APLICACIONES COMPUTACIONALES [15063]
INGENIERÍA DE EJECUCIÓN MECÁNICA

Miércoles, 22 de enero de 2014 (Tiempo: 120 min)

# PRUEBA OPTATIVA DE REEMPLAZO

**Problema 1** (1 pts): Como resultado de un experimento diseñado para medir el coeficiente de roce, se obtienen los datos mostrados en la tabla: Texto

| $x$ (m) | 0 | 5 | 10 | 15 | 20 | 25 | 30 |
|---|---|---|---|---|---|---|---|
| $F(x)$ (N) | 0,0000 | 1,5297 | 9,5120 | 8,7025 | 2,8087 | 1,0881 | 0,3537 |

donde $F(x)$ es la fuerza aplicada y $x$ el desplazamiento en la dirección de la fuerza. Se pide:
(a) Calcular el trabajo $W = \int F(x)\,dx$ con la regla del trapecio usando todos los datos mostrados. (0,7 pts)
(b) Cuantificar el error relativo porcentual si el resultado teórico es 129,52 (N·m). (0,3 pts)

**Solución**:

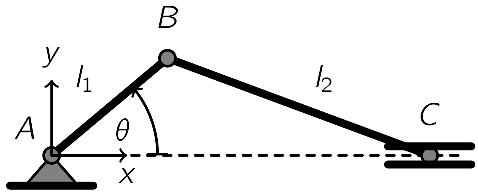
(a) Aplicando la regla del trapecio se obtienen los siguientes valores para cada intervalo:

```
% Metodo del trapecio

x = 0:5:30;
F = [0.0000 1.5297 9.5120 8.7025 2.8087 1.0881 0.3537];

I(1) = (x(2) - x(1))*(F(2) + F(1))/2;
I(2) = (x(3) - x(2))*(F(3) + F(2))/2;
I(3) = (x(4) - x(3))*(F(4) + F(3))/2;
I(4) = (x(5) - x(4))*(F(5) + F(4))/2;
I(5) = (x(6) - x(5))*(F(6) + F(5))/2;
I(6) = (x(7) - x(6))*(F(7) + F(6))/2;

I =
    3.8243   27.6043   45.5363   28.7780    9.7420    3.6045

I_trapz = sum(I)
  119.0893

%% Funcion de Matlab
trapz(x,F)
  119.0893
```

(b) Teniendo en cuenta el valor teórico, se obtiene el error como:

```
% Error relativo al valor teorico en porcentaje
>> error = (129.52 - 119.0893)/129.52 * 100
error =
    8.0534
```

**Problema 2** (1 pts): Para el mecanismo formado por las barras de longitudes $l_1 = 2\,\text{m}$ y $l_2 = 3\,\text{m}$ mostrado en la figura se pide:




(a) Escribir una función en lenguaje Matlab que devuelva un vector con la posición $x$ e $y$ de las uniones $A$, $B$ y $C$, teniendo como dato de entrada el ángulo $\theta$ (0,5 pts).

(b) Escribir un script en lenguaje Matlab que dibuje la posición $x(t)$ del extremo $C$ si se conoce la variación del ángulo con el tiempo, cuyos datos son proporcionados en un archivo de texto en formato matriz (columna 1 [sec]: tiempo, columna 2 [deg]: ángulo). Use en su script la función programada en el punto anterior (0,5 pts).

**Solución**:

(a) Función para obtener las posiciones de $A$, $B$ y $C$.

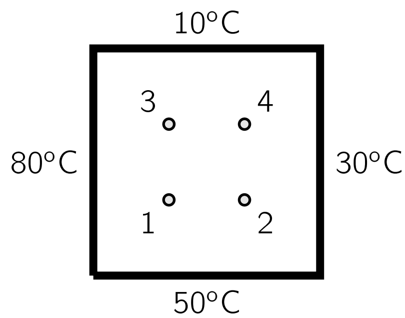
```
% Funcion para obtener las posiciones
function pos = position(theta)

l1 = 2;
l2 = 3;

x(1) = 0;
y(1) = 0;

x(2) = l1 * cos(theta);
y(2) = l1 * sin(theta);

beta = asin(y(2)/l2);
x(3) = x(2) + l2*cos(beta);
y(3) = 0;

pos = [x' ;y'];
%plot(x,y)

end
```

(b) Script para dibujar la posición $x(t)$ del punto $C$.

```
%% Mecanismo

% Se crea un archivo de datos y se guarda en "datos.txt"
%t = 0:0.05:10;
%p = sin(t).*t.^2.*sqrt(t);
%dt = [t' p'];
%save('datos.txt','dt','-ascii')

% Se carga el fichero "datos.txt" y
% se alamcena en el vector 'datos'.
datos = load('datos.txt');
```

```
t = datos(:,1);
a = datos(:,2);

for i=1:length(t)
    pos = position(a(i)/180*pi);
    x(i) = pos(3,1);
end

plot(t,x)
```

**Problema 3** (2 pts): Para una placa de aluminio con las temperaturas aplicadas en el perímetro como muestra la figura. Se pide
(a) Aplicar el método de diferencias finitas centradas para el estado estacionario y establecer el sistema de ecuaciones lineales, cuya solución proporciona las temperaturas en los puntos 1, 2, 3 y 4 al interior de la placa. Considere $\Delta x = \Delta y$. (1,0 pts)
(b) Resuelva el sistema de ecuaciones obtenido usando el método de eliminación Gaussiana o la Regla de Cramer. (1,0 pts)




**Solución**:

(a) Ecuación que se debe aplicar para obtener el sistema de ecuaciones:

$$4T_{i,j} - T_{i+1,j} - T_{i-1,j} - T_{i,j+1} - T_{i,j-1} = 0$$

```
% Matriz A del sistema A x = b
A =
     4    -1    -1     0
    -1     4     0    -1
    -1     0     4    -1
     0    -1    -1     4
```

```
% Vector b del sistema A x = b
b =
    90
    40
   130
    80
```

(a) Solución del sistema lineal usando la regla de Cramer:

```
% Calculamos el determinante de A
det(A)
ans =
   192
```

```
% Reemplazando la primera columna por el vector b se obtiene:
C =
    90    -1    -1     0
    40     4     0    -1
   130     0     4    -1
    80    -1    -1     4
det(C)
ans =
   8400
T3 = 8400/192
T3 =
   43.7500

% Reemplazando la segunda columna por el vector b se obtiene:
C =
     4    90    -1     0
    -1    40     0    -1
    -1   130     4    -1
     0    80    -1     4
det(C)
ans =
   6000
T4 = 6000/192
T4 =
   31.2500

% Reemplazando la tercera columna por el vector b se obtiene:
C =
     4    -1    90     0
    -1     4    40    -1
    -1     0   130    -1
     0    -1    80     4
det(C)
ans =
   10320
T1 = 10320/192
T1 =
   53.7500

% Reemplazando la cuarta columna por el vector b se obtiene:
C =
     4    -1    -1    90
    -1     4     0    40
    -1     0     4   130
     0    -1    -1    80
```

```
det(C)
ans =
   7920
T2 = 7920/192
T2 =
   41.2500
```

```
% Vector solucion x = A\b
x =
   43.7500
   31.2500
   53.7500
   41.2500
```

```
% Distribucion de temperaturas en la placa
T =
   45.0000   10.0000   10.0000   20.0000
   80.0000   43.7500   31.2500   30.0000
   80.0000   53.7500   41.2500   30.0000
   65.0000   50.0000   50.0000   40.0000
```

**Problema 4** (2 pts): El coeficiente de fricción en una tubería de sección circular que transporta agua está dado por la fórmula de Colebrook (ecuación no lineal):

$$\frac{1}{\sqrt{f}} = 1,14 - 2\,\log\left(\frac{e}{D} + \frac{9,35}{R_e\sqrt{f}}\right),$$

donde $f$ es el coeficiente de fricción, $e$ la rugosidad de la tubería, $D$ el diámetro de la sección de la tubería y $R_e$ el número de Reynolds. Se pide:

(a) Utilizar el método de Newton-Raphson para encontrar el coeficiente de fricción, usando los siguientes parámetros: $D = 0{,}1$ m $e = 0{,}0001$ m y $R_e = 5 \times 10^6$. Haga 5 iteraciones e inicie con $f = 0{,}001$. (1,0 pts)

(b) Escribir un programa en Matlab que resuelva el problema no lineal usando el método de Newton-Raphson. (1,0 pts)

**Solución**:

(a) El método de Newton-Raphson nos permite obtener el valor de la fricción de forma iterativa usando la expresión:

$$f_i = f_{i-1} - \frac{g(f)}{g'(f)}$$

A partir de la fórmula de Colebrook se obtienen las siguientes expresiones:

$$g(f) = 2\,\log\left(\frac{9{,}35}{\sqrt{f}\,R_e} + \frac{e}{D}\right) + \frac{1}{\sqrt{f}} - 1{,}14$$

$$s(f) = g'(f) = -\frac{9{,}35}{f^{\frac{3}{2}}\left(\frac{9{,}35}{\sqrt{f}\,R_e} + \frac{e}{D}\right)\,R_e} - \frac{1}{2\,f^{\frac{3}{2}}}$$

Aplicando el método de Newton-Raphson se obtienen los siguientes resultados para las primeras 5 iteraciones:

| | |
|---|---|
| $i = 0$ | 0,001000000000000 |
| $i = 1$ | 0,002057662844640 |
| $i = 2$ | 0,003391427181664 |
| $i = 3$ | 0,004288482976061 |
| $i = 4$ | 0,004496177255745 |
| $i = 5$ | 0,004503962119312 |

(b) La programación del método de Newton-Rapshon para obtener la fricción quedaría como:

```
% Metodo de Newton-Raphson
% Formula de Colebrook

D = 0.1;
e = 0.0001;
R = 5e6;

g =@(x) 1/sqrt(x)-1.14+2*log(e/D+9.35/(R*sqrt(x)));
s =@(x) -9.35/(x^(3/2)*(9.35/(sqrt(x)*R)+e/D)*R)-1/(2*x^(3/2));

x = 0.001;
v = [x];
for i=1:5
    x = x - g(x) / s(x);
    v = [v; x];
end
```

# Formulario

- Regla del trapecio:
$$I = \frac{f(b) + f(a)}{2}(b - a)$$
- Regla de Simpson 1/3:
$$I = \frac{h}{3}\left(f(x_0) + 4f(x_1) + f(x_2)\right)$$
- Formulas de Gauss-Legendre:
$$I = \sum_{i=0}^{n} C_i\, f(z_i), \quad x = \frac{b+a}{2} + \frac{b-a}{2}z \quad \text{(cambio de variable)}$$
- Newton-Raphson:
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

- Diferencias finitas para la conducción del calor en estado transitorio (caso 1D):

$$T_i^{n+1} = T_i^n + \lambda \left( T_{i+1}^n - 2T_i^n + T_{i-1}^n \right), \quad \text{con} \quad \lambda = k \frac{\Delta t}{(\Delta x)^2}$$

- Diferencias finitas para la conducción del calor en estado estacionario (caso 2D):

$$\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} = 0$$

- Regla de Cramer (ejemplo matrix $3 \times 3$, $A\mathbf{x} = b$):

$$x_1 = \frac{1}{\det A} \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}, \quad x_2 = \frac{1}{\det A} \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}, \quad x_3 = \frac{1}{\det A} \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

- Eliminación de Gauss (ejemplo matrix $3 \times 3$, $A\mathbf{x} = b$):

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & \vdots & b_1 \\ a_{21} & a_{22} & a_{23} & \vdots & b_2 \\ a_{31} & a_{32} & a_{33} & \vdots & b_3 \end{vmatrix} \quad \Rightarrow \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} & \vdots & b_1 \\ 0 & a'_{22} & a'_{23} & \vdots & b'_2 \\ 0 & 0 & a''_{33} & \vdots & b''_3 \end{vmatrix}$$

- Algunas derivadas útiles:

$$\frac{d}{dx} (\log x) = \frac{1}{x}$$