Equações diferenciais

Introdução:

Considere um corpo de massa m e feito de um material cujo calor específico à pressão constante seja c_p. Este corpo está inicialmente a uma temperatura T_0, e é imerso rapidamente em um meio a uma temperatura que está T_F °C acima de T_0 (T_F constante). A área de troca de calor é A, e o coeficiente de troca de calor é k. Seja ainda T a diferença entre a temperatura do corpo e a temperatura inicial T_0. Uma variação infinitesimal dT da diferença de temperatura T pode ser calculada por:

$mc_p dT = Ak(T_F - T)dt =$ calor trocado no intervalo de tempo dt.

Reescrevendo a equação:

$$\frac{dT}{dt} = \frac{Ak}{mc_p}(T_F - T)$$

ou, chamando

$T = y$ e $\frac{Ak}{mc_p} = C$

obtemos:

$$\dot{y} = C(T_F - y)$$

Foi estabelecida uma equação diferencial cuja solução é a diferença de temperatura T do corpo (diferença entre a temperatura do corpo e a temperatura T_0). Esta equação é um modelo matemático do aquecimento do corpo. Observe que $T = y$ aumenta até que o corpo entre em equilíbrio térmico, ou seja, até que $T = T_F$. A variável T é a variável de estado e está relacionada com a quantidade de calor acumulada no corpo de massa m.

Algoritmo de integração numérica - Euler:

Considerando a seguinte equação diferencial:

$$\dot{y} = f(y(t), u(t))$$

A solução numérica pelo método de Euler é (h é o passo de integração):

$$y(t+h)=y(t)+h.f(y(t),u(t))$$

No caso da equação $\dot{y} = C(T_F - y)$, obtemos:

$\dot{y} = f(y(t), u(t)) = C(T_F - y)$, e como C é constante e $T_F = u$ também é constante:

$$y(t+h)=y(t)+h.f(y(t))$$

Algoritmo de integração numérica - Runge Kutta:

Considerando a seguinte equação diferencial:

$$\dot{y} = f(y(t), u(t))$$

A solução numérica pelo método de Runge Kutta é:

$k_1=h.f(y(t), u(t))$
$k_2=h.f(y(t)+k_1/2, u(t+h/2))$
$k_3=h.f(y(t)+k_2/2, u(t+h/2))$
$k_4=h.f(y(t)+k_3, u(t+h))$
$y(t+h)=y(t)+(k_1+2.k_2+2.k_3+k_4)/6$

No caso da equação $\dot{y} = C(T_F - y)$, obtemos:

$\dot{y} = f(y(t), u(t)) = C(T_F - y)$, e como $T_F = u =$ constante:

$k_1 = h.f(y(t))$
$k_2 = h.f(y(t)+k_1/2)$
$k_3 = h.f(y(t)+k_2/2)$
$k_4 = h.f(y(t)+k_3)$
$y(t+h) = y(t)+(k_1+2.k_2+2.k_3+k_4)/6$

Exemplo 1:

Arquivo 'numericoE.sce' que calcula a solução da equação diferencial abaixo, usando o método de Euler, e que plota o gráfico de $y(t)$ em função de t, e o gráfico da solução exata $y_e(t)$ em função de t.

Equação diferencial ordinária linear: $\dot{y}(t) = [1 - y(t)]/2$
(é o modelo desenvolvido na Introdução adotando-se $C = 1/2$ e $T_F = 1$)

Instante inicial: $t = 0$ Instante final: $t_f = 10$ Condição inicial: $y(0) = 0$

Solução analítica (exata) da equação diferencial: $y_e(t) = 1 - e^{-t/2}$

Implementação da função 'funcao.sci' (escrever no editor do Scilab, e salvar como funcao.sci):

```
function [ydot]=funcao(y)
ydot=(1-y)/2;
endfunction
```

Implementação do arquivo 'numericoE.sce' (escrever no editor do Scilab, salvar como numericoE.sce e **executar** no Scilab):

```
// Conjunto de comandos para solucao numerica de equacao diferencial dada pela funcao funcao.sci
// Apagando dados anteriores:
clear
// Carregando a equacao diferencial:
// Carregue a função usando o comando Load do Scilab
// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.5;
// Calculo de numero de passos):
n=round(tf/h);
// Integracao numerica usando o metodo de Euler:
        // Comando for:
        for i=1:n
        // Vetor de tempo:
        t(i+1)=t(i)+h;
        // Solucao numerica:
        y(i+1)=y(i)+h* funcao(y(i));
        // Solucao exata:
        ye(i+1)=1-%e^(-t(i+1)/2);
        // Termino do comando for:
        end
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao numerica","Solucao exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure",1);
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao   entre   solucao   numerica   e   solucao   exata","Tempo   t","Solucao","Solucao
numerica","Solucao exata");
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends([T(4),T(5)],[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
```

Observe que a solução exata dada por meio de uma expressão analítica nem sempre existe ou está disponível.

Exemplo 2:

Arquivo ‘numericoR.sce’ que calcula a solução da equação diferencial abaixo, usando o método de Runge Kutta, e que plota o gráfico de $y(t)$ em função de t, e o gráfico da solução exata $y_e(t)$ em função de t.

Equação diferencial ordinária linear: $\dot{y}(t) = [1 - y(t)]/2$
(é o modelo desenvolvido na Introdução adotando-se $C = 1/2$ e $T_F = 1$)

Instante inicial: $t = 0$ Instante final: $t_f = 10$ Condição inicial: $y(0) = 0$

Solução analítica (exata) da equação diferencial: $y_e(t) = 1 - e^{-t/2}$

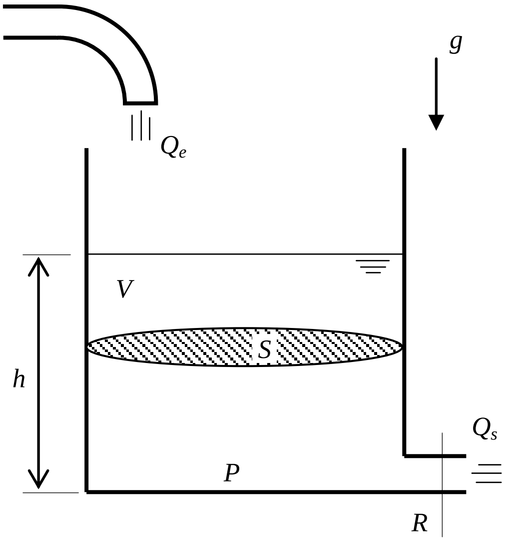
Implementação do arquivo ‘numericoR.sce’ (escrever no editor do Scilab, salvar como numericoR.sce e **executar** no Scilab):

```
// Conjunto de comandos para solucao numerica de equacao diferencial  [1-y(i)]/2
// Apagando dados anteriores:
clear
// Instante inicial:
t(1)=0;
// Instante final:
tf=10;
// Condicao inicial:
y(1)=0;
// Valor inicial da solucao exata:
ye(1)=0;
// Passo de integracao (experimente alterar o passo):
h=0.5;
// Calculo de numero de passos):
n=round((tf-t(1))/h);
// Integracao numerica usando o metodo de Runge Kutta:
        // Comando for:
        for i=1:n
        // Vetor de tempo:
        t(i+1)=t(i)+h;
        // Solucao numerica:
        k1=h*(1-(y(i)))/2;
        k2=h*(1-(y(i)+k1/2))/2;
        k3=h*(1-(y(i)+k2/2))/2;
        k4=h*(1-(y(i)+k3))/2;
        y(i+1)=y(i)+((k1+2*k2+2*k3+k4)/6);
        // Solucao exata:
        ye(i+1)=1-%e^(-t(i+1)/2);
        // Termino do comando for:
        end
// Plotando solucao numerica y versus vetor de tempo t e solucao exata ye versus vetor de tempo t:
plot2d([t,t],[y,ye],[-1 -2]);
// Colocando uma legenda na parte inferior direito da figura (parametro 4):
legends(["Solucao numerica","Solucao exata"],[-1,-2],4)
// Colocando um titulo na figura e nomeando os eixos:
xtitle("Comparacao entre solucao numerica e solucao exata","Tempo t","Solucao")
// Abrindo uma nova janela de graficos:
set("current_figure", 1);
// Aumentando a espessura das linhas:
xset(“thickness”,2)
// Aumentando o tamanho da fonte:
xset(“font size”,4)
// Desenhando outro grafico com linhas diferentes:
plot2d([t,t],[y,ye],[1 2]);
// Usando a variavel do tipo 'lista':
T=list("Comparacao   entre   solucao   numerica   e   solucao   exata","Tempo   t","Solucao","Solucao
numerica","Solucao exata");
// Diminuindo a espessura das linhas:
xset(“thickness”,1)
// Colocando uma legenda na parte superior esquerda da figura (parametro 2):
legends([T(4),T(5)],[1,2],2);
// Colocando um titulo na figura e nomeando os eixos:
xtitle(T(1),T(2),T(3));
// Colocando uma grade no grafico:
xgrid(1)
```

Observe que, neste segundo exemplo, não definimos a função que expressa a equação diferencial (`'funcao.sci'`). A expressão foi colocada diretamente no algoritmo de integração. Não foi necessário, portanto, carregar uma função pelo comando **Load** do Scilab.

Exercício:

Implemente um programa no Scilab que resolva numericamente a equação diferencial que modela o sistema abaixo, tanto pelo método de Euler como Runge Kutta.



Reservatório com água
Parâmetros:
S = 10 m^2 - área da seção transversal (constante)
R = 2×10^8 Pa/(m^3/s)^2 - parâmetro que relaciona vazão com queda de pressão (perda de carga)
ρ = 1000 kg/m^3 - massa específica da água
G = 10 m/s^2 - aceleração da gravidade na superfície da terra

Variáveis:
Q_e = 0,010247 m^3/s - vazão de entrada
h: nível do reservatório [m]
V: volume de água no reservatório [m^3]
P: pressão relativa à atmosférica, no fundo do reservatório [Pa]
Q_s: vazão de saída [m^3/s]

Admite-se que a água seja incompressível.

Pela equação da continuidade:

$$\frac{dV}{dt} = Q_e - Q_s$$

Vamos admitir que a perda de carga na saída é modelada pela expressão:

$$P = RQ_s^2 \quad \Rightarrow \quad Q_s = \sqrt{\frac{P}{R}}$$

Por outro lado, a pressão no fundo do reservatório é:

$$P = \rho g h$$

Volume de água no reservatório:

$$V = Sh \quad \Rightarrow \quad \dot{V} = S\dot{h}$$

Substituindo:

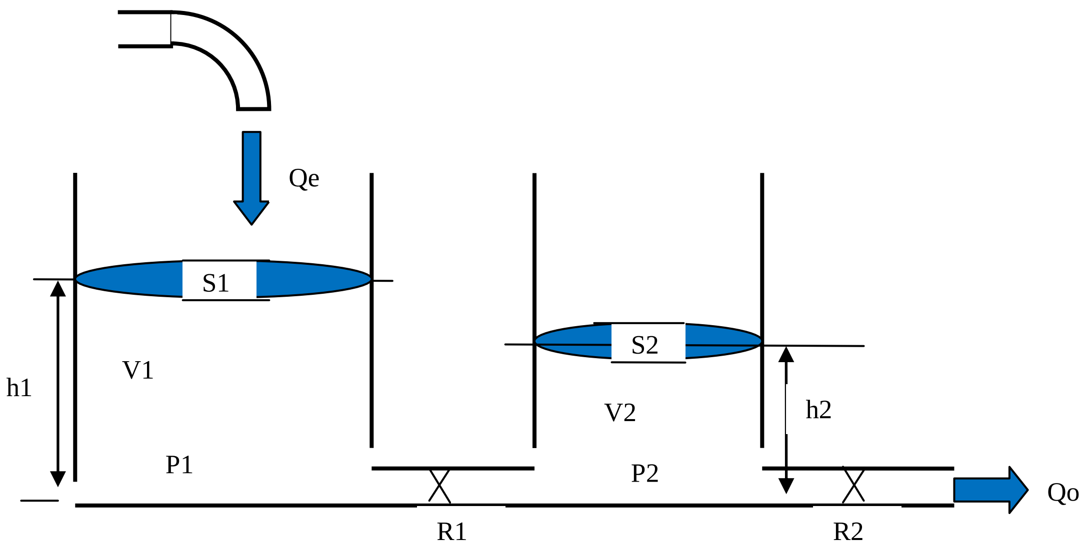
$$S\dot{h} = Q_e - \sqrt{\frac{\rho g h}{R}}$$

Resultando na seguinte equação difererencial ordinária não linear (modelo de 1 reservatório):

$$\dot{h} = \left(-\sqrt{\frac{\rho g h}{R}} + Q_e \right) \frac{1}{S}$$

Considere uma entrada Q_e constante.

Desenvolva um programa em Scilab que resolva numericamente o sistema de equações diferencias que modela o sistema com dois reservatórios, usando tanto Euler como Runge Kutta. Dica: raciocine com vetores.



Modelo do sistema de 2 reservatórios (considere a entrada constante e perdas de carga não lineares como no caso do ex. de 1 tanque).

$$\begin{cases} \dot{h}_1 = \left[Q_e - \sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} \right] \dfrac{1}{S_1} \\ \dot{h}_2 = \left[\sqrt{\dfrac{\rho g}{R_a}(h_1 - h_2)} - \sqrt{\dfrac{\rho g}{R_s} h_2} \right] \dfrac{1}{S_2} \end{cases}$$

3 – Estude a lista C.