ é possível obter as amostragens discretas da função, logo, podemos encontrar um polinômio que seja igual ou parecido com essa função, seguindo as tarefas iremos obter um gráfico da interpolação.

3.1.1 Tarefa 1

Com o objetivo de isolar raízes, analisamos manualmente o comportamento da função e definimos um limite inferior e um limite superior como [-1,3].

3.1.2 Tarefa 2

Um plot do gráfico é feito usando os limites definidos na tarefa anterior, com uma resolução de $\Delta x = 0.01$, também é desenhado o eixo y = 0.

3.1.3 Tarefa 3

Através da função difdiv calculamos a tabela de diferenças divididas usando $x_0 = -1, n = 10, \Delta x = 0.4$ e passando um vetor X e Y como parâmetros para a função. Os vetores X e Y precisaram ser transpostos para serem utilizados na função. A tabela é exibida através da função $showDD()$;

3.1.4 Tarefa 4

Através de um loop, analisando os coeficientes de cada grau até que seja encontrado um de grau nulo. Quando isso ocorre o loop é finalizado e a variável count para de ser incrementada.

O valor de $count$ deve ser subtraído de 2 pois o loop termina apenas quando é encontrado uma ordem com coeficientes nulos, quando na verdade a resposta corresponde a ordem anterior. Além disso, a coluna 1 da tabela corresponde aos valores de x.

3.1.5 Tarefa 5

Uma função observa a tabela de diferenças divididas e conclui se é ou não um polinômio, olhando o grau com coeficientes não nulos, caso não seja um polinômio, isso acontece logo.

3.1.6 Tarefa 6

Usando um único ponto $x0$ o polinômio de Newton é calculado com a função interpol-Newton e então desenhado no gráfico. Observamos o grau através da quantidades de coeficientes - 1.

3.1.7 Tarefa 7

Aumentando a quantidade de pontos de 1 até ordem+1 (ordem encontrada na tarefa 4) é feito a interpolação de Newton, achando assim $p_0(x) \ldots p_{ordem}(x)$

3.1.8 Tarefa 8

Desenha no gráfico os polinômios obtidos na tarefa 7.

3.2 Módulo 2: Função $polyfunc1(x)$: Integração

Utilizando a mesma função do módulo 1, $polyfunc1(x)$, agora o foco é realizar uma integração numérica. Para tanto são utilizados os seguintes métodos:

- Regra do Trapézio (Repetida)
- Regra de Simpson (Repetida)
- Quadratura Gaussiana

Os limites inferior e superior são os mesmos do módulo 1, isto é, a = -1 e b = 3.

Para realizar as integrações são utilizadas as funções integralTrapeziosRepetidaFunc,integralSimpsonRepetidaFunc e integralGaussLegendreFunc. Além das integrações são utilizados os métodos de erro referentes a cada tipo de integração. Para tanto são passados os valores de a e b, o número de iterações (que vai de 1 até 15) e os coeficientes da $polyfunc1(x)$ que foram calculados no módulo1.

Os valores calculados são colocados em uma tabela assim como foi pedido no enunciado. A tabela é exibida no terminal no final do código.

3.2.1 Tarefa 1

Calculamos e exibimos a tabela com os resultados, onde as colunas são o resultado do método e o erro calculado, as linhas são a quantidade de iterações, nesse caso, 15.

3.3 Módulo 3: Raízes

Neste módulo utilizamos de um método iterativo para o cálculo das raízes.

3.3.1 Tarefa 1

É feito uma estimativa dos coeficientes do polinômio $p_n(x)$ com a função $polyfit(x, y, n)$ onde x e y são os pontos de amostragem da função e n é a $ordem$ obtida no módulo 1 tarefa 4.

3.3.2 Tarefa 2

Utilizamos a função auxiliar

$$u(x) := \frac{f(x)}{f'(x)} \tag{3}$$

Para determinar as raízes da função $f(x)$, uma vez que ela possui as mesmas raízes mas sempre troca de sinal na raiz. Para calcular a função $u(x)$ fizemos o seguinte cálculo: $u(x) = polyval(Coeficientes, X)/polyval(Coeficientes\ da\ derivada, X)$ em seguida, um loop compara os valores de $u(x)$ com os valores de $polyfunc1()$ e quando ambos são zero, temos uma raíz.
Após todos os dados serem calculados, os resultados são plotados em um gráfico.

3.4 Módulo 4: Função $alien(x)$

Neste módulo calculamos numericamente as informações desejadas, semelhante aos outros módulos, porém dessa vez usando a função ofuscada $alien(x)$

3.4.1 Tarefa 1

Calculamos e exibimos em um gráfico a amostragem da função $alien(x)$ no intervalo [0,100], com uma resolução de x = 0.1.

3.4.2 Tarefa 2

Assim como no módulo1, a tabela de diferenças divididas é calculada através da função $difdiv$,entretanto, dessa vez utilizamos uma versão não recursiva da função difdiv porque o número de chamadas recursivas estava muito alto,ocasionando erros no programa. Através da tabela é determinado o grau do polinômio , que será passado como parâmetro para função $polyfit$, junto com um intervalo e a imagem da função, para que os coeficientes da função $alien(x)$ possam ser estimados e assim possa ser determinado a interpolação de grau n da função.

3.4.3 Tarefa 3

Nesse módulo utilizamos as informações obtidas até então para plotar o polinômio $p_n(x)$.

3.4.4 Tarefa 4

As raízes da função foram encontradas a partir da função $roots$.Não foi possível realizar isso através de outro método. (Essa foi a única parte do trabalho onde essa função foi utilizada)

Para criar uma tabela contendo as raízes e sua multiplicidade primeiramente pegamos apenas a parte real das raízes. Em seguida criamos um conjunto a partir do vetor de raízes através da função $unique$. Verificamos quantas vezes cada elemento do conjunto aparece no vetor de raízes para determinar sua multiplicidade, depois preenchemos a tabela com os resultados obtidos.

3.4.5 Tarefa 5

A partir das funções sum e $prod$, o somatório e o produtório das raízes calculadas pela tarefa anterior são determinados. Com esses dados são realizados os cálculos apresentados na documentação do trabalho para determinar as coordenadas do quartel geral do comando superior dos alienígenas.

3.5 Função Principal

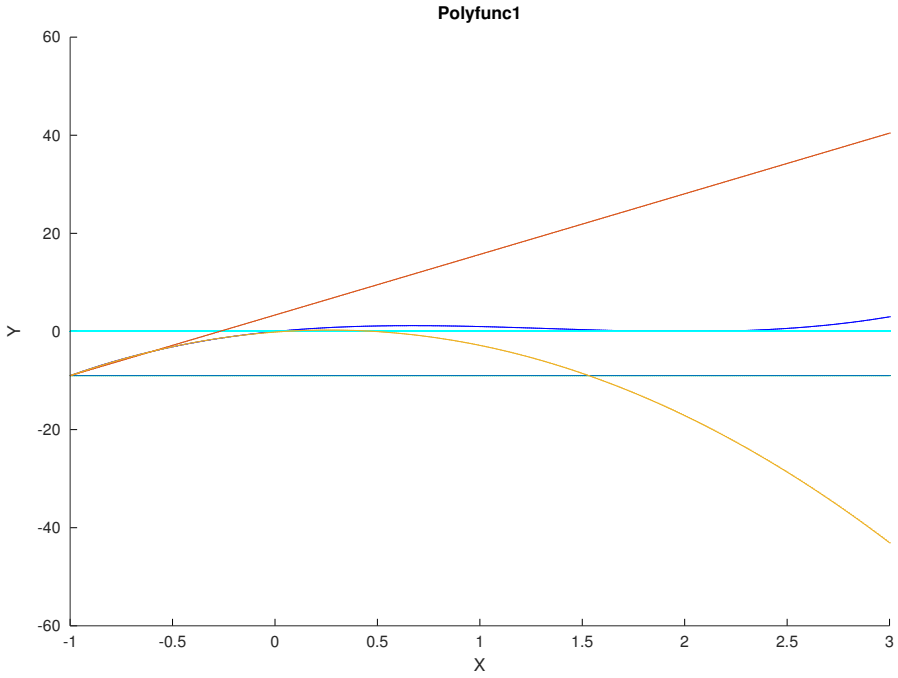
A função principal faz a chamada de cada módulo, para que fique fácil e prático calcular e exibir todos os resultados desejados de uma só vez, de forma que os gráficos de cada módulo fiquem separados em diferentes janelas de plot.

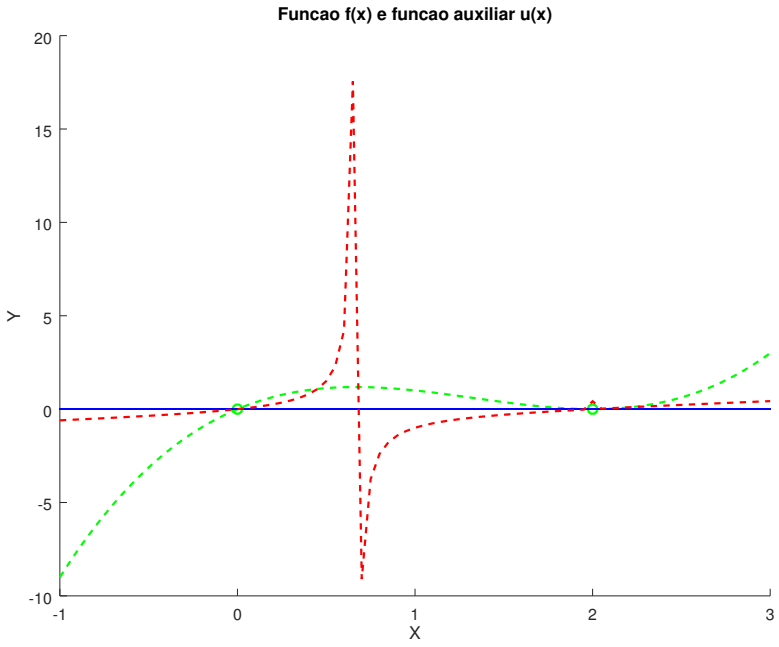
4 Resultados e Avaliação

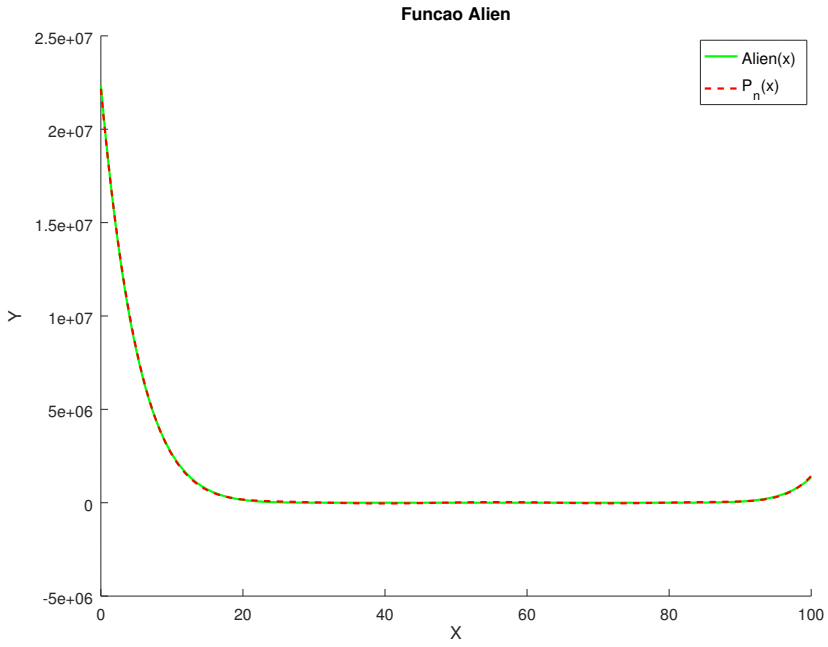
4.1 Resultados

4.1.1 Gráficos

Segue abaixo os gráficos plotados respectivamente pelos módulos 1,3 e 4.




Funcao f(x) e funcao auxiliar u(x)
20
15
10
5
0
-5
-10
Y
-1
0
1
2
3
X


Funcao Alien
Alien(x)
P n (x)
2.5e+07
2e+07
1.5e+07
1e+07
5e+06
0
-5e+06
Y
0
20
40
60
80
100
X

4.2 Avaliação

Avaliando os resultados, concluímos que os métodos de Interpolação possuem convergência diferente,tendo assim uns com melhor aproximação do que outros dependendo do intervalo escolhido, assim como os métodos de Integração possuem majorantes de erros também diferente, variando assim junto com o nível de complexidade do cálculo de cada método, onde os mais complexos possuem menor erro. Os métodos são muito úteis para calcular numericamente, seja para funções desconhecidas ou muito complexas, possibilitando aproximações realmente boas, que podem ser conferidas no resultado do trabalho.

5 Referências

- https://pt.wikipedia.org/wiki/Interpola%C3%A7%C3%A3o_polinomial
- https://inf.ufes.br/~luciac/cn/MatlabOctave.pdf
- https://inf.ufes.br/~thomas/gradua/calcnum/www/