UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Relatório Final:

1 - Erros
2 - Equações Não Lineares
3 - Aproximação De Funções
4 - Integração Numérica

Licenciatura em Engenharia Informática

Métodos Computacionais em Engenharia

Prof. João Matias

Autores:

Marcelo Pinto - 60102

Nuno Lopes - 60141

Ricardo Cardoso - 28382

Vila Real, 2016

Índice

Relatório 1 – Erros:

Introdução:

Na primeira fase do trabalho de Métodos Computacionais em Engenharia foi-nos pedida a escolha de uma série numérica bem identificada e caracterizada, convergente para fazermos depois o desenvolvimento em série de Taylor.

A definição da série de Taylor diz-nos que uma função com derivadas de ordem k com k=1,2,3,...,N em algum intervalo que contenha a como um ponto inferior. Desta forma, para qualquer inteiro n de 0 a N, o polinómio de Taylor de Ordem n gerado por f em x = a é o polinómio:

$$P_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

Desenvolvimento:

Desta forma decidimos escolher a função x^2*sin(x). Realizamos várias derivadas consecutivas de forma a descobrirmos um padrão que pudesse depois ser traduzido em código para o cálculo dos termos da série de Taylor após cada iteração.

A função criada vai aceitar três valores. O primeiro é o x que queremos calcular na função x^2*sin(x). O segundo valor é o número de iterações que desejamos que sejam feitas e o terceiro é o erro, que vai ser usado para correr o ciclo em função do erro.

A função a chamar é a função **Relatorio1.m**. A título de exemplo fizemos a chamada a função da seguinte forma: parte1_relatorio(5,25,0.000000025).

O output foi:

Iterações: 13

Erro: 2.500000e-08

Valor aproximado: -2.397311e+01

Valor exato: -2.397311e+01

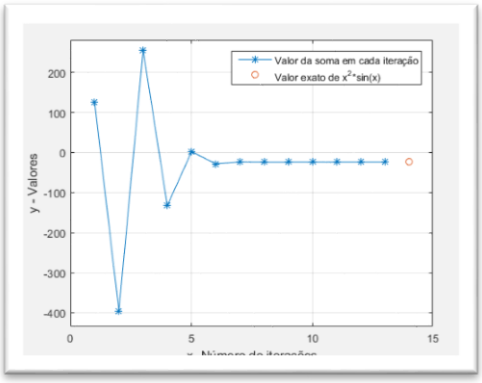

Figura 1 – Exemplo gráfico relatório 1

Como podemos comprovar a função parou devido ao facto de o erro ser menor que os 0.000000025 definidos através do parâmetro de entrada e que apenas precisou de 13 iterações.

Ao compararmos o valor aproximado com o valor exato é possível verificar que são iguais, o que nos leva a constatar que o cálculo da série de Taylor foi bem implementado e que apresenta valores bastante próximos ou iguais aos "exatos".

Relatório 2 – Equações Não Lineares:

Introdução:

Para a realização da segunda parte do trabalho foi-nos pedida a escolha de uma equação não linear com pelo menos uma solução real para a qual devemos então descobrir as raízes existentes.

Desenvolvimento:

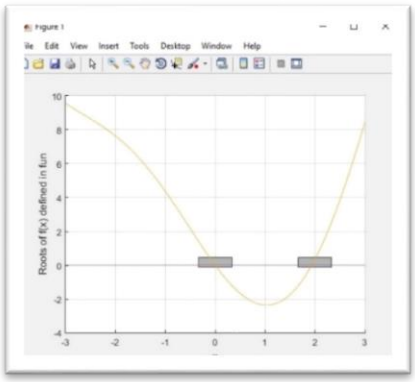

Figura 2 - Raizes de f(x) usando brackPlot

Escolhemos a função f(x) = x^2-4sen(x) e para uma primeira estimativa das raízes da função foi utilizado o método gráfico (função brackPlot, figura 1) no qual foi possível observar duas raízes nos intervalos [-0.33 , 0.33] e [1.67 , 2.33].

Após o descrito anteriormente, desenvolvemos uma função em Matlab à qual denominamos **Relatorio2.m** que irá determinar os zeros da função usando o Método de Newton estudado nas aulas.

O Método de Newton tem como objetivo estimar as raízes de uma função, para isso, escolhe-se uma aproximação inicial para esta, após isso, calcula-se a equação da reta tangente (derivada) da função nesse ponto e a interseção dela com o eixo das abcissas, a fim de encontrar uma melhor aproximação para a raiz. Repetindo-se o processo, cria-se um método iterativo para encontramos a raiz da função.

A função requer como parâmetros de entrada um valor inicial (xi), uma tolerância ou margem de erro (tol) e um número máximo de iterações (nmax). Caso não sejam inseridos os dois últimos parâmetros, a função toma valores por defeito.

No final será apresentado ao utilizador uma tabela com os valores obtidos das sucessivas iterações da função usando o método de Newton, um gráfico representando esta evolução e ainda o resultado obtido usando o Método de Newton por nós implementado, seguido do valor usando a função fzero do Matlab.

Utilizando o comando *format long* e de seguida o comando Relatorio2(3), podemos comparar o valor obtido através do Metodo de Newton por nós implementado com o valor obtido pela função fzero do Matlab, podemos concluir que os valores são muito similares e que provavelmente obtivemos uma boa aproximação.

Relatório 3 – Aproximação De Funções:

Introdução:

No relatório 3 foi-nos pedido a escolha de um conjunto de pontos e posterior cálculo e exibição gráfica dos polinómios interpolador, interpolador segmentado e aproximação pelo método dos mínimos quadrados. Decidimos escolher um assunto original e interessante para a escolha dos pontos: A relação entre o valor de compra de uma determinada quantia em euros ao longo dos últimos 55 anos. Para cada um dos 3 pontos pedidos, está feita uma função, cada uma delas respondendo claramente ao que nos foi pedido, fazendo o melhor possível, tendo em conta aperfeiçoamentos e originalidade.

Desenvolvimento:

Escolhemos 12 pontos, extraídos do *site* da PORDATA e fizemos 3 funções onde utilizamos 3 métodos para obtenção de polinómios que relacionem esses pontos, em cada uma das funções entra um parâmetro *valor* que é o valor atual em euros para o qual o utilizador quer saber a evolução ao longo dos últimos anos. Descrevemos de seguida os diferentes métodos, e o relatório em geral.

A função **Relatorio3Interpolador.m** calcula e exibe o polinómio interpolador de Lagrange. O polinómio de Lagrange é o polinómio de interpolação de um conjunto de pontos na forma de Lagrange. Dado um conjunto de k+1 pontos:

$$(x_0, y_0), \ldots, (x_k, y_k)$$

com todos xj distintos, o polinômio de interpolação de um conjunto de pontos na forma de Lagrange é a combinação linear dos polinômios na base de Lagrange:

$$L(x) := \sum_{j=0}^{k} y_j l_j(x)$$ onde lj(x) são os polinômios na base de Lagrange.

A função **Relatorio3InterpoladorSegmentado.m** calcula e exibe o polinómio interpolador segmentado linear e interpolador segmentado com splines no mesmo gráfico para se fazer uma comparação (splines – azul / linear - verde).

O polinómio interpolador segmentado **linear** é o método de interpolação segmentada que se utiliza a partir de uma função linear p(x) para representar, por aproximação, uma suposta função f(x) que originalmente representaria as imagens de um intervalo descontínuo contido no domínio de f(x). Acontece uma alteração brusca nas junções das funções lineares.

O polinómio interpolador segmentado com **splines** permite uma atenuação e uma passagem discreta nos pontos, a função derivada é continua, sendo esta a principal diferença para o método de interpolação segmentada linear.

A função **Relatorio3MinimosQuadrados.m** calcula e exibe o polinómio originado com o método dos mínimos quadrados. Este método é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados. Nesta função, entra também o grau do polinómio de ajuste dos dados, no entanto, se não for introduzido este parâmetro, o grau é colocado em 6, porque testando hipóteses, foi o polinómio que se ajustou melhor aos dados.

Em qualquer um destes métodos, a tendência do polinómio é a diminuição do poder de compra ao longo dos anos para a mesma quantia de euros, assim como a tendência dos pontos.

No entanto, verificamos que o método dos mínimos quadrados era o que melhor descrevia a relação entre os pontos. Uma grande vantagem deste método é sua capacidade de tratar pequenos conjuntos de dados, além disso, os pesos enfatizam mais as estimativas de semi-variâncias para um grande número de pares e também quando os valores do semivariograma estão próximos à origem, que é a parte mais crítica do ajuste. O polinómio interpolador e interpolador segmentado não descrevem de forma eficaz a situação em geral, eles apenas passam nos pontos, o que numa situação real pouco ou nada importa, o que queremos é achar um polinómio que descreva um conjunto de pontos, na sua totalidade e preveja onde possíveis futuros pontos iram provavelmente estar, não exatamente mas perto.

Relatório 4 – Integração Numérica:

Introdução:

Para a realização da quarta etapa do trabalho tivemos de escolher uma função não polinomial, f(x) = exp(sin(x)), e calcular a área entre o gráfico da função e o eixo dos XX's no intervalo [-1;2]. De modo a representar graficamente a área pretendida usamos o comando area do Matlab.

Desenvolvimento:

Para a realização desta etapa do trabalho e para uma melhor organização do código decidimos criar 3 ficheiros. O ficheiro principal **Relatorio4.m** e os ficheiros auxiliares trapezio.m e simpson.m que irão calcular o valor da área nos intervalos pretendido usando a regra dos trapézios e a Regra de Simpson, respetivamente.

A regra dos trapézios é um método numérico que permite calcular uma aproximação do valor do integral de uma função entre dois pontos através do cálculo da área do trapézio formado ao traçarmos uma reta entre os dois pontos e o eixo dos XX's. No entanto é obtido uma aproximação grosseira com erro relativamente grande (depende da função). Uma forma de resolver este problema é subdividir o intervalo em vários subintervalos e aplicar sucessivamente a regra dos trapézios a cada subintervalo, melhorando assim a aproximação, de uma forma iterativa, sendo esse o método por nós utilizado.

Outra forma de obter uma aproximação do valor do integral entre dois pontos é através da aplicação da regra de Simpson. A regra de Simpson baseia-se em aproximar o valor do integral entre dois pontos através de um polinómio interpolador de grau 2. Para tal precisamos de 3 pontos (2 subintervalos). No entanto, ao aproximarmos o valor da função por um polinómio de grau 2, como na regra anterior, também poderemos encontrar um erro grande. Então, como na regra anterior, a solução passa por aumentar o número de subintervalos diminuindo o erro e consequentemente melhorando a aproximação. Esta regra pode também ser aplicada de uma forma iterativa.

Após aplicação dos dois métodos à nossa função no intervalo pretendido chegamos à conclusão que nos dois métodos o erro diminui quando aumentamos o número de intervalos, conforme era espectável, e que obtemos um valor muito próximo ao obtido através da função integral do Matlab, no entanto, ao contrário do esperado, obtemos aproximações mais próximas a esta função com a regra dos trapézios, o que nos leva a pensar que a função integral do Matlab é baseada na regra dos trapézios.

No entanto, pelos conhecimentos adquiridos nas aulas de MCE, a regra de Simpson costuma ser mais eficaz, pelo facto de um polinómio de segundo grau por fazer uma curva no intervalo, usualmente estima melhor do que a regra de trapézios para um número de intervalos pequeno a médio. Para um intervalo de pontos muito grande, a regra dos trapézios usualmente é melhor, pelo facto que quando há muitos pontos, o ideal é fazer uma reta entre eles em vez de uma curva. A regra dos trapézios também é mais eficaz quando a função é uma função de primeiro grau ou quando por exemplo a função é uma exponencial e o intervalo é bastante negativo, onde a melhor estimativa não é claramente feita por curvas mas sim por retas.