T4\_CN 1ª questão

Roteiro de Pedro

1. Olá. Eu sou Pedro Mendonça e vou apresentar a 1ª questão do trabalho 4 de Cálculo Numérico.
2. Considerando a **seguinte** **tabela**, é pedido na **letra A** para ajustar os dados às funções dadas. Na **letra B**, é perguntado qual das funções se ajusta melhor aos dados. Na **letra C**, é pedido para usar funções do MATLAB para fazer o mesmo procedimento e para comparar os resultados obtidos.
3. O Método dos Mínimos Quadrados realiza ajustes de curvas. Quando temos um **conjunto** de dados, queremos achar uma curva que se ajuste aos dados mostrando sua tendência e que tenha o menor erro possível. A curva pode ser **linear**, **quadrática**, **cúbica** ou de diversas outras formas.
4. Numa boa escolha do tipo de curva, podemos obter resultados bastante satisfatórios. (50”)
5. Considerando uma curva **linear**, precisamos definir o menor erro possível para que a curva se ajuste aos dados fornecidos. Segundo o Método dos Mínimos Quadrados, o **erro** é calculado pela seguinte fórmula.
6. A partir da fórmula do erro, calculando as **derivadas parciais** e igualando a zero, encontramos os valores em que o erro é mínimo. Isso nos fornece um **sistema linear**. Esse sistema linear também pode ser visto **dessa forma**, onde os valores da matriz dos coeficientes e do vetor resposta é encontrado pelo produto interno, em que **n** é um vetor unitário.
7. Quando a função **não** é linear, devemos realizar a **linearização** dessa função, encontrando uma **função linear** equivalente, cuja solução já foi mostrada. Assim, facilmente encontramos os **valores** para os coeficientes da função não linear.
8. Para polinômios de graus superiores, o raciocínio é o mesmo e simplesmente estendemos o sistema linear. **Este** é o sistema linear para um polinômio de grau 2. Para um polinômio de **grau 3**, estende-se novamente o sistema linear. Isso se **repete** para qualquer polinômio de ordem m.
9. Esta é **outra forma** de visualizar o sistema linear, através de vetores PHI, em que **seus elementos** são os elementos do vetor X elevados às potências de zero até a potência m, de acordo com o índice de PHI. (2’22”)
10. Criamos uma função para calcular o produto interno que recebe um **vetor a**, um **vetor b** e retorna um inteiro **produto**. Primeiro fazemos uma **validação** que garante que temos dois vetores linha com o mesmo número de elementos. **Depois**, calculamos de fato o produto interno.
11. Criamos uma função para realizar o ajuste de curvas com polinômios de ordem n. A função recebe um **vetor x** e um **vetor y** que representam os dados, e um **número n** que indica o grau do polinômio. A função retorna um **vetor c** com os coeficientes do polinômio e uma variável **erro** com o somatório dos erros calculados. Na validação, garantimos que **x e y** são vetores, que **têm o** mesmo tamanho, e que **n é positivo** e menor que o tamanho de x.
12. Na **linha 28**, criamos o primeiro vetor phi, que é unitário. Na **linha 29**, temos um for que calcula os demais valores **de phi**, conforme o método. Nas linhas **34 e 35**, percorremos toda a matriz dos coeficientes calculando os **produtos internos** dos vetores phi. Nas linhas **7 e 8**, criamos o vetor resposta também realizando produtos internos, conforme o método. E na **linha 45**, atribuímos a c o vetor solução do sistema linear formado, que é calculado pela função eliminacao\_gauss, que já foi explicada no trabalho 3.
13. **Aqui**, calculamos a imagem de cada ponto em x de acordo com a curva que queremos e os coeficientes obtidos. Nas linhas **52** e **53**, plotamos os dados inicias e depois a curva ajustada aos dados. Em **seguida**, calculamos o erro em cada ponto e somamos todos eles. (4’2”)
14. Para um polinômio de **grau 1**, aplicamos a função **ajuste** e obtivemos os coeficientes **a** e **b**, e também o **erro**. O **gráfico** mostra o polinômio ajustado aos dados.
15. Para um polinômio de **grau 2**, **obtivemos os coeficientes** **e o** **erro**. **Aqui** está seu gráfico.
16. Para um polinômio de **grau 3**, **encontramos os coeficientes e o erro**. **Aqui** temos o **gráfico.**
17. Para uma **função** exponencial, fazemos a **linearização** e encontramos uma função **linear** equivalente. Aplicando a função **ajuste**, encontramos **b1** e **a**. Para encontrar **b**, tomamos o **exponencial** de **b1**. Aqui está o **gráfico** da função exponencial.
18. Para uma **função** potência, FAZEMOS um PROCESSO SEMELHANTE e ENCONTRAMOS os COEFICIENTES. Aqui está o **gráfico** da função potência.
19. A **curva** que obteve o menor **erro** foi a do polinômio de **ordem 3**, por isso é a que melhor se ajusta aos dados.
20. Usando a função polyfit do MATLAB, que também usa o Método dos Mínimos Quadrados, encontramos valores semelhantes para o polinômio de grau 1.
21. Também para o polinômio de grau 2.
22. E, da mesma forma, para o polinômio de grau 3, assim como nas funções não lineares.
23. Observando o tempo de processamento de cada função, vemos que polyfit tem um desempenho melhor, mas deve-se levar em consideração que a função ajuste também faz a plotagem do gráfico, logo, era esperado que levasse mais tempo para processar.