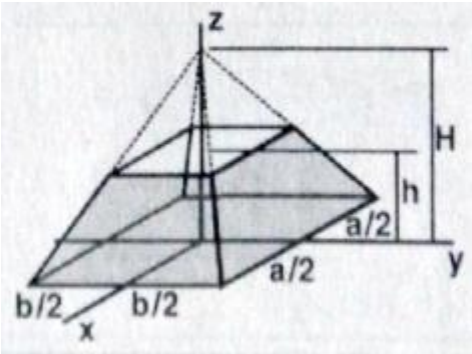
**UNIVERSIDAD NACIONAL DE INGENIERIA** **P.A. 2015-1**
**FACULTAD DE INGENIERIA MECANICA** **15/05/2015**
**DACIBAHCC**

# EXAMEN PARCIAL 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

Sea la pirámide truncada mostrada en la figura:




Cuyo volumen es: $V = \dfrac{abh\left(3H^2 - 3Hh + h^2\right)}{3H^2}$. Si *a=5 cm, b=8 cm, H=18.5 cm, h=9.111 cm*.

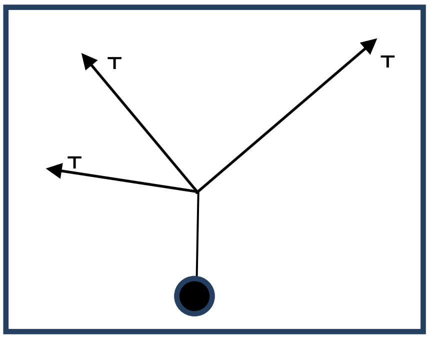
a) **(2.5 pts)** ¿Cuál es el error absoluto permisible en las variables, tal que V tenga un error inferior al 0.5 %?
**Nota: usar el principio de igual efecto**

b) **(1.5 pts)** Si se cuenta con el **sistema IEEE-754 de simple precisión**, halle el Volumen V (redondeado a 2 decimales) obtenido en a) en decimal y binario.

c) **(1.0 pts)** Implemente una ***function*** en MATLAB para hallar el Volumen usando los parámetros de entrada y salida adecuados y muestre el comando para llamarlo con los datos del problema. **Sug**. cabecera: ***function V=calcula(a,b,h,H)***

## Problema 2

En un sistema de cuerdas que sostiene una masa m espacialmente, la tensión en Newtons en cada una se calcula con el siguiente sistema de ecuaciones:




$$2T_1+T_2+T_3=1$$
$$T_1-T_3=1$$
$$4T_1+T_2-2T_3=2$$

Resuelva lo siguiente:

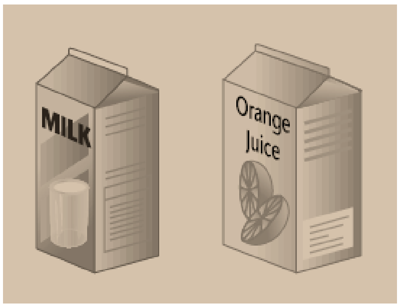
a) **(1.0 pts)** Verifique si el sistema de ecuaciones tiene solución única.

b) **(3.0 pts)** Calcule la solución usando el método de eliminación Gaussiana con pivoteo parcial o total, indicando todos los resultados parciales.

c) **(1.0 pts)** Desarrolle un script en MATLAB que verifique si el sistema está mal condicionado.

## Problema 3

Una persona quiere usar la leche y jugo de naranja para aumentar la cantidad de calcio y vitamina A en su dieta diaria. Una onza de leche contiene 38 miligramos de calcio y 56 microgramos de vitamina A. Una onza de jugo de naranja contiene 5 miligramos de calcio y




60 microgramos de vitamina A. Cada día la persona al beber debería de proporcionar a su dieta exactamente 550 miligramos de calcio y 1.200 microgramos de vitamina A.

a) **(1.0 pts)** Del enunciado del problema demostrar el modelo matemático:

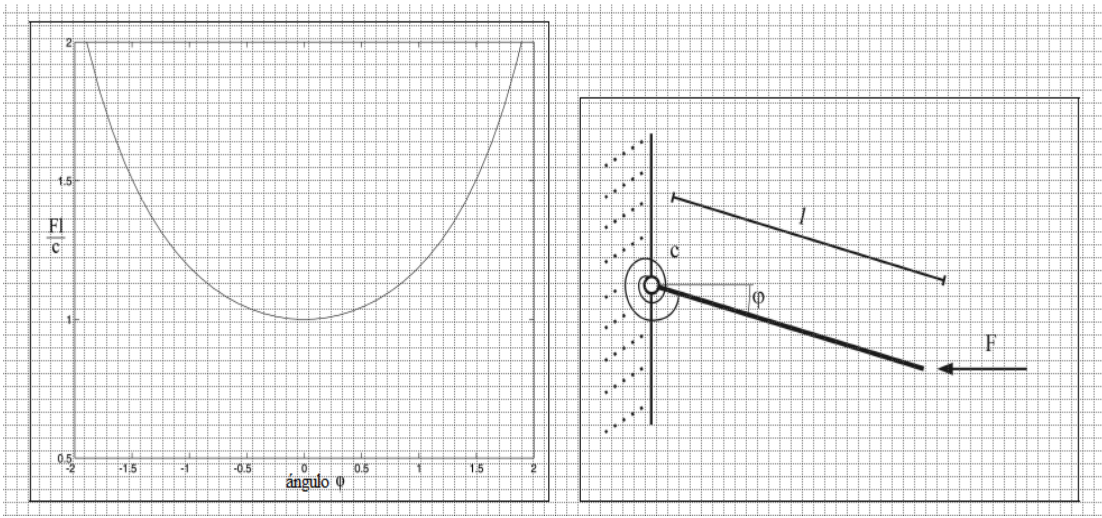
$$\mathbf{38x+5y=550}$$
$$\mathbf{56x+60y=1200}$$

b) **(1.0 pts)** De la parte a) averiguar si al aplicar método el de Jacobi es convergente.

**c)** **(2.0 pts)** Realice 03 iteraciones para aproximar la cantidad de onzas de leche y jugo de naranja utilizando el método de Jacobi. Considere x0=[13;7].

d) **(1.0 pts)** Implementar un script en MATLAB que resuelva c) .

**Problema 4**




En la figura se tiene una varilla de longitud ($l$) que actúa como un péndulo, la rótula presenta fricción (c), y la varilla se desplaza con ángulo $\varphi$ ante una fuerza $F$. Si $f(\varphi)=\varphi-\frac{Fl}{c}\sin(\varphi)$

donde $\frac{Fl}{c}$=1.5, determine el ángulo $\varphi$>0 con <u>**cinco cifras decimales exactas**</u>, para lo cual deberá realizar los siguientes pasos:

**a)** **(1.5 pts)** Utilice el método de la **Bisección** para hallar aproximadamente $\varphi$>0 a partir de las siguientes condiciones: intervalo inicial que tenga un ancho (b-a) = 0.5. **Realice tres iteraciones.**

b) **(1.5 pts) C**onsidere como valor inicial la última iteración de Bisección para aplicar el método de **Newton Rapshon**. **Muestre el algoritmo** y realice **las iteraciones necesarias** para conseguir la aproximación pedida.

c) **(1.0 pts)** Determine el **algoritmo convergente de punto fijo** para encontrar la solución $\varphi$>0 justificando porque converge. No realice iteraciones.

d) **(1.0 pts)** Elabore la función en MATLAB que utiliza el algoritmo de Bisección y que compruebe antes de aplicar el algoritmo si la raíz pertenece al intervalo [a b] en caso contrario envíe un mensaje de error (‘no se puede aplicar el método’) y salga.

**Cabecera : function[ k x]= biseccion(f , a , b, tol)**
**% a, b: límites del intervalo % f: dirección de la f(x) % tol: error máximo**
**% k : número final de iteraciones % x aproximación a la raíz**

**Los Profesores**

**Solucionario**
**Problema 1**

a) **Principio de igual efecto:**

$$V = 214.4223\,cm^3$$
$$\xi_V^* = 0.5\% * V = 1.0721$$

$$\frac{\partial V}{\partial a} = 42.8845$$
$$\frac{\partial V}{\partial b} = 26.8028$$
$$\frac{\partial V}{\partial h} = 10.3028$$
$$\frac{\partial V}{\partial H} = 6.5164$$

$$\xi_a^* = \frac{\xi_V^*}{4\left|\frac{\partial V}{\partial a}\right|} = 0.0062 \quad \xi_b^* = \frac{\xi_V^*}{4\left|\frac{\partial V}{\partial b}\right|} = 0.01 \quad \xi_h^* = \frac{\xi_V^*}{4\left|\frac{\partial V}{\partial h}\right|} = 0.026 \quad \xi_H^* = \frac{\xi_V^*}{4\left|\frac{\partial V}{\partial H}\right|} = 0.0411$$

b) **Número de Simple precision**

V=214.42=11010110. 0110101110000101=1. 10101100110101110000101*2^7

Ei-127=7    Ei=134=10000110

0  10000110  10101100110101110000101

V=214.4199981689453

c) **Funcion en Matlab**

```
function V=calcula(a,b,h,H)

V=a*b*h*(3*H^2-3*H*h+h^2)/(3*H^2);

>>calcula(5,8,9.111,18.5)
```

**Problema 2**

a) Las matrices a analizar:

$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 0 & -1 \\ 4 & 1 & -2 \end{bmatrix}$  Aampliada= $\begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 0 & -1 & 1 \\ 4 & 1 & -2 & 2 \end{bmatrix}$

Por simple inspección se puede notar que cada fila es linealmente independiente por lo tanto el rango en ambos casos es 3 y por consiguiente, tiene solución única.

b) Tomando la matriz ampliada

| | |
|---|---|
| 2 1 1 1<br>1 0 -1 1<br>4 1 -2 2 | El mejor pivote es 4<br>Se cambia fila 1 x 3 |
| 4 1 -2 2<br>1 0 -1 1<br>2 1 1 1 | Aplicamos<br>fila2=fila2-fila1/4<br>fila3=fila3-fila1/2 |
| 4 1 -2 2<br>0 -0.25 -0.5 0.5<br>0 0.5 2 0 | El mejor pivote es 2<br>Se cambia columna 2 x 3<br>Se cambia fila 2 x 3 |
| 4 -2 1 2<br>0 2 0.5 0<br>0 -0.5 -0.25 0.5 | Aplicamos<br>fila3=fila3-fila2/4 |
| 4 -2 1 2<br>0 2 0.5 0<br>0 0 -0.125 0.5 | Aplicamos<br>fila3=fila3-fila2/4 |
| X3=-4<br>X2=1<br>X1=2 | Aplicando sustitución regresiva |
| X1=2<br>X2=-4<br>X3=1 | Cambiamos el orden debido al cambio de columna 2 x 3 |

c) El script seria

```
A=[2  1  1
   1  0 -1
   4  1 -2]
k=norm(A,inf)*norm(inv(A),inf)
if k>10^4
    disp('Esta mal condicionada')
else
    disp('No está mal condicionada')
end
```

**Problema 3**

a)

| | Leche | Juego de Naranja | |
|---|---|---|---|
| **Calcio** | 38 | 5 | 550 |
| **Vitamina A** | 56 | 60 | 1,200 |

x: número de onzas de leche

Y: número de onzas de jugo de naranja

$$38x + 5y = 550$$
$$56x + 60y = 1200$$

b) El sistema es diagonalmente estrictamente dominante por lo tanto el método de Jacobi es convergente.

c)

| | |
|---|---|
| 13.0000 | 7.0000 |
| 13.5526 | 7.8667 |
| 13.4386 | 7.3509 |
| 13.5065 | 7.4573 |

d)

```
D=diag(diag(A))
L=tril(-A,-1)
U=triu(-A,1)
Tj=inv(D)*(L+U);
cj=inv(D)*b;
z=[x0'];
for k=1:3
x1=Tj*x0+cj;
z=[z;x1'];
x0=x1;
end
z
```

**Problema 4**

**a)Bisección**

$$f(\varphi)=\varphi-\frac{Fl}{c}\sin(\varphi)=0$$

Usando el gráfico supongo *a*=1.75 y *b*=1.25 f(a)*f(b)<0 ➔ $\varphi^*\in[a\ b]$

| k | a | b | x=(a+b)/2 | f(a) | f(x) | e=(b-a)/2 |
|---|---|---|---|---|---|---|
| 0 | 1.25 | 1.75 | 1.5 | - | + | 0.25 |
| 1 | 1.25 | 1.5 | 1.375 | - | - | 0.125 |
| 2 | 1.375 | 1.5 | 1.4375 | - | - | 0.0625 |
| 3 | 1.4375 | 1.5 | 1.4688 | - | - | 0.0313 |

**b) Algoritmo de Newton**

$$\varphi_{i+1}=\varphi_i-\frac{\varphi_i-1.5\sin(\varphi_i)}{1-1.5\cos(\varphi_i)}$$

$\varphi_o$=1.4688

| k | $\varphi_k$ | $f(\varphi_k)$ | $f'(\varphi_k)$ | abs( $f(\varphi_k)/f'(\varphi_k)$ ) | |
|---|---|---|---|---|---|
| 0 | 1.4688 | -0.023404 | 0.847271 | 0.0276232 | |
| 1 | 1.49642 | 0.00057 | 0.888543 | 0.641 $\times10^{-3}$ | |
| 2 | 1.495782 | 0.3076$\times10^{-6}$ | 0.887584 | **0.3465 $\times10^{-6}$** | < **Tol= 0.5 $\times10^{-5}$** |
| 3 | 1.495782 | | | | |

**c) Algoritmo convergente de punto fijo**

$$g(\varphi) = 1.5\sin(\varphi)$$

Comprobando la convergencia *a*=1.75 y *b*=1.25

$g'(\varphi_i) = \sup\left|1.5\cos(\varphi_i)\right|_a^b \leq 0.473 = k < 1$ Por lo tanto converge

$$\varphi_{i+1} = 1.5\sin(\varphi_i)$$

**d) Function en MATLAB**

```
function [k,x]=biseccion(f,a,b,tol)
if f(a)*f(b)>0 , error('no se puede aplicar el método')
end;
k=0;
while (b-a)/2>tol
    k=k+1;
    x=(a+b)/2;
        if f(a)*f(x)<0, b=x;
        else a=x;
        end
end
```