**Universidad Técnica de Loja**
**Escuela de Ingeniería Química**
**Introducción a MATLAB**

**Curso Marzo de 2007 – Agosto de 2007**

**MATRICES Y VECTORES**

**P001.** Dados los vectores $\mathbf{x} = [3\ \ 2\ \ 6\ \ 8]$ y $\mathbf{y} = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 5 \end{bmatrix}^T$,

a. Obtenga la suma de los elementos de **x** y **y**
b. Obtenga un vector **z** cuyos componentes sean los elementos del vector **x** elevados a la potencia especificada por cada elemento correspondiente en el vector **y**.
c. Dividir cada elemento de **y** para cada elemento correspondiente de **x**.
d. Obtenga un vector **z** cuyos componentes sean los elementos del vector **x** multiplicados por cada elemento correspondiente del vector **y**.
e. Ejecutar la operación $\mathbf{x}^\mathbf{T}\mathbf{y} - \mathbf{z}$

**P002.** Obtener un vector cuyos componentes:

a. Se encuentren entre 5 y 25, y separados por 5 unidades.
b. Sean los números entre 10 y 30 separados por una unidad.
c. 6 números entre 0 y 20 igualmente espaciados

**P003.** Construir una matriz A de 2x3 cuyas filas son los 6 primeros impares consecutivos.

a. Anular el elemento (2,3)
b. Obtener la matriz B = A'
c. Construir una matriz C, formada por la matriz B y la matriz identidad de orden 3 adosada a su derecha
d. Construir una matriz D extrayendo las columnas impares de la matriz C
e. Construir una matriz E formada por la intersección de las dos primeras filas de C y sus columnas tercera y quinta
f. Construir una matriz F formada por la intersección de las dos primeras filas y las tres últimas columnas de la matriz C
g. Construir una matriz diagonal G tal que los elementos de su diagonal principal son los mismos que los de la diagonal principal de D
h. Calcular el orden de la matriz C

**P004.** Dada una matriz M cuadrada aleatoria uniforme de orden 3:

a. Obtener su inversa, su transpuesta y su diagonal
b. Transformarla en una matriz triangular inferior y en otra superior.
c. Obtener la suma de los elementos de la primera fila y la suma de los elementos de la diagonal.
d. Extraer la submatriz cuya diagonal son los elementos $a_{11}$ y $a_{22}$ y extraer también la submatriz cuyos elementos de la diagonal son $a_{11}$ y $a_{33}$.

**P005.** Dados $\mathbf{x} = [3\ 1\ 5\ 7\ 9\ 2\ 6]$, explicar el significado de los siguientes comandos:

a. x(3)
b. x(1:3)
c. x(1:end)
d. x(1:end-1)
e. x(6:-2:1)
f. x([1 6 2 1 1])

**P006.** Dados los arreglos $\mathbf{x} = [1\ 4\ 8]$, $\mathbf{y} = [2\ 1\ 5]$, y $\mathbf{A} = \begin{bmatrix} 3 & 1 & 6 \\ 5 & 2 & 7 \end{bmatrix}$, determine cuál de los siguientes planteamientos no se ejecutará correctamente:

a. $A - [x'\ y']$
b. $[x\,;\,y']$
c. $[x\,;\,y]$
d. A - 3

**P007.** Ingresar las matrices $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 4 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} -1 & 1 \\ 5 & 5 \end{bmatrix}$ y $\mathbf{D} = \begin{bmatrix} -1 & 1 & 3 \\ 2 & 2 & 2 \end{bmatrix}$, y evaluar cada una de las siguientes expresiones. Explicar cualquier mensaje de error:

a. 2*A+3*C
b. A-4*D
c. B^2
d. B.^2
e. A*B
f. B*A
g. C*D
h. C.*D
i. A*B+D

**P008.** Considere los escalares $x_1 = -2$, $x_2 = 1$, $x_3 = -4$, y los vectores

$$\mathbf{a_1} = \begin{bmatrix} -1 \\ 1 \\ 2 \\ 3 \end{bmatrix},\ \mathbf{a_2} = \begin{bmatrix} -2 \\ 0 \\ 2 \\ 0 \end{bmatrix},\ \mathbf{a}_3 = \begin{bmatrix} -1 \\ 2 \\ 3 \\ -3 \end{bmatrix},\ \mathbf{a}_4 = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

a. Use MATLAB para calcular la combinación lineal $x_1\mathbf{a_1} + x_2\mathbf{a_2} + x_3\mathbf{a_3}$

**P009.** Considere los escalares $x_1 = -2$, $x_2 = 3$, $x_3 = 2$, y los vectores

$$\mathbf{v_1} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}^T, \quad \mathbf{v_2} = \begin{bmatrix} 0 \\ -4 \\ 5 \end{bmatrix}^T, \quad \mathbf{v_3} = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}^T$$

b. Use MATLAB para calcular la combinación lineal $x_1\mathbf{v_1} + x_2\mathbf{v_2} + x_3\mathbf{v_3}$

c. Forme el vector fila $\mathbf{x} = [x_1 \;\; x_2 \;\; x_3]$ y la matriz $\mathbf{v} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \mathbf{v_3} \end{bmatrix}$ y calcular el producto vector-matriz $\mathbf{xV}$.

**P010.** Crear las matrices $\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 & 3 \\ 2 & 2 & 0 & 0 \\ 5 & 6 & 3 & 1 \\ 0 & 2 & 0 & 5 \end{bmatrix}$, $\mathbf{Y} = \begin{bmatrix} 1 & 2 & 3 & 3 \\ -1 & 2 & 0 & 0 \\ 5 & 6 & 3 & 1 \\ 0 & 2 & 0 & 5 \end{bmatrix}$. Usar operaciones entre arreglos para responder cada uno de los literales:

a. Crear una matriz A cuyos elementos i,j son $a_{ij} = 2x_{ij} - 3y_{ij}$
b. Crear una matriz A cuyos elementos i,j son $a_{ij} = {x_{ij}}^2$
c. Crear una matriz A cuyos elementos i,j son $a_{ij} = y_{ij} - 3$
d. Crear una matriz A cuyos elementos i,j son $a_{ij} = {x_{ij}}^2 - {y_{ij}}^2$
e. Crear una matriz A cuyos elementos i,j son $a_{ij} = e^{{x_{ij}}^2 + {y_{ij}}^2}$

**P011.** Sea la matriz cuadrada

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 3 \\ 3 & 3 & 1 \end{bmatrix}$$

a. Construir una matriz añadiendo la matriz identidad de rango 3 a la derecha de la matriz A.
b. Sumar a la tercera fila, la primera fila multiplicada por 3.
c. Cambiar la primera columna de A por la tercera.
d. Construir una nueva matriz cuyas columnas sean las columnas primera y tercera de A.
e. Construir una nueva matriz cuyas filas sean las columnas primera y tercera de *A*.

**P012.** Sea la matriz cuadrada

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 0 \\ 0 & 2 & 3 \\ 3 & 3 & 7 \end{bmatrix}$$

a. Hallar el valor mínimo dentro de cada fila de A.
b. Ordenar los elementos de A en orden descendente dentro de cada columna.
c. Ordenar los elementos de A en orden ascendente dentro de cada fila.
d. Formar una lista con los elementos de A ordenada de forma ascendente.

e. Hallar el máximo en valor absoluto de los elementos de la matriz A.

**P013.** En una sola orden de MATLAB crear una matriz 3 *x* 5 cuyo único elemento sea el 7.

**P014.** Con una sola orden de MATLAB crear una matriz aleatoria 4 *x* 4 de números reales entre -5 y 5.

**P015.** Considerar la siguiente orden de MATLAB: A = magic(5). En una sola orden:

a. Definir una matriz B formada por las filas pares de la matriz A.
b. Definir una matriz C formada por las columnas impares de la matriz A.
c. Definir una vector d formada por la tercera columna de la matriz A.
d. Eliminar la tercera fila de la matriz A.

**P016.** Sea x = (0:pi/2:2*pi). Con una sola orden de MATLAB crear una matriz cuya primera fila es x, su segunda fila es el seno de cada elemento de x y cuya tercera fila el coseno de cada elemento de x.

**P017.** Definir un vector a formado por los cuatro primeros números impares y otro b formado por los cuatro primeros números pares de varias formas distintas. Emplearlos para construir la matriz

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 6 & 8 \\ 6 & 12 & 18 & 24 \\ 10 & 20 & 30 & 40 \\ 14 & 28 & 42 & 56 \end{bmatrix}$$

**P018.** Construir una matriz n *x* n, C = ($c_{ij}$)

a. Con $c_{ij} = i \cdot j$;
b. Con $c_{ij} = \cos(i \cdot j)$;

**P019.** Construir de distintas formas la matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

**P020.** En una sola instrucción, cambiar todos los valores de la diagonal de una matriz cuadrada a cero.

**P021.** En una sola instrucción, sustituir todos los valores de la diagonal de una matriz cuadrada por los elementos de un vector dado.

**P022.** Ordenar los elementos de una matriz del menor al mayor manteniendo su forma (indicación: emplear la orden reshape.)

**P023.** En una sola instrucción, poner a cero todos los elementos negativos de una matriz.

**P024.** En una sola instrucción, poner a cero todos los elementos de una matriz que estén entre -1 y 1. (La conjunción lógica es &).

**P025.** De tres formas distintas (cada una en una sola instrucción), averiguar el número de elementos de una matriz, de forma que al final tengamos un número.

**P026.** Resolver los siguientes sistemas de ecuaciones lineales:

## OPERADORES RELACIONALES

**P027.** Dados los escalares $a = 5, b = 2, c = -4$, determinar los valores de verdad de las proposiciones.

a. $(a < b \text{ and } a > c) \text{or} (b > c)$
b. $(a \geq b \text{ and } b \leq c) \text{and} (a > c)$
c. $\text{or } (a > b) \text{or} (a > c)$
d. $\text{or} (a > b) \text{or} ((a > c) \text{and} (b < a))$
e. $(a < b \text{ and } a > c) \text{or} (b > c)$

## PROGRAMACIÓN

**P028.** Escribir un programa que permita determinar si un número entero dado es par o impar. (Utilizar el operador **mod**)

**P029.** Utilizando el operador relacional > (mayor que), escribir un archivo.m que permita definir si un número **a** es mayor que un número **b**. El programa debe admitir ingresar los números a y b, e imprimir el resultado ***a es mayor que b***, o ***a es menor que b***, o ***a es igual a b***.

**P030.** Escribir un archivo.m que de como resultado el menor de tres números a, b, c. Utilice la sentencia de control **if…elseif…end**

**P031.** Resolver E3 utilizando la sentencia **switch…case…otherwise…end**

**P032.** Crear un archivo.m que calcule las raíces de la ecuación: $a \cdot x^2 + b \cdot x + c = 0$. Teniendo en cuenta los siguientes casos:

1. Si a=0 y b=0, imprimiremos un mensaje diciendo la ecuación es degenerada.
2. Si a=0 y b≠0, existe una raíz única con valor –c / b.
3. En los demás casos utilizaremos la fórmula siguiente:

$$x_i = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2a}$$

La expresión $d=b^2-4ac$ se denomina discriminante.

- Si $d \geq 0$ entonces hay dos raíces reales
- Si $d < 0$ entonces hay dos raíces complejas de la forma $x + yj$, $x - yj$

Indicar con literales adecuados los datos a introducir, así como los resultados obtenidos.
**a.** Utilizando la sentencia de control **if…then…elseif…end**
**b.** Utilizando la sentencia **switch…case…otherwise…end**

**P033.** Escribir un programa que permita imprimir los números impares del 1 al 100. Utilizar la sentencia **for…end**

**P034.** Escribir un programa que permita imprimir los números impares del 1 al 100. Utilizar la sentencia **while … end**

**P035.** Escribir un archivo.m que de como resultado la suma de los 100 primeros números naturales.

a. Utilizar la sentencia **for…end**
b. Utilizar la sentencia **while … end**

**P036.** Crear un archivo.m que de como resultado la suma de los números pares comprendidos entre el 2 y el 100.

**P037.** Escribir un programa que dibuje un triángulo de n filas, empleando el caracter asterisco. Crear el programa utilizando:

a. Sentencias **for…end**
b. Sentencias **while…end**

Por ejemplo, para n=4
```
*******
 *****
  ***
   *
```

**P038.** Implementar un programa que permita evaluar el factorial de un número entero positivo. Por ejemplo
Si n = 5: S = 5*4*3*2*1
Si n = 3: S = 3*2*1

**P039.** Crear archivos.m de función para evaluar la suma total de los n primeros términos dadas las siguientes series:

a. $S = 1 + 3 + 5 + 7 + ...$

b. $S = -\frac{1}{1} + \frac{1}{2} - \frac{1}{3} + \frac{1}{4} - ...... + \frac{1}{n}$

c. $S = 1-2+3-5+8-13+21-34+.....$

d. $S = \frac{1}{1!} - \frac{1}{3!} + \frac{1}{5!} - \frac{1}{7!} + ......$

**P040.** Implementar un programa que permita ingresar n números y ordenarlos de acuerdo a las opciones de mayor a menor.

**P041.** Implementar un programa que permita ingresar n números y ordenarlos de acuerdo a las opciones de menor a mayor.

**P042.** Implementar un programa que permita determinar si un número es primo.

**P043.** Utilizando la sentencia **for…end**, escribir un programa que imprima un tablero de ajedrez en el que las casillas blancas se simbolizarán con una B y las negras con una N. Así mismo el programa deberá marcar con * las casillas a las que se puede mover un alfil desde una posición dada. La solución será similar a la siguiente:

Posición del alfil:
Fila 3
Columna 4

B * B N B * B N
N B * B * B N B
B N B * B N B N
N B * B * B N B
B * B N B * B N
* B N B N B * B
B N B N B N B *
N B N B N B N B

**P044.** Crear un archivo.m que permita evaluar las series. Los argumentos de entrada son **x** y **n**, donde n es el número de términos que se evalúan en la serie y x es un número real:

a. $S = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots$

b. $S = \frac{1}{2}x - \frac{1\cdot 3}{2\cdot 4}x^3 + \frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6}x^5 - \frac{1\cdot 3\cdot 5\cdot 7}{2\cdot 4\cdot 6\cdot 8}x^7 + \ldots$

c. $S = x - \frac{1}{2}\frac{x^1}{(1+x)^1} + \frac{(1-3)}{2\cdot 4}\frac{x^2}{(1+x)^3} - \frac{(1-3+5)}{2\cdot 4\cdot 6}\frac{x^3}{(1+x)^5} + \frac{(1-3+5-7)}{2\cdot 4\cdot 6\cdot 8}\frac{x^4}{(1+x)^7} - \ldots$

**P045.** Implementar un programa que permita ingresar un vector con N números y posteriormente permita evaluar la media aritmética.

**P046.** Realizar un programa que permita ingresar una matriz, y posteriormente imprimir la transpuesta de dicha matriz. Por ejemplo

$$\text{Matriz} = \begin{bmatrix} 1 & 8 & 4 \\ 2 & 3 & 8 \\ 3 & 9 & 12 \end{bmatrix}$$

$$\text{MatrizTranspuesta} = \begin{bmatrix} 1 & 2 & 3 \\ 8 & 3 & 9 \\ 4 & 8 & 12 \end{bmatrix}$$

**P047.** Implementar un programa que permita ingresar n números y posteriormente ordenar e imprimir los resultados de mayor a menor.

**P048.** Implementar un programa que de como resultado los números primos menores que un número dado n (positivo y entero).

**P049.** Escribir un programa que lea una cadena de n caracteres e imprima el resultado que se obtiene cada vez que se realice una rotación de un carácter a la derecha sobre dicha cadena. El proceso finalizará cuando se haya obtenido nuevamente la cadena de caracteres original. Por ejemplo:
HOLA AHOL LAHO OLAH HOLA

**P050.** Escribir un programa que permita ingresar una palabra, y posteriormente imprimir la palabra con las letras en orden inverso. Por ejemplo:
Se ingresa: INGENIERIA
Se imprime: AIREINEGNI

**P051.** Pedir al usuario un número e indicarle si corresponde a un año bisiesto o no. (NOTA: Por lo general, son bisiestos los años múltiplos de cuatro, salvo aquellos que también sean múltiplos de 100, que no son bisiestos. De esto últimos, aquellos que sean múltiplos de 400 sí son bisiestos. Así, 1996 , 2000 y 2004 son bisiestos, pero 1900 y 2001 no lo son.)

**P052.** Imprimir los múltiplos de 7 existentes entre dos valores ***m*** y ***n*** pedido al usuario. Modificar el algoritmo para que calcule cuántos números hay y su suma.

**P053.** Imprimir los ***n*** primeros términos de la serie de Fibonacci. Los términos de dicha serie se calculan de la siguiente forma: $a_0=1$; $a_1=1$; $a_i= a_{i-1}+a_{i-2}$, $i=2,\ldots n$.

**P054.** Pedir al usuario dos números ***n*** y ***k*** y calcular (combinaciones de *n* elementos tomados de ***k*** en ***k***).

**P055.** Imprimir todos los números primos que hay entre dos pedidos al usuario.

**P056.** Construir un algoritmo para adivinar un número secreto. Guardar en una variable dicho número "secreto" y permitir que el usuario haga intentos para adivinarlo. En cada intento, el algoritmo deberá informar si ese intento es mayor o menor que el número secreto. Al acertar, se informará del número de intentos que han sido necesarios. (NOTA: en DFD, podéis usar la función random para generar aleatoriamente un número secreto.).

**P057.** Calcular la posición de la primera componente negativa de un vector pedido al usuario.

**P058.** Calcular cuántas componentes de una matriz son negativas, positivas y cero.

**P059.** Calcular el número de cifras de un número entero ***n*** pedido al usuario. (NOTA: puede dividirse sucesivamente n entre 10, hasta alcanzar un cociente que valga cero).

**P060.** Diseñar un algoritmo que permita calcular la menor componente de un vector. Modificar el algoritmo para que también indique cuál es su posición.

.

**P061.** Diseñar un programa que calcule el producto de dos matrices pedidas al usuario.

**P062.** Diseñar un algoritmo que permita calcular la posición (fila y columna) de la mayor componente de una matriz pedida al usuario.

**P063.** Convertir a binario puro un número real en base 10 pedido al usuario.

**P064.** Diseñar un programa que informe al usuario si una frase pedida al usuario es un palíndromo. Un palíndromo es una frase que se lee igual en los dos sentidos ('A torre da derrota', 'Dábale arroz a la zorra el abad'). Para simplificar, el usuario introducirá la frase en minúsculas, sin tildes ni espacios en blanco. NOTA: En DFD puede utilizarse la función ***substring.***

**P065.** Calcular si dos números enteros ***a*** y ***b*** son amigos. Se dice que ***a*** es amigo de ***b*** si ***a*** es igual a la suma de los divisores primos de ***b***. A estos efectos, no se cuenta como divisor el propio número ***b***. (Ejemplo: el 1 es amigo del 19, y también el 11 del 21).

**P066.** Usar la función anterior para diseñar un algoritmo que encuentre todos los números perfectos entre 0 y un número ***a*** pedido al usuario. (NOTA: un número es perfecto si es amigo de sí mismo).

## GRAFICACIÓN

**P067.** Use los comandos de MATLAB para graficar las siguientes funciones, en el dominio especificado:

a. $f(x) = 5 - 4x - x^2$, [-6,2]
b. $f(x) = 2x^2 - 8x - 11$, [-1,5]
c. $f(t) = te^{-2t}$, [-1,5]
d. $h(t) = e^{-0.1t}\sin(2t)$, $[0,4\pi]$

**P068.** Use los comandos de MATLAB para graficar dentro de los mismos ejes, las siguientes funciones, en el dominio especificado:

a. $f(x) = \cos(3x) + \sin(3x)$, $v(x) = -2\sin(2x) + 3\cos(3x)$, $[0,4\pi]$

b. $f(x) = xe^{-3x}$, $v(x) = e^{-3x}(1 - 3x)$, $[0,2]$

c. $f(x) = \sin(3t)\cos(2t)$, $g(x) = \frac{1}{2}\cos(t) + \frac{5}{2}\cos(5t)$, $[0,4\pi]$

d. $f(t) = (1 + 2\sin(t))\cos(t)$, $g(t) = (1 + 2\sin(t))\sin(t)$, $[0,2\pi]$

**P069.** En cada uno de los siguientes ejercicios, x y y están definidos en términos de un parámetro t sobre un intervalo dado. Use los comandos de MATLAB para graficar y versus x. Etiquete cada eje y coloque un título al gráfico.

a. $x(t) = t - \sin(t)$, $y(t) = 1 - \cos(t)$, $[0,6\pi]$

b. $x(t) = \cos^3 t$, $y(t) = \sin^3 t$, $[0,2\pi]$

c. $x(t) = 2\cos(t) + \sin(t)$, $y(t) = 2\sin(t) - \sin(2t)$, $[0,2\pi]$

d. $x(t) = \sin(3t)$, $y(t) = \cos(3t)$, $[0,2\pi]$

**P070.** Usar el comando **comet** para animar los gráficos de los ejercicios P069.

Ejemplo:

```
>> t=linspace(0,2*pi,4000);
>> x=cos(5*t);
>> y=sin(7*t);
>> comet(x,y)
```

**P071**. Representar la superficie cuya función explícita es la siguiente:

a. $z = \dfrac{\sin\left(\sqrt{x^2 + y^2}\right)}{\sqrt{x^2 + y^2}}$, $-2 \le x \le 2$, $-2 \le y \le 2$

b. $z = \sin(x + y)\exp^{-x^2+y^2}$, $-2 \le x \le 2$, $-2 \le y \le 2$

c. $z = 3x^2 + y^2 - xy + x + 2y$, $-2 \le x \le 2$, $-2 \le y \le 2$

d. $z = x^2 - 2y^2 + \frac{1}{2}xy - 2x + y$, $-2 \le x \le 2$, $-2 \le y \le 2$

## PROBLEMAS ADICIONALES

### P072. Centro de masas de un sistema de partículas

Realizar un programa en Matlab que calcule el centro de masas de un sistema de partículas en 3D. Interés: Sumatorios y medias.

### P073. Solución de una ecuación por Newton-Raphson

Realizar un programa en Matlab que calcule la solución de una ecuación de forma iterativa utilizando el método de Newton-Raphson. Interés: Uso de funciones y cálculos iterativos.

**P074. Producto escalar y vectorial**

Realizar un programa en Matlab que calcule el producto escalar y vectorial de dos vectores en 3D. Interés: Manipulación y operación con vectores.

**P070. Resolución de sistemas de ecuaciones lineales**

Realizar un programa en Matlab que resuelva un sistema de ecuaciones lineales para el caso de 3 ecuaciones con 3 incógnitas. Utilizar por una parte las facilidades de matlab (operador \) así como la programación directa del método de Cramer. Interés: Bucles anidados y manipulación de matrices

**P075. Integral aproximada por rectángulos**

Realizar un programa en Matlab que calcule la integral aproximada de una función utilizando una aproximación por rectángulos. Interés: Resolución numérica

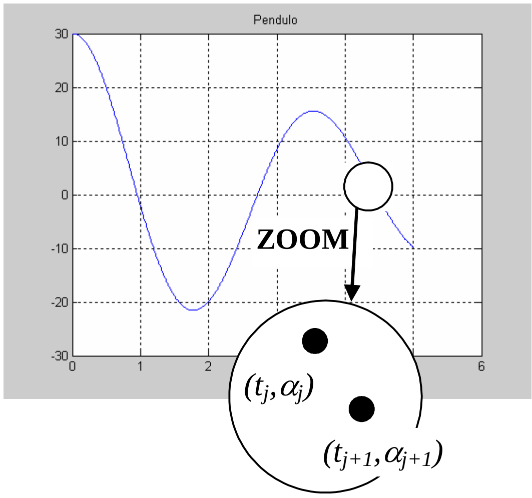
**P076. Simulación dinámica**

Realizar un programa en Matlab que simule dinámicamente un péndulo por el método de Euler. Las ecuaciones diferenciales que rigen la posición de un péndulo, sometido a rozamiento, vienen dadas por:

$$\frac{dv}{dt} = -\frac{g}{l} \cdot \sin(\alpha) - \mu \cdot v$$
$$\frac{d\alpha}{dt} = v$$

siendo $v$ la velocidad angular (en rad/s), $\alpha$ el ángulo (en radianes), g la aceleración de la gravedad (9.8 $m/s^2$), $l$ la longitud del péndulo y $\mu$ el coeficiente de rozamiento.
Para realizar la simulación en el ordenador de dichas ecuaciones diferenciales, existen métodos de integración que nos dan de forma iterativa los valores que van tomando las variables que están diferenciadas (en nuestro caso, $v$ y $\alpha$). Uno de los métodos más sencillos se denomina *método de Euler*. Aplicando dicho método al caso dado, se puede obtener que el ángulo $\alpha$ a lo largo del tiempo se puede ir calculando de forma iterativa de la siguiente manera:

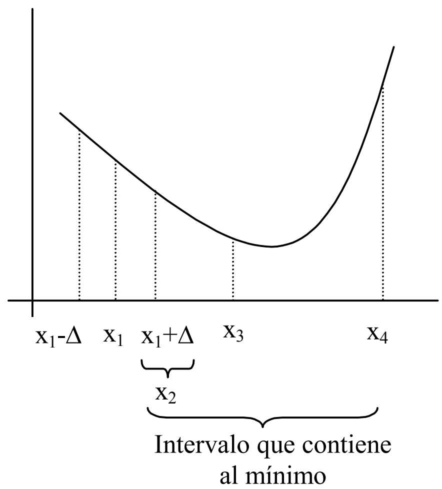
$$v_{j+1} = v_j - h \cdot \left( \frac{g}{l} \cdot \sin(\alpha_j) + \mu \cdot v_j \right)$$
$$\alpha_{j+1} = \alpha_j + h \cdot v_j$$
$$t_{j+1} = j \cdot h$$




donde *h* se denomina paso de integración, que debe elegir el usuario, y *j* indica el punto que estamos calculando (*j*=1,2,3,...), siendo las condiciones iniciales: $v_1$=0 (soltamos el péndulo sin darle impulso), $\alpha_1$ elegido por el usuario, $t_1$=0, siendo *t* el tiempo.
Realizar un programa que pida al usuario la longitud del péndulo, el ángulo inicial (en grados), el coeficiente de rozamiento, el paso de integración y el tiempo máximo de simulación. El programa generará la evolución del péndulo a lo largo del tiempo en forma de puntos ($t$,$\alpha$) hasta que se llegue al tiempo máximo. Finalmente, el programa dibujará $\alpha$ a lo largo del tiempo.
*Nota*: El valor de *h* es función del tipo de problema al que se aplica. Si se elige una longitud del péndulo de 3 m, un ángulo inicial de 30º y un coeficiente de rozamiento de 0.4, un valor adecuado de *h* podría ser 0.01 para una simulación de 5 s.

## P077. Acotación del mínimo de una función

Realizar una función en Matlab que aplique el método de Swann a una función. Muchos métodos de optimización escalar precisan un intervalo donde la función a optimizar sea unimodal. Ello significa que la función tiene un único mínimo, por lo que la forma de la función será como aparece en la figura de la derecha. Existen métodos para, dado un punto cualquiera $x_1$, encontrar un intervalo [a,b] en el cual haya un mínimo local. Uno de estos métodos consiste en lo siguiente:




Dado un punto $x_1$ y un valor Δ, la primera fase consiste en averiguar en qué dirección la función decrece. Si $f(x_1 - \Delta) > f(x_1) > f(x_1 + \Delta)$ (como en el dibujo), entonces la función decrece hacia la derecha, por lo que se considera Δ como positivo. Si $f(x_1 - \Delta) < f(x_1) < f(x_1 + \Delta)$, entonces la función decrece hacia la izquierda, por lo que se cambia a Δ de signo.
Existen otros dos casos. Si $f(x_1 - \Delta) > f(x_1) < f(x_1 + \Delta)$ entonces el intervalo es $[x_1 - \Delta, x_1 + \Delta]$ y no es necesario realizar más cálculos; y si $f(x_1 - \Delta) < f(x_1) > f(x_1 + \Delta)$ entonces la función no es unimodal y devolvemos un mensaje indicándolo.
Sin pérdida de generalidad, supongamos que la función decrece hacia la derecha y llamemos $x_2=x_1+\Delta$ y k=3. A continuación, la segunda fase realiza de forma **iterativa** un cálculo del siguiente punto y una comprobación para ver si se ha encontrado el intervalo de la siguiente manera:

1. $x_k = x_{k-1} + 2^{k-2}\Delta$

2. Si $f(x_k)>f(x_{k-1})$, entonces el intervalo es $[x_{k-2}, x_k]$
    - ✓ En caso contrario, hacemos k=k+1 y volvemos al paso 1

Véase en la figura una explicación gráfica del procedimiento.
El procedimiento iterativo debe incluir un límite máximo de iteraciones N, para el caso de que no se encuentre dicho intervalo.

Realizar una función (llamada acotar___) en Matlab que devuelva el intervalo correspondiente cuando se le pasan como argumentos el punto inicial $x_1$, el valor $\Delta$ y el máximo número de iteraciones N. La función que se debe considerar es $f(x) = (100 - x)^2$ y se debe probar el funcionamiento de la función ejecutando [a,b]=acotar___(30,5,10) donde los argumentos son respectivamente $x_1$, $\Delta$ y N.