Algoritmos Numéricos DI/CT/UFES

Roteiro para a Aula de Laboratório sobre Interpolação Polinomial

Parte A

Faça, inicialmente (na "mão") os exercícios abaixo para entender os conceitos. Começe entendeno o problema e os conceitos e veja, se quiser, as notas de aulas.

1. Dados os pontos da função na forma tabelar abaixo:

x_k	0.5	0.8	1.0
$y_k = f(x_k)$	2.4	1.5	1.7

(a) Obter o polinômio interpolador , de grau 2, escrito na forma $p_2(x) = a_0 + a_1x + a_2x^2$. Escreva as restrições de interpolação que devem ser satisfeitas, monte o sistema linear $Xa = y$ correspondente e obtenha o polinômio interpolador. Resolva o sistema linear da forma que achar melhor.
Dica: use, se quiser, o comando do octave:

```
>> X\y
```

para resolver o sistema linear obtido com as restrições. Esse comando resolve um sistema linear via eliminação de Gauss com pivoteamento.

2. Dados os pontos da função na forma tabelar abaixo (são os mesmo do exercício anterior): Calcule as diferenças divididas ascendentes de ordem 1 até 2 e obtenha o polinômio interpolador

x_k	0.5	0.8	1.0
$y_k = f(x_k)$	2.4	1.5	1.7

de grau 2, via forma de Newton.

Parte B

Usando o código DifDivididasAsc.m fornecido (ou uma implementação equivalente de sua preferência) resolva os exercícios abaixos. Observe que a função implementada tem como entrada dois vetores: o vetor x e o vetor y, que correspondem ao dados.

3. Calcule as diferenças divididas ascendentes de ordem 1 até 2, dos pontos da tabelas abaixo: Para estes pontos, por exemplo, a chamada seria:

x_k	1.0	1.5	3.0
$y_k = f(x_k)$	2.6	3.4	7.6

```
>> x=[1.0  1.5   3.0]
>> y=[2.6  3.4   7.6]
>> [DD, vetB] = DifDivididasAsc(x,y)
```

Obs: se quiser visualizar os pontos, em par de eixos cartesianos, faça

```
>> x=[1.0  1.5   3.0]
>> y=[2.6  3.4   7.6]
>> plot (x, y, 'b*')
```

4. Implemente, agora, um código que dado um conjunto de pontos $P = ((x_0, y_0), (x_0, y_0), ... (x_n, y_n))$ e um ponto z em $D = [x_0, x_n]$, avalie o polinomio interpolador deste pontos, neste ponto z. Dica: pode fazer a implementação de várias formas:
(a) usando como dados de entrada a matriz DD (calculada pelo DifDivididasAsc(x,y)) e o ponto z e fazendo a "ativação" das duas funções na sequencia. Por exemplo, se o nome do código implementado fosse avaliaPolin, para o conjunto de pontos do exercício anterior a chamada seria:

```
>> x = [1.0  1.5   3.0]
>> y =[ 2.6  3.4   7.6]
>> [DD, vetB] = DifDivididasAsc(x,y)
>> pz = avaliaPolin(DD, z)
```

(b) Os dados de entrada sendo os vetores x e y e gerando a matriz (ou um vetor) das diferenças divididas e fazer todos os cálculos em um único código. Por exemplo, para um conjunto de pontos do exercício anterior a chamada seria:

```
>> x =[1.0  1.5   3.0]
>> y =[2.6  3.4   7.6]
>> pz = avaliapol(x, y , z)
```

5. Funções conhecidas como funções de Bessel (há diversos tipos) são muito empregadas em vários ramos da engenharia. Avaliação destas funções em um ponto x é muito custosa pois envolve uma série. Assim, é comum encontrar estas funções apresentadas na forma tabelar. A tabela abaixo apresenta os valores (exibindo apenas 3 casas decimais) de uma função de Bessel específica (a de 1^a espécie e de índice 0, denotada comumente por $J_o(x)$), em $D = [1.8, 2.4]$:

x_i	1.8	2.0	2.2	2.4
$y_i = J_o(x_i)$	0.582	0.578	0.556	0.520

Usando os códigos implementados, calcule:
(a) todas as diferenças divididas ascendentes que podem ser obtidas com os dados fornecidos.
(b) obtenha o polinômio interpolador da função de Bessel em $D = [1.8, 2.4.]$ usando todos os pontos fornecidos na tabela, via interpolação de Newton e calcule $f(z = 2.1)$ pelo polinômio obtido.