

EXAMEN SUSTITUTORIO DE METODOS NUMERICOS (MB536)

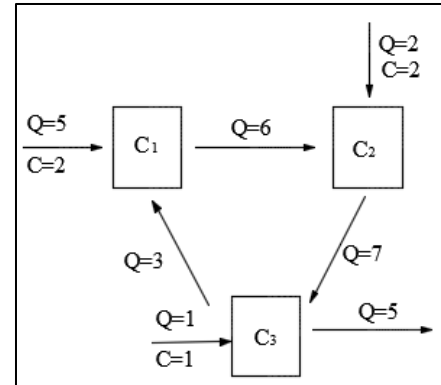
- SOLO SE PERMITE EL USO DE UNA HOJA DE FORMULARIO Y CALCULADORA
- ESCRIBA CLARAMENTE SUS PROCEDIMIENTOS
- PROHIBIDO EL USO DE CELULARES U OTROS EQUIPOS DE COMUNICACION ELECTRONICA
- DURACION: 110 MINUTOS

Problema 1

Aplicando el principio de conservación de masas en un sistema de mezcladoras de fluidos con la configuración mostrada en el esquema, se cumple las siguientes relaciones:

$$\begin{aligned} 6C_1 - 3C_3 &= 10 \\ -6C_1 + 7C_2 &= 4 \\ -7C_2 + 8C_3 &= 1 \end{aligned}$$

Donde Q es el caudal en m^3/min y C es la concentración del fluido en mg/m^3



- (1pto) Determine la convergencia de los métodos iterativos para la solución de este sistema de ecuaciones para hallar las concentraciones.
- (3ptos) Determine las concentraciones usando el método de Jacobi en 3 iteraciones, partiendo las concentraciones desde 0.
- (1pto) Desarrolle un script en matlab para resolver este sistema, usando el método de Gauss-Seidel hasta con un error (norma euclidiana) de 10^{-12} .

Problema 2

A una empresa le cuesta $C(q)$ dólares producir q gramos por día de cierto producto químico, donde

$$C(q) = 1000 + 2q + 3q^{2/3}$$

La empresa puede vender cualquier cantidad de la sustancia química en \$4 un gramo.

- (1pto) Modele la ecuación no lineal que permita encontrar el punto de equilibrio de la empresa, es decir, cuánto debe producir por día con el fin de no tener ni ganancias ni pérdidas.
- (2ptos) Utilice el método de Newton para aproximar el punto de equilibrio. Realice 02 iteraciones. considere $q_0 = 600$.
- (2ptos) Implemente un función en MatLab con la siguiente cabecera

```
function z=newton(fu,dfu,x0,Tol)
%fu: function no lineal
%dfu: derivada de la function no lineal
%x0: valor inicial
%Tol: tolerancia
```

Problema 3

Considere la curva espacial con ecuación paramétrica:

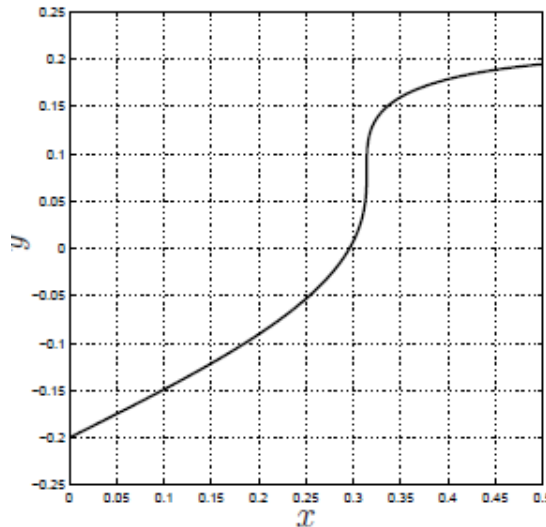
$$x(t) = 0.1 \sin(t) + 0.1t$$

$$y(t) = 0.4 \sin(0.25t) - 0.2$$

Como se muestra en la figura de la derecha.

El vector unitario tangente conocido esta dado por:

$$e_r = \frac{\begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{bmatrix}}{\begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{bmatrix}}$$



Para determinar los vectores tangentes use las fórmulas de diferencias, según lo indicado en a) y b):

- a) **(1.0pto)** El vector tangente en $t = 3$ usando la fórmula de diferencia central con el tamaño de paso $h = 0,5$. Anote el resultado en la figura.
- b) **(1.0pto)** El vector tangente en $t = 1$ usando dos puntos con las diferencias hacia adelante con el tamaño de paso $h = 0,5$. Anote el resultado en la figura.
- c) **(2.0ptos)** Determine los polinomios segmentados de grado 1 usando el polinomio de Lagrange en los tramos de tiempo de $t \in [1 \ 3]$ seg. Usar tres puntos de apoyo. Estos polinomios deben ser dados por tramos. Ayuda: La interpolación se realiza en (t,x) y en (t,y) .
- d) **(1.0pto)** Realice la función en matlab que pueda evaluar las curvas paramétricas dadas en este problema si se conoce el argumento t , con la siguiente cabecera: `function [x,y]=param(t)`. Use los comandos de Matlab para realizar los coeficientes de los polinomios por tramos.

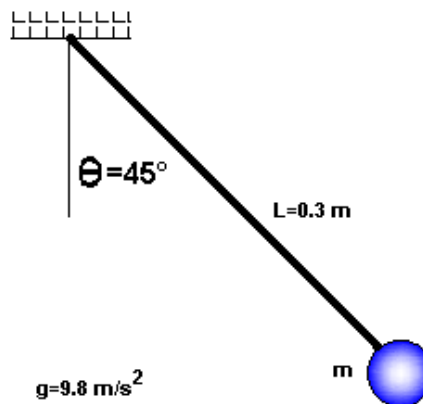
Problema 4

La Ecuación del movimiento de un péndulo simple (partícula de masa “m” suspendida de un hilo liviano de largo “L”) es:

$$\frac{\partial^2 \theta}{\partial t^2} + \frac{g}{L} \text{sen}(\theta) = 0$$

Se sabe que el péndulo de longitud 0.30 m y $g=9.8 \text{ m/s}^2$, se suelta del reposo desde un ángulo de 45° :

- a) **(2.0ptos)** Determine el ángulo y la velocidad angular al cabo de 0.2 segundos, aplique Runge-Kutta 2 con $\Delta T=0.1$ seg.
- b) **(2.0ptos)** ¿Qué tiempo será necesario para que el péndulo pase por $\theta = 0^\circ$, use Euler con $\Delta T=0.1$ seg?
- c) **(1.0pto)** Escriba el código MATLAB para resolver la pregunta a)



SOLUCIONARIO

Problema 1

Solución:

$$a) \begin{bmatrix} 6 & 0 & -3 \\ -6 & 7 & 0 \\ 0 & -7 & 8 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix}$$

Se puede observar la matriz de coeficientes A, que es estrictamente diagonal dominante, por lo tanto converge para cualquiera de los dos métodos iterativos revisados en clase.

b) Preparando las matrices:

$$D = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{bmatrix} \quad L = \begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 7 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{bmatrix}^{-1} * \left(\begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 7 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & 0 & 1/2 \\ 6/7 & 0 & 0 \\ 0 & 7/8 & 0 \end{bmatrix}$$

$$c = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 8 \end{bmatrix}^{-1} \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 10/6 \\ 4/7 \\ 1/8 \end{bmatrix}$$

Primera iteración:

$$X = \begin{bmatrix} 0 & 0 & 1/2 \\ 6/7 & 0 & 0 \\ 0 & 7/8 & 0 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 10 \\ 6 \\ 4 \\ 7 \\ 1 \\ 8 \end{bmatrix} = \begin{bmatrix} 10 \\ 6 \\ 4 \\ 7 \\ 1 \\ 8 \end{bmatrix} = \begin{bmatrix} 1.6667 \\ 0.5714 \\ 0.1250 \end{bmatrix}$$

Segunda iteración:

$$X = \begin{bmatrix} 0 & 0 & 1/2 \\ 6/7 & 0 & 0 \\ 0 & 7/8 & 0 \end{bmatrix} * \begin{bmatrix} 1.6667 \\ 0.5714 \\ 0.1250 \end{bmatrix} + \begin{bmatrix} 10/6 \\ 4/7 \\ 1/8 \end{bmatrix} = \begin{bmatrix} 1.7292 \\ 2 \\ 0.6250 \end{bmatrix}$$

Tercera iteración:

$$X = \begin{bmatrix} 0 & 0 & 1/2 \\ 6/7 & 0 & 0 \\ 0 & 7/8 & 0 \end{bmatrix} * \begin{bmatrix} 1.7292 \\ 2 \\ 0.6250 \end{bmatrix} + \begin{bmatrix} 10/6 \\ 4/7 \\ 1/8 \end{bmatrix} = \begin{bmatrix} 1.9792 \\ 2.0536 \\ 1.8750 \end{bmatrix}$$

c)

```
A=[ 6    0  -3
   -6   7   0
    0  -7   8]
b=[10 4 1]
D=diag(diag(A)); L=-tril(A,-1); U=-triu(A,1);
T=inv(D-L)*U; c=inv(D-L)*b;
x=[0 0 0]';
xa=x;
for i=1:100
    x=T*x+c;
    er=norm(x-xa,2);
```

```

    xa=x;
    if er<1e-12
        break
    end
end
disp(x)
    
```

Problema 2

a) Ingreso = Costo

$$4q = 1000 + 2q + 3q^{2/3}$$

$$f(q) = 2q - 3q^{2/3} - 1000 = 0$$

b) .

$$q_{n+1} = q_n - \frac{2q_n - 3q_n^{2/3} - 1000}{2 - 2q_n^{-1/3}}, \quad q_0 = 600$$

$$q_1 = q_0 - \frac{2q_0 - 3q_0^{2/3} - 1000}{2 - 2q_0^{-1/3}} = 607.6089$$

$$q_2 = 607.6080$$

c) .

```

function z=newton(fu,dfu,x0,Tol)
z=[x0' 1]
error=1
while Tol<error
    x1=x0-inv(feval(dfu,x0))*feval(fu,x0);
    error=norm(-inv(feval(dfu,x0))*feval(fu,x0),inf)
    z=[z;x1' error];
    x0=x1;
end
    
```

Problema 3

Solución

a) Diferencias centrales en los puntos $t = 2,5 \ 3,0 \ 3,5$ (3 puntos)

$$x(2,5) = 0,1 \sin(2,5) + 0,1 \cdot 2,5 = 0,3098$$

$$x(3,5) = 0,1 \sin(3,5) + 0,1 \cdot 3,5 = 0,3149$$

$$y(2,5) = 0,4 \sin(0,25 \cdot 2,5) - 0,2 = 0,0340$$

$$y(3,5) = 0,4 \sin(0,25 \cdot 3,5) - 0,2 = 0,1070$$

$$\partial x(3) = \frac{x(3,5) - x(2,5)}{2 \cdot 0,5} = 0,0051$$

$$\partial y(3) = \frac{y(3,5) - y(2,5)}{1} = 0,073$$

$$e_T = \begin{bmatrix} 0,0694 \\ 0,9976 \end{bmatrix}$$

Graficar en :

$$x(3) = 0,3141$$

$$y(3) = 0,0727$$

b) Diferencias hacia adelante (2 puntos)

t= 1 1.5

$$x(1) = 0,1841$$

$$x(1,5) = 0,2497$$

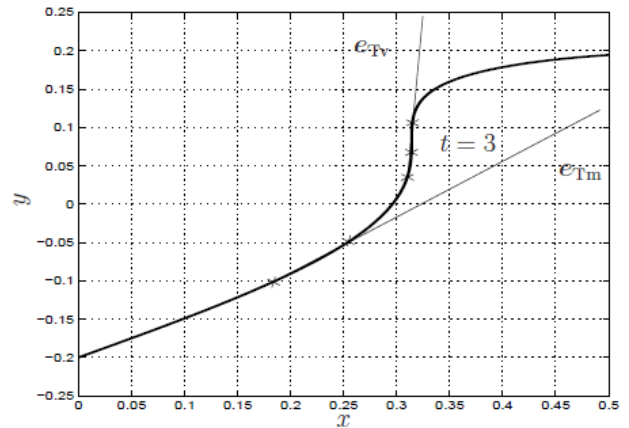
$$y(1) = -0,101$$

$$y(1,5) = -0,0535$$

$$\delta_+x(1) = \frac{x(1,5) - x(1)}{0,5} = 0,1312$$

$$\delta_+y(1) = \frac{y(1,5) - y(1)}{0,5} = 0,0951$$

$$e_T = \begin{bmatrix} 0,8097 \\ 0,5869 \end{bmatrix}$$



c)

Puntos base

$$t_0 = 1; \quad t_1 = 2; \quad t_2 = 3$$

Polinomio de Lagrange

$$P_1(t) = f_n \frac{t - t_{n+1}}{t_n - t_{n+1}} + f_{n+1} \frac{t - t_n}{t_{n+1} - t_n}$$

Puntos para formar los polinomios segmentados (data)

$$x(1) = 0,1841 \quad x(2) = 0,2909 \quad x(3) = 0,3141$$

$$y(1) = -0,1010 \quad y(2) = -0,0082 \quad y(3) = 0,0727$$

$$P_x = 0,1841 \cdot \frac{t-2}{1-2} + 0,2909 \cdot \frac{t-1}{2-1}$$

$$= -0,1841 \cdot (t-2) + 0,2909 \cdot (t-1) \quad \text{para } t \in [1; 2]$$

$$P_y = -0,1010 \cdot \frac{t-2}{1-2} + (-0,0082) \cdot \frac{t-1}{2-1}$$

$$= 0,1010 \cdot (t-2) - 0,0082 \cdot (t-1) \quad \text{para } t \in [1; 2]$$

$$P_x = 0,2909 \cdot \frac{t-3}{2-3} + 0,3141 \cdot \frac{t-2}{3-2}$$

$$= -0,2909 \cdot (t-3) + 0,3141 \cdot (t-2) \quad \text{para } t \in [2; 3]$$

$$P_y = 0,0082 \cdot (t-3) + 0,0727 \cdot (t-2) \quad \text{para } t \in [2; 3]$$

d)

```
function [x,y]=param(t)
    x=0.1*sin(t)+0.1*t;
    y=0.4*sin(0.25t)-0.2;
end
```

```
t1=1:2; [x1,y1]=param(t1);
t2=2:3; [x2,y2]=param(t2);
cx1=polyfit(t1,x1,1);
cy1=polyfit(t1,y2,1);
cx2=polyfit(t2,x2,1);
cy2=polyfit(t2,y2,1);
```

Problema 4

a)

Reduciendo a un sistema de EDO de primer orden:

$$\frac{\partial \theta}{\partial t} = \omega \quad \theta(0) = \pi/4$$

$$\frac{\partial \omega}{\partial t} = -\frac{g}{L} \text{sen}(\theta) \quad \omega(0) = 0$$

Algoritmo de Runge-Kutta 2

$$t_0 = 0$$

$$\theta_0 = \pi/4$$

$$\omega_0 = 0$$

$$h = 0.1$$

$$t_{n+1} = t_n + h$$

$$k_1 = h * \omega_n$$

$$l_1 = h * -\frac{g}{L} \text{sen}(\theta_n)$$

$$k_2 = h * (\omega_n + l_1)$$

$$l_2 = h * -\frac{g}{L} \text{sen}(\theta_n + k_1)$$

$$\theta_{n+1} = \theta_n + 0.5 * (k_1 + k_2)$$

$$\omega_{n+1} = \omega_n + 0.5 * (l_1 + l_2)$$

Evaluando:

tn	Θ_n	Ω_n	K1	L1	K2	L2
0	0.7854(45°)	0	0	-2.3099	-0.2310	-2.3099
0.1	0.6699(38.4°)	-2.3099	-0.2310	-2.0283	-0.4338	-1.3882
0.2	0.3375(19.3°)	-4.0181				

Algoritmo de Euler:

$$t_0 = 0$$

$$\theta_0 = \pi/4$$

$$\omega_0 = 0$$

$$h = 0.1$$

$$t_{n+1} = t_n + h$$

$$\theta_{n+1} = \theta_n + h * \omega_n$$

$$\omega_{n+1} = \omega_n + h * -\frac{g}{L} \text{sen}(\theta_n)$$

Evaluando:

t_n	Θ_n	ω_n
0	0.7854 (45°)	0
0.1	0.7854 (45°)	-2.3099
0.2	0.5544 (31.8°)	-4.6198
0.3	0.0924 (5.3°)	-6.3395
0.4	-0.5415 (-31°)	-6.6410

Tiempo requerido para que pase por $\theta = 0^\circ$: interpolando linealmente con los 2 últimos puntos, se tiene $t=0.3146$ seg

c)

Código ***MATLAB***

```
% theta' = omega
% omega' = -g/L*sin(theta)
clc, clear all, g=9.8, L=0.30
t(1)=0, theta(1)=pi/4, omega(1)=0, h=0.1
for i=1:2
    t(i+1)=t(i)+h, k1=h*omega(i), l1=h*-g/L*sin(theta(i))
    k2=h*(omega(i)+l1), l2=h*-g/L*sin(theta(i)+k1)
    theta(i+1)= theta(i)+0.5*(k1+k2)
    omega(i+1)= omega(i)+0.5*(l1+l2)
end
```