Práctica 1

1. Escribir la secuencia de comandos:

```
x=0;
while x~=10
   x=x+0.1
end
```

   en un fichero (con extensión .m) y ejecutarlo en MATLAB. Para interrumpir la ejecución, pulsar CTRL+C. ¿Qué ocurre si en lugar de incrementarse la variable en 0.1 lo hace en 0.125?. ¿Por qué?.

2. Las funciones de Bessel $J_n(x)$ satisfacen la relación de recurrencia:

$$J_{n+1}(x) = -J_{n-1}(x) + \frac{2n}{x} J_n(x),$$

   pero como las funciones de Bessel de segunda especie cumplen la misma relación y $\lim_{n\to\infty} J_n(x)/Y_n(x) = 0$, el cálculo de las funciones J_n a partir de J_0 y J_1 está mal condicionado, pues una pequeña perturbación en los datos iniciales J_0 y J_1 contamina nuestra secuencia de funciones $\{J_n\}$ con la secuencia $\{Y_n\}$, que crece más rápido con n.

   Comprobar este hecho, empezando, por ejemplo, con los valores en precisión simple $J_0(2)$ y $J_1(2)$, y aplicando la recurrencia hacia adelante hasta $n = 25$.

3. Escribir una rutina que implemente el algoritmo de Horner (**horn.m**) para la evaluación de polinomios:

   Algoritmo de Horner

   Dado el polinomio

$$P(x) = a_0 + a_1\,x + ... + a_n\,x^n\,,\ a_n \neq 0$$

   la evaluación de $P(x)$ para cierto valor $x = z$ se puede realizar en n pasos mediante

   (1) $b = a_n$
   (2) Repetir mientras $n > 0$
   (3) $n = n - 1$
   (4) $b = a_n + z * b$
   (5) Volver a **(2)**
   (6) $p(z) = b$.

   La sintaxis de la rutina puede ser:

```
>> p=horn(coefs,x);
```

   donde **p** es el valor del polinomio, **coefs** es un vector con los coeficientes del polinomio, de mayor a menor grado y x es el valor de la variable independiente. Es decir que si, por ejemplo, hacemos:

```
>> p=horn([1 -5 6 2],2);
```

entonces en la variable **p** se almacenará el valor $P(2)$ donde $P(x) = x^3 - 5x^3 + 6x + 2$.

Comparar en tiempos de CPU (comando **cputime**) el cálculo del valor de un polinomio de grado elevado en un punto utilizando el algoritmo de Horner y el cálculo directo del polinomio.

4. Es bien conocido la sucesión de números reales $\{a_n\}_{n=2}^{\infty}$ generada por el siguiente procedimeiento

$$\begin{cases} a_2 = 2\sqrt{2} \\ a_{n+1} = 2^n \sqrt{2\left(1 - \sqrt{1 - \left(\frac{a_n}{2^n}\right)^2}\right)} \end{cases}$$

converge hacia el número π.

(a) Explicar por qué falla este procedimiento cuando se utiliza en el computador para aproximar el número π.

(b) Sea

$$r_{n+1} = 2\left(1 - \sqrt{1 - \left(\frac{a_n}{2^n}\right)^2}\right), \quad n \geq 2.$$

Probar que

$$r_3 = \frac{2}{2 + \sqrt{2}} \;\text{ y }\; r_{n+1} = \frac{r_n}{2 + \sqrt{4 - r_n}} \text{ para } n \geq 3.$$

(c) Utilizar las fórmulas del apartado anterior para calcular $\{r_n\}_{n=3}^{\infty}$ y, escribiendo $a_n = 2^n\sqrt{r_{n+1}}$, obtener una aproximación de π con al menos 12 dígitos de precisión.

5. Reescribir las siguientes expresiones para evitar cancelaciones cerca de $x = 0$:

a) $\sqrt{x+1} - 1$ **b)** $\dfrac{1 - \cos x}{\operatorname{sen} x}$

c) $\dfrac{1 - \cos x}{x}$ **d)** $\dfrac{1}{1 + 2x} - \dfrac{1 - x}{1 + x}$

Comprobar numéricamente las diferencias entre las expresiones originales y las obtenidas evitando cancelaciones.

6. Escribir dos **funciones** MATLAB para la resolución de ecuaciones de segundo grado $ax^2 + bx + c = 0$, una de ellas implementando la fórmula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

y otra utilizando las expresiones alternativas:

$$x_1 = L/2a\,,\; x_2 = 2c/L$$

donde $L = -b - \text{signo}(b)\sqrt{b^2 - 4ac}$.

Comparar el funcionamiento de ambas rutinas para el cálculo de las raíces del siguiente polinomio: $P(x) = x^2 - 10^6 x + 1$.

7. Probar que la sucesión de números reales $\{y_n\}_{n=0}^{\infty}$ definida por

$$y_n = \frac{1}{\mathrm{e}} \int_0^1 x^n \exp x \, dx, \quad n \geq 0$$

puede calcularse usando la recurrencia

$$\begin{cases} y_0 = \dfrac{\mathrm{e}-1}{\mathrm{e}} \\ y_{n+1} = 1 - (n+1) y_n, \quad n \geq 0. \end{cases}$$

(a) Usar la recurrencia anterior para calcular los primeros 25 términos de la sucesión. Interpretar los resultados teniendo en cuenta que $y_n > 0$ para todo $n \geq 0$ y que $\lim_{n\to\infty} y_n = 0$.

(b) Tomar y_{50} como cero y utilizar la recurrencia hacia atrás para calcular y_{30}, y_{20}, y_{15}, y_{10}, y_5 e y_0. Explicar los resultados.