Roteiro de Pedro – Trabalho 2 – 2017.2

Slide 2: A quarta questão pede para usar o método de Newton-Raphson para encontrar, com cinco dígitos significativos exatos, as raízes das funções fornecidas nas letras a, b e c, considerando os intervalos também dados. Logo em seguida, é pedido para determinar os pontos iniciais e a ordem de convergência em função das iterações. A ordem de convergência será mostrada ao final. A quinta questão pede para refazer a quarta questão, mas usando o método da secante. **Definindo** os pontos iniciais para o método de **Newton-Raphson**, escolhemos pontos no interior do intervalo. São eles **2,5**; **3,5**; **0,5**; **3,5** e **1,5**. Usando o **método da** **Secante**, precisamos de dois pontos, portanto usamos os próprios **extremos** dos intervalos fornecidos.

Slide 3: Plotando os gráficos das funções no MATLAB, podemos identificar a localização aproximada das raízes nos intervalos fornecidos.

Slide 4: Resumidamente, o método de Newton-Raphson é um método iterativo para se calcular a raiz de uma função utilizando, para isso, a derivada da função em um ponto inicial x1. É encontrado, portando o valor de x2 a partir do cruzamento do eixo x com a tangente à função no ponto x1, e esta é a nova estimativa da raiz. A terceira estimativa é calculada da mesma forma, mas tomando como referência a segunda estimativa. O valor da próxima estimativa é calculado pela **seguinte** equação. O processo continua até que o valor do erro calculado a cada iteração seja menor que uma tolerância estabelecida previamente pelo usuário.

Slide 5: O método da Secante é semelhante ao método de Newton-Raphson, porém dessa vez não é necessário ter a expressão da derivada da função, e sim dois pontos na vizinhança da raiz. A partir desses dois pontos é definida uma reta secante à função e o valor da estimativa seguinte é obtido através do cruzamento do eixo x com a reta secante, como pode ser visto na **seguinte** equação. O método tem o mesmo critério de parada que o método de Newton-Raphson.

Slide 6: A nossa implementação do algoritmo de Newton-Raphson em MATLAB foi em forma de função, onde f é a função matemática a ser utilizada, que pode ser um vetor de caracteres, uma function\_handle ou uma função inline; dfdx também é do mesmo tipo de f e é a derivada da função f; Xe é a estimativa inicial; tol é o valor de tolerância e imax é o número máximo de iterações. Raiz é a variável de saída.

Slide 7: Das linhas 15 a 40 temos apenas uma validação dos parâmetros da função. Esse trecho garante que f e dfdx são dos tipos: vetor de caractere, function\_handle e inline; e que Xe, tol e imax são do tipo numérico. Caso alguma dessas condições não seja satisfeita, a função mostra uma mensagem de erro na tela e é finalizada.

Slide 8: Na **linha 43**, definimos o formato dos números como longo com notação exponencial; na **linha 44**, imprimimos o cabeçalho das informações a serem exibidas; na **linha 45** temos um for que vai de 1 até o número máximo de iterações; na **linha 47**, calculamos o valor da próxima estimativa conforme a fórmula mostrada e armazenamos em Xi; na linha 48, calculamos o erro relativo da iteração atual e armazenamos na variável Ti; na **linha 49**, imprimimos os valores calculados na iteração atual; **na linha 52**, testamos se o erro atual é menor que a tolerância definida pelo usuário; em caso afirmativo, temos uma solução aproximada e ela é passada para a variável de saída raiz e o laço de repetição é finalizado; caso contrário, o laço continua executando; na **linha 57**, atualizamos o valor de Xe com o valor calculado na iteração atual, Xi; na **linha 59**, se i for igual ao número máximo de iterações, uma mensagem é exibida na tela com essa informação e o valor de Xi é atribuído à raiz; na **linha 63**, imprimimos o número total de iterações.

Slide 9: A nossa implementação do algoritmo da Secante em MATLAB também foi em forma de função e é bem semelhante ao de Newton-Raphson. Na declaração da função, o que muda é que não temos a **derivada** e temos dois valores de estimativa: x1 e x2.

Slide 10: Na validação de dados, o que **muda** também é com relação aos valores x1 e x2.

Slide 11: Na etapa de processamento, o algoritmo é quase idêntico ao de Newton-Raphson. O que gostaríamos de ilustrar com mais ênfase é que, na **linha 42**, a próxima estimativa é calculada com base nas duas estimativas anteriores, conforme a fórmula apresentada; na **linha 43**, o erro é calculado com relação a x3 e x2 e; nas **linhas 52 e 53**, as estimativas x1 e x2 são atualizadas com os valores de x2 e x3, respectivamente.

MATLAB – mostrar resultados.

Slide 12: Para mostrar a ordem de convergência, **alteramos** um pouco os códigos de **Newton-Raphson e da Secante**. Com **esses** comandos a mais, criamos dois vetores, um contendo o logaritmo do erro de cada iteração e outro contendo o número de cada iteração. Ao final, adicionamos comandos plot para gerar gráficos.

Slide 13: Esses são os gráficos de cada situação da quarta questão. Percebemos como o erro diminui bastante a cada iteração.

Slide 14: E esses são os gráficos de cada situação da quinta questão. Notamos comportamento semelhante aos da quarta questão.