UNIVERSIDADE FEDERAL DE OURO PRETO
Instituto de Ciências Exatas e Biológicas
Departamento de Computação

José Álvaro Tadeu Ferreira

Cálculo Numérico – Roteiros de aulas

O uso do Scilab

Ouro Preto
Outubro de 2009

Aula - 1

1 – Introdução

O Scilab é um ambiente voltado para o desenvolvimento de software para a resolução de problemas numéricos.

→ Criado em 1990: pesquisadores do INRIA (**I**nformática e **A**utomação).

→ Distribuído na internet como freesoftware desde 1994 (**www.scilab.org**).

→ A partir de 2003 passou a ser mantido pelo **Consórcio Scilab**.

→ Similar ao **MatLab**.

→ Dispõe, em seu ambiente, de ferramentas para cálculo numérico, programação e elaboração de gráficos 2D e 3D.

→ O ***prompt*** do Scilab: --> (está aguardando uma linha de comando).

→ Todo comando deve ser seguido de ***enter*** para que seja executado.

→ Para obter ajuda: ***Help***.

→ Para obter ajuda sobre um objeto específico: ***Help nome do objeto***.

2 – Operações básicas

→ Pode ser usado como uma calculadora.

```
--> 2 + 3
 ans =
          5
```

→ É mais conveniente o uso de variáveis.

```
--> a = 2 + 3
 a =
       5
```

Falar sobre o uso do ponto-e-vírgula.

→ Operadores aritméticos: +, -, *, /, ^

3 – Variáveis

→ Diferencia maiúsculo de minúsculo.

→ Padrão minúsculo.

→ O nome deve começar com uma letra.

→ Algumas variáveis (**?**) pré-definidas: %e, %pi, %i, %eps, %inf.

4 – Funções

→ Exemplos de algumas funções pré-definidas: abs(x), sin(x), cos(x), tan(x), cotg(x), exp(x), log(x), log2(x), log10(x).

5 – Vetores

→ Criando um vetor: sintaxes.

Exemplo: seja cria o vetor v = [1 2 3 4] nas formas **linha** e **coluna**.

6 – Matrizes

→ Criando uma matriz: sintaxes.

Exemplo – 1.1

Seja criar a matriz $m = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 1 & 3 & 2 \end{bmatrix}$.

→ Algumas operações básicas: det(m), inv(m), m'.

7 – Resolução de Sistemas de Equações Lineares (a.x = b)

→ Método de **Gauss com pivotação parcial**

Função **x = a\b**.

Exemplo – 1.2

Seja resolver o sistema de equações a.x = b no qual $a = \begin{bmatrix} 1 & 5 & 1 \\ 10 & 2 & 1 \\ 2 & 3 & 10 \end{bmatrix}$ e $b = [-8\ 7\ 6]^t$.

A solução deste sistema de equações é $x = [1\ -2\ 1]^t$.

→ Método da **Decomposição LU com pivotação parcial**

Função **[l u p] = lu(a)**.

Na sequência, resolvem-se os sistemas de equações y = l\(p*b) e x = u\y.

Exemplo – 1.3

Seja o mesmo sistema de equações do exemplo – 1.2.

Aula - 2

1 – Interpolação polinomial

→ A função **interp1(.)**.

Sintaxe: yp = interp1(x, y, xp, “método”)

x e y: abscissas e ordenadas dos pontos da dos.

xp : parâmetro da interpolação.

yp : valor estimado de y para x = xp

método: pode ser “linear” ou “spline”.

Exemplo – 2.1

Os pontos a seguir relacionam a altura, em metros, da água em um tanque, com o tempo, em minutos.

tempo	0	1	2	3	4	5
altura	0	0,7	2,4	3,1	4,2	4,8

Estime a altura da água no instante t = 3,5min.

Solução

-- > t = 0:1:5

-- > y = [0 0.7 2.4 3.1 4.2 4.8]

Usando interpolação linear

-- > y1 = interp1(t, y, 3.5, “linear”)

y1 = 3.65

Usando interpolação spline

-- > y1 = interp1(t, y, 3.5, “spline”)

y1 = 3.60375

2 – Definição de funções

Sintaxe

deff(“vd = nf(vi)”, “vd = fa”)

Exemplo – 2.2

Seja definir a função $y = f(x) = 25 - 4.x - 10.e^{-0,4.x}$.

3 – Integração numérica

3.1 – A forma analítica da função é conhecida

3.1.1 – A função **integrate(.)**

Sintaxe

valor = integrate("fa", "variável", li, ls, [ea, er])

Não é necessário definir a função previamente.

ea e er (erros limites) são parâmetros opcionais, os valores *default* são 10^{-8} e 10^{-14}, respectivamente.

Exemplo – 2.3

Sendo $f(x) = -0{,}05.\ln(0{,}4076).e^{-0{,}05.x}$, estime $I = \int_0^{50} f(x).dx$.

```
-- > v = integrate("-0.05*log(0.4076)*exp(-0.05*x)", "x", 0, 50)
     v = 0.8238002
```

3.1.2 – A função **intg(.)**

Sintaxe

[valor erro] = intg(li, ls, nf, [ea, er])

erro: valor estimado do erro absoluto no resultado

ea e er (erros requeridos no resultado) são parâmetros opcionais, têm como valores *default* 10^{-14} e 10^{-8}, respectivamente.

É necessário definir a função previamente.

Exemplo – 2.4

Seja a mesma integral do exemplo – 2.3.

```
-- > deff("y = f(x)", "y = - 0.05*log(0.4076)*exp(-0.05*x)")
-- > [v ea] = intg(0, 50, f)
     ea = 9.146D-15
     v = 0.8238002
```

3.2 – A forma analítica da função não é conhecida

3.2.1 – A função **inttrap(.)**

<u>Sintaxe</u>

inttrap(x, y)

Exemplo – 2.5

Sendo f(x) = ln(x + 2) -1, estime $I = \int_{2}^{3,2} f(x).dx$ com h = 0,24.

-- > deff("y = f(x)", "y = log(x + 2) - 1")

-- > x = 2:0.24:3.2

-- > y = f(x);

-- > i = inttrap(x, y)

i = 0.6275706

3.2.2 – A função **intsplin(.)**

Esta função produz um resultado melhor que inttrap(.).

<u>Sintaxe</u>

intsplin(x, y)

Exemplo – 2.6

Seja a mesma integral do exemplo – 2.5.

-- > i = intsplin(x, y)

i = 0.6278471

Aula - 3

Resolução de Equações Não Lineares

1 – A função fsolve(.)

Sintaxe

[raiz vf] = fsolve(x0, nomefunc, [prec])

x0: estimativa inicial para a raiz

nf: nome da função (**deve ser definida previamente**).

prec: precisão desejada (opcional, o valor *default* é D-10).

raiz: valor obtido para a raiz.

vf: valor da função no ponto raiz.

Exemplo 3.1

Seja determinar o zero da função $f(x) = e^{-x^2} - \cos(x)$ situado no intervalo (1, 2).

```
--> deff("y = f(x)","y = exp(-x^2)-cos(x)")
--> [raiz v] = fsolve(1,f)
     v  = - 1.388D-17
      x  = 1.4474143
```

2 – Polinômios

2.1 – Definição de polinômios

É utilizada a função **poly(.)**.

Sintaxe

nomepol = poly(v, “var”, “opção”)

v: vetor dos coeficientes (do grau menor para o maior) ou das raízes.

var: variável do polinômio.

opção: se v for o vetor dos coeficientes, então opção = c, se for o vetor das raízes, então opção = r.

Exemplo 3.2

Seja definir o polinômio $p(x) = x^5 – 2.x^4 – 7.x^3 + 9.x^2 + 8.x – 6$ por meio dos seus coeficientes.

```
--> p = poly([- 6 8 9 – 7 – 2 1], “x”, “c”)
```

Exemplo 3.3

Seja definir o polinômio cujas raízes são – 2, 1 e 2.

--> p1 = poly([– 2 1 2], “x”, “r”)

p1 = $x^3 - x^2 - 4.x + 4$

2.2 – Valor numérico de um polinômio

É utilizada a função horner(.).

<u>Sintaxe</u>

horner(nomepol, ponto)

nomepol: nome do polinômio (definido previamente)

ponto: ponto de avaliação (pode ser um escalar ou um vetor).

Exemplo 3.4

Seja avaliar o polinômio definido no exemplo 3.2 em x = 5 e z = [0 1 2].

--> val = horner(p, 5)

val = 1259

val1 = horner(p, z)

val1 = [- 6 3 – 10]

2.3 – Operações aritméticas entre polinômios

2.3.1 – Soma

Dados dois polinômios, p1 e p2, basta, simplesmente, fazer p1 + p2.

Exemplo 3.5

Seja somar os polinômios p e p1 dos exemplos 3.2 e 3.3.

--> p2 = p + p1

p2 = $x^5 – 2.x^4 – 6.x^3 + 8.x^2 + 4.x – 2$

2.3.2 – Multiplicação

Dados dois polinômios, p1 e p2, basta, simplesmente, fazer p1 * p2.

Exemplo 3.6

Seja multiplicar os polinômios p e p1 dos exemplos 3.2 e 3.3.

--> p3 = p * p1

p3 = $x^8 – 3.x^7 – 9.x^6 + 28.x^5 + 19.x^4 – 78.x^3 + 10.x^2 + 56.x – 24$

2.3.3 – Divisão

É utilizada a função **pdiv(.)**.

Sintaxe

[r q] = pdiv(nomepol_1, nomepol_2)

r: resto

q: quociente

nomepol_1: nome do polinômio 1 (o dividendo, definido previamente)

nomepol_2: nome do polinômio 2 (o divisor, definido previamente)

Exemplo 3.7

Seja dividir o polinômio p por p1 dos exemplos 3.2 e 3.3.

--> [r q] = pdiv(p, p1)

$q = x^2 - x – 4$

$r = – 3.x^2 – 4.x + 10$

2.4 – Derivada de um polinômio

É utilizada a função **derivat(.)**.

Sintaxe

derivat(nomepol)

nomepol: nome do polinômio (definido previamente).

Exemplo 3.8

Seja derivar o polinômio p do exemplo 3.2.

--> dp = derivat(p)

$dp = 5.x^4 – 8.x^3 – 21.x^2 + 18.x + 8$

Exemplo 3.9

Seja construir a sequência de Sturm associada a $p(x) = x^5 – 2.x^4 – 7.x^3 + 9.x^2 + 8.x – 6$.

O primeiro termo é p(x).

O segundo termo é a primeira derivada de p(x): $p1 = 5.x^4 – 8.x^3 – 21.x^2 + 18.x + 8$

Daí em diante, cada termo é o resto, com o sinal trocado, da divisão dos dois termos anteriores.

-->[r q]=pdiv(p,p1)

$q = - 0.08 + 0.2x$

$r = - 5.36 + 7.84x + 3.72x^2 - 3.44x^3$

-->p2 = -r

$p2 = 5.36 - 7.84x - 3.72x^2 + 3.44x^3$

Quarto termo:
-->[r q]=pdiv(p1,p2)

q = - 0.7537858 + 1.4534884x
r = 12.040292 + 4.2996214x - 12.408734x^2

-->p3 = -r
p3 = - 12.040292 - 4.2996214x + 12.408734x^2

Quinto termo

-->[r q]=pdiv(p2,p3)

q = - 0.2037308 + 0.2772241x

r = 2.9070217 - 5.3781064x

-->p4 = -r

p4 = - 2.9070217 + 5.3781064x

Sexto termo

-->[r q]=pdiv(p3,p4)

q = 0.4476775 + 2.3072683x

r = - 10.738884

-->p5 = -r

p5 = 10.738884

2.5 – Zeros de um polinômio

Usa a função **roots(.)**. O polinômio pode ter coeficientes reais ou complexos.

Sintaxe

roots(nomepol)

nomepol: nome do polinômio (definido previamente).

Exemplo 3.10

Seja determinar os zeros dos polinômios p e p1 dos exemplos 3.2 e 3.3.

--> rp = roots(p)

Os zeros de p são: -2,138; -1,054; 0,566; 1,507 e 3,119.

--> rp1 = roots(p1)

Os zeros de p1 são: -2; 2 e 1.