UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ: CAMPUS DE FOZ DO IGUAÇU

CENTRO DE ENGENHARIAS E CIÊNCIAS EXATAS

**Cálculo Numérico**

***Atividade #2***

**Instruções**:

* Entrega individual, via “Tarefas” do Teams e arquivo único em .pdf;
* Use este arquivo .docx para fazer sua atividade, e ao finalizar, gere o .pdf.
* Além de incluir os algoritmos no .pdf, eles devem ser upados em anexo, cada um individualmente e um arquivo txt;
* **Discente**: Daniel Marques da Silva

1. Exercício 1 - Implementar em Python o método da Iteração de ponto fixo simples; apresente um exemplo, e o processo de cálculo para verificar a convergência.

|  |
| --- |
| **Resposta**:  Foi implementado em Python, um projeto que busca a raiz de uma função , onde foi elaborado uma função que ficasse encarregada do processo para não poluir muito o projeto. Os resultados para um *Xn=0,5* e Erro para critério de parada 0,001%. Os resultados obtidos pelo projeto são descritos na figura 1 a seguir. Ressalta-se que foi mantida uma condição de entrada para o usuário determinar o *Xn* desejado.    *Figura 1 – Resultado estimado pelo Ponto Fixo*  Inicialmente é pedido o *Xn* ao usuário, com o programa rodando logo após a confirmação e trazendo os resultados: Raiz estimada na primeira linha, “Lista” da progressão do Erro relativo e o número de interações necessária.  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Implementar em Python o método de Newton-Raphson

|  |
| --- |
| **Resposta**:  O método de Newton-Raphson é considerado um mais aplicados nos mais diversos ramos da ciência, como forma de testado foi utilizado uma função e sua respectiva derivada para definir a raiz de sua função.  Seu método é definido como  A figura a seguir demonstra os resultados estimados pelo método, com um Xo=3. A forma de demonstrar esses foi mantida igual ao do exercício anterior.    *Figura 2 – Saída do Método*  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Newton-Raphson em SciLab e Excel

|  |
| --- |
| **Resposta**:  Conforme proposto, foi implementado o método de newton em SciLab e Excel. Poucas alterações foram necessárias para migrar o programa para a linguagem compreendida pelo SciLab. No caso do Excel, esse em geral sai de forma muito mais simples, uma vez que é utilizado essa plataforma para gerar a lógica necessária para qualquer método apresentado anteriormente. As figuras apresentam, respectivamente as saídas do Scilab e do Excel. É descrito em vermelho as funções de ambos os projetos.    *Figura 3 – Respostas dadas pelo SciLab*    *Figura 4 – Tabela contendo o método de Newton, assim como as f(x) usadas*  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Implementar em Python o método da Secante

|  |
| --- |
| **Resposta**:  No método da secante, as condições para sua conversão são dadas por  Onde xa equivale a xi-1, que deve ser indicado pelo usuário também, um inconveniente pois esse método necessita um ponto de inicio e um ponto anterior a esse.  Como resposta, ele obteve uma rápida conversão para , com os resultados sendo apresentados a seguir, na mesma estrutura que o Ex 1.    *Figura 5 – Resultado da Secante*  Os resultados aqui são, Raiz em 1, erro caiu instantaneamente para 0 (esse Erro se deve ao uso da raiz verdadeira e um valor abaixo dela) e 1 interação.  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Implementar em Python o método da Secante Modificado

|  |
| --- |
| **Resposta**:  Para a secante modificado, com determinação da raiz com o uso de um ponto de análise e um qsi (valor infinitesimal após o valor previamente definido) é feito pela expressão .  Para o resultado apresentado a seguir, temos a mesma função fornecida anteriormente, contudo, o resultado possui algumas variações interessantes. Começando com a raiz, que foi estimada e não só “retornada”, e o seu erro que foi sendo definido também.    *Figura 6 -Resultado da Secante Modif.*  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Comparação de todos os métodos já estudados

|  |
| --- |
| **Resposta**:  Conforme proposto, foram realizadas 5 interações com os métodos abertos e estimadas suas características e seu tempo de simulação para a . Essas são apresentadas na tabela a seguir.    *Figura 7 – Tabela com características das múltiplas aplicações dos métodos*  Obs. Código-Projeto encontra-se em “.txt” em anexo. |

1. Explicação das particularidades dos métodos para cada gráfico

|  |
| --- |
| **Resposta**:  Função 1:   * Bissecção: Poderia não achar a raiz, pois ela não cruza o eixo x, e é necessário esse cruzamento como critério de análise do método; * Falsa Posição: Realizaria muitas retas até encontrar a raiz, como a função não cruza o eixo pode achar um valor não correto para a função; * Falsa posição modificado: seria pouco menos demorado quanto a falsa posição, sofreria os mesmos entraves que o seu não modificado; * Ponto fixo: Acharia essa raiz com relativa facilidade, não necessita de teste de sinais para achar o valor da raiz. * Newton: Poderia achar com poucas interações e com pouco erro essa raiz, contudo pela natureza da função pode acabar por divergir, já que a raiz toca o eixo x; * Secante: Segue a mesma proporção da falsa posição, podendo facilmente divergir devido a sua sucessão de estimativas; * Secante Modi: Possui uma progressão que deve ser controla por quis, se bem definido qsi, pode achar a raiz, do contrário pode divergir facilmente;   Função 2:     * Bissecção: Poderia achar a raiz com nítida facilidade função cruza o eixo x * Falsa Posição: Realizaria algumas retas até encontrar a raiz, pode se aproximar com relativa facilidade, mas nada muito bom quanto a bissecção; * Falsa posição modificado: seria pouco menos demorado quanto a falsa posição, sofreria os mesmos entraves que o seu não modificado; * Ponto fixo: Acharia essa raiz, demoraria algumas interações. * Newton: Poderia divergir com muita facilidade, quanto mais próximo da raiz, mais longe a derivada manda xn; * Secante: Segue a mesma proporção da falsa posição, pode ser mais eficiente ou não dependendo dos paramentos iniciais; * Secante Modi: Possui uma progressão por qsi onde pode achar a raiz, demora algumas interações se bem selecionado qsi;   Função 3:   * Bissecção é o mais indicado, os demais métodos demorarão (Newton) muito ou poderão divergir com muita facilidade; |

1. Descarga de Corona

|  |
| --- |
| **Resposta**:  Usando o método da bissecção, os resultados adquiridos para Req são descritos na figura a seguir. O método utilizado foi o da bissecção pois, para os métodos abertos foi constatado muitos erros de divisão por zero.    *Figura 8 – Resultados obtidos pelo método da Bissecção* |