Introducción a la Programación y Análisis Numérico
Práctica 1: Programación en OCTAVE/MATLAB

Ej. 1: Lo que impone el sistema

Dado el siguiente sistema de ecuaciones

$$\begin{cases} 42x + 66y + 68z + 66w &= 7 \\ 92x + 4y + 76z + 17w &= 3 \\ 79x + 85y + 74z + 71w &= 10 \\ 96x + 93y + 39z + 3w &= 0 \end{cases}$$

a) Expresar en forma matricial y verificar que tiene solución única.

b) Resuelva utilizando OCTAVE/MATLAB. Verificar que el conjunto de valores hallados es efectivamente solución del sistema.

Ej. 2: Ilusión óptica

Defina los vectores `t`, `x` e `y` de la siguiente manera:

```
t=[0:2*pi/50:2*pi];
x=2*cos(t);
y=5*sin(t);
```

a) Grafique `x` e `y` en función `t`. Los comandos `plot`, `figure` y `subplot` pueden serle útiles.

b) La curva $\mathbf{r}(\mathbf{t}) = (\mathrm{x}(\mathrm{t}), \mathrm{y}(\mathrm{t}))$ representa una elipse descripta en forma paramétrica. Para graficarla puede utilizar `plot(x,y)`.

c) ¿Cuál parece ser el eje mayor de la elipse? ¿Cuál es realmente el eje mayor de la elipse?¿A qué se debe ésto? Puede serle de utilidad la sentencia `axis equal`

d) Grafique la siguiente curva descripta en forma paramétrica:

$$\begin{aligned} x &= \cos(t) \cdot \left(e^{\cos(t)} - 2\cos(4t) - sen^5\left(\frac{t}{12}\right)\right) \\ y &= sen(t) \cdot \left(e^{\cos(t)} - 2\cos(4t) - sen^5\left(\frac{t}{12}\right)\right) \end{aligned}$$

Vea que sucede al modificar el paso de la variable t (puede comenzar con el mismo t de los incisos anteriores). También pruebe extendiendo el límite superior de la variable t, por ejemplo, a 4π, 8π, etc... Resulta interesante ver cómo se va formando la curva a medida que varía el parámetro t. Para ello podemos intentar una especie de "animación" en OCTAVE/MATLAB con los comandos siguientes:

```
figure, axis([-3 3 -4 4]), hold on
for i=1:length(t)-1
plot(x(i:i+1),y(i:i+1));
pause(1/10);
end
hold off
```

Ej. 3: `for` ever

a) Dado el siguiente *script* de OCTAVE/MATLAB:

```
for k=1:10
    for h=1:10
        if k==h
            A(k,h)=1;
        elseif (k<h) && ~(1>2 || k==h)
            A(k,h)=2./(((k+h).^2)+1);
        else
            A(k,h)=0;
        end
    end
end
```

Describa qué resulta de su ejecución.

c) Genere un vector con todos los números $a \in \mathbb{Z}^+$ de modo que $a < 2000$ y que a sea múltiplo de 2, 7 y 13. **Pista:** Puede serle útil el comando `mod`.

d) Escriba una función de OCTAVE/MATLAB que permita obtener la *matriz de columnas cíclicas* de orden n dado un vector $v \in \mathbb{R}^n$. Es decir, para $n = 4$, si $v = [a, b, c, d]^T \in \mathbb{R}^4$, la matriz buscada será

$$A = \begin{pmatrix} a & b & c & d \\ b & c & d & a \\ c & d & a & b \\ d & a & b & c \end{pmatrix}$$

Pueden serle de utilidad las funciones `circshift`, `size` y `length`.

Ej. 4: La geométrica es cosa serie

Se sabe que la suma de la serie geométrica es:

$$\sum_{n=1}^{\infty} a \cdot r^n = \frac{a \cdot r}{1 - r} \quad \text{si } |r| < 1$$

a) Intente verificar con OCTAVE/MATLAB que

$$\sum_{n=1}^{\infty} \frac{1}{2^n} = \frac{\frac{1}{2}}{1 - \frac{1}{2}} = 1$$

para lo cual puede seguir los siguientes pasos:

- Para un número N suficientemente grande (por ejemplo 50, 100, etc.) defina un vector $\mathbf{x}$ de largo N cuyos elementos sean
$$\mathbf{x}(n) = \left(\frac{1}{2}\right)^n$$
- Defina un segundo vector $\mathbf{y}$, también de largo N, donde cada elemento $\mathbf{y}(n)$ contenga el resultado de la suma de los elementos de $\mathbf{x}$, desde 1 a n, así:

$$\begin{aligned} \mathbf{y}(1) &= \mathbf{x}(1) \\ \mathbf{y}(2) &= \mathbf{x}(1) + \mathbf{x}(2) \\ &\vdots \\ \mathbf{y}(N) &= \mathbf{x}(1) + \mathbf{x}(2) + \cdots + \mathbf{x}(N-1) + \mathbf{x}(N) \end{aligned}$$

- Grafique los elementos del vector $\mathbf{y}$, y vea como se comportan los mismos para valores grandes de n. Para interpretar este comportamiento quizá le sirva graficar también los elementos del vector $\mathbf{x}$.

b) De la misma forma intente probar que

$$
\begin{aligned}
\sum_{n=1}^{\infty}(0{,}9)^n &= \frac{0{,}9}{1-0{,}9} = 9 \\
\sum_{n=1}^{\infty}(0{,}99)^n &= \frac{0{,}99}{1-0{,}99} = 99
\end{aligned}
$$

Ahora quizá le convenga tomar un valor de N mayor que en el caso anterior y comparar.

Ej. 5: En el ejercicio anterior lo que se hizo fue aproximar el valor de una serie por el de la suma de sus N primeros términos.

a) ¿Por qué es necesario hacer esta aproximación?

b) ¿Qué criterio adoptaría para decidir hasta qué valor de N es necesario desarrollar la sumatoria para que el resultado se aproxime lo suficiente al resultado verdadero? ¿Qué significa *aproximarse suficientemente*?

c) Implemente un algoritmo que aproxime el valor de la serie incorporando el criterio de aproximación elegido.

Ej. 6: Funciones

Además de las funciones que ya vienen incorporadas en el programa, OCTAVE/MATLAB te permite definir tus propias funciones. Una de las formas es creando un archivo **.m**, que deberá tener el mismo nombre de la función (`function [salida1,salida2,...] = nombre (arg1,arg2,...)`), a la que podemos llamar desde la línea de comandos, de la misma forma que cualquier otra orden o función ya definida en OCTAVE/MATLAB; pero también se pueden definir funciones dentro de la misma línea de comando, utilizando las llamadas funciones anónimas: `g = @(var1,var2,...) expr`.

a) Buscar en la ayuda de OCTAVE/MATLAB ejemplos de funciones definidas en una y otra forma. `help function`. `help function_handle`.

b) Defina y grafique la función

$$
f(x) = \begin{cases}
2 & \text{si} \quad x < -2 \\
x^2 & \text{si} \quad -2 \leq x < 3 \\
\frac{1}{x} & \text{si} \quad 3 \leq x \leq 10 \\
x & \text{si} \quad 15 < x < 20 \\
3 - x & \text{si} \quad x = 22 \\
0 & \text{en} \quad \text{cc}
\end{cases}
$$

los comandos `if` y `plot` pueden serle útiles.

c) Recordando la definición dada en forma recursiva del número factorial

$$
n! = \begin{cases}
1 & \text{si} \quad n = 0 \\
n(n-1)! & \text{si} \quad n \geq 1
\end{cases}
$$

definir una función que te permita calcular $n!$ para cualquier valor $n \in N$. Hacerlo como función anónima `@` y como archivo **.m**. Comparar los resultados, que obviamente tendrían que ser los mismos, y la forma de ejecutar cada una de las opciones.

d) Comparar los resultados obtenidos en **(b)** con los que se obtiene con la función propia de OCTAVE/MATLAB `factorial(N)`.