Asignatura : Informática
Grado en Ingeniería Aeroespacial - ETSIAE

Hitos semanales del primer semestre:

- **Hito 1 :** Escribir un programa que se llame `Hello_world` que abra una pantalla que y imprima "Hello world".

  Instalar el entorno Microsoft Visual Studio mediante:

  Libro en amazon.es

  PDF del libro

  La implementación de todos los hitos se encuentran en el enlace del github :

  GitHub: Juan A. Hernández Ramos

- **Hito 2 :** Raíces de una ecuación algebraica de segundo grado.

  Calcular las raíces de una ecuación algebraica de segundo grado de coeficientes reales de la forma:

  $$ax^2 + bx + c = 0. \tag{1}$$

  Para todo $a, b, c \in \mathbb{R}$. El programa debe contemplar la posibilidad de que cualquiera de los coeficientes sea cero. Los resultados deben aparecer por pantalla. Discutir el condicionamiento numérico de las soluciones en función de a, b, c.

- **Hito 3 :** Suma de series numéricas.

  1. Calcular la suma de los primeros N números naturales.
  2. Calcular la suma de los N primeros números naturales impares.
  3. Calcular el factorial de un número N. Implementar este cálculo de dos formas distintas : incrementos positivos e incrementos negativos de un bucle. Comprobar el máximo valor de N válido para diferentes tipos de dato. El número N se debe poder introducir por teclado y el resultado aparecerá por pantalla.
  4. Calcular las siguientes sumas de series las numéricas:

  $$S_4 = \sum_{n=1}^{\infty} \frac{1}{2^n}, \qquad S_5 = \sum_{n=1}^{\infty} \frac{1}{n^2}, \qquad S_6 = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2^n}, \qquad S_7 = \sum_{n=1}^{\infty} \frac{1}{n!}.$$

- **Hito 4 :** Vectores y matrices.

  Sean los vectores $V, W \in \mathbb{R}^N$ y la matriz de Vandermode $A \in \mathcal{M}_{N\times N}(\mathbb{R})$:

  $$\{v_i = 1/i^2, \quad i = 1, \dots N\} \qquad \{w_i = (-1)^{i+1}/(2i+1), \quad i = 1, \dots N\}$$

  $$\{a_{ij} = (i/N)^{j-1}, \quad i = 1, \dots N, \quad j = 1, \dots N\}.$$

  Escribir un programa que realice las operaciones siguientes y que muestre los resultados por pantalla:

  1. Sumar todas las componentes del vector V.
  2. Sumar todas las componentes de la matriz A.
  3. Sumar las componentes del vector V mayores que cero.
  4. Sumar las componentes de la matriz A menores que cero.
  5. Producto escalar de los vectores V y W.
  6. Producto escalar del vector V y la columna N de la matriz A.
  7. Multiplicar la matriz A por el vector V.
  8. Matriz traspuesta de la matriz A
  9. Obtener el valor máximo de la matriz A y su posición.

  Verificar los resultados con las funciones intrínsecas correspondientes.

- **Hito 5 :** Asignación dinámica de memoria.

  Dada la matriz de Vandermonde $A_M \in \mathcal{M}_{M\times M}(\mathbb{R})$ del hito anterior, calcular las siguientes operaciones:

  1. Suma de las trazas de A_M para matrices de diferente dimensión.

     $$S_1 = \sum_{M=1}^{10} trace(A_M)$$

  2. Suma de las trazas del cuadrado de A_M para matrices de diferente dimensión.

     $$S_2 = \sum_{M=1}^{5} trace(A_M^2)$$

  3. Traza de la suma de potencias de A_M con $M = 8$.

     $$S_3 = trace\left(\sum_{k=0}^{5} A_8^k\right)$$

  Implementar la operaciones anteriores mediante un módulo llamado **Algebra** que contenga la función **trace** que calcula la traza de una matriz, la función **Vandermonde** que crea la matriz $A_M \in \mathcal{M}_{M\times M}(\mathbb{R})$ y la función **power** que permite calcular la potencia k–ésima de cualquier matriz cuadrada.

- **Hito 6 :** Primitivas y derivadas numéricas de funciones. Funciones definidas a trozos. Impelentar la función `integral` que calcule la integral de una función genérica $f : \mathbb{R} \to \mathbb{R}$ entre $[a, b]$ y la función `derivative` que calcule la derivada de una función genérica $f : \mathbb{R} \to \mathbb{R}$ en $x = x_0$ mediante las siguientes fórmulas aproximadas:

$$\int_a^b f(x)dx \approx \sum_{k=0}^{N} f(x_k)\Delta x, \qquad \frac{df}{dx}(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h},$$

  donde $\Delta x = (b - a)/N$ y h es un valor pequeño.

  1. Validar las funciones `integral` y `derivative` mediante funciones simples con derivada e integral analítica.
  2. Implementar la subrutina `plot` que dibuje la función genérica $f : \mathbb{R} \to \mathbb{R}$ desde $x = a$ hasta a $x = b$. Esta subrutina deberá calcular la imagen de la función anterior en $M + 1$ puntos de acuerdo a la siguiente partición equiespaciada:

$$\{x_i = a + i\Delta x, \quad i = 0 \ldots M\}, \qquad \Delta x = \frac{b - a}{M}.$$

  3. Representar la función seno para $x \in [-2\pi, +2\pi]$.
  4. Implementar la función $f(x)$ definida a trozos siguiente:

$$f(x) = \begin{cases} 0, & -\infty \leq x \leq -\dfrac{\pi}{2}, \\ \cos(x), & -\dfrac{\pi}{2} < x < \dfrac{\pi}{2}, \\ 0, & \dfrac{\pi}{2} \leq x \leq +\infty. \end{cases}$$

  5. Representar la función anterior definida a trozos para $x \in [-2\pi, +2\pi]$.
  6. Representar la integral y la derivada de la función definida a trozos anterior para $x \in [-2\pi, +2\pi]$.

  Las funciones `integral`, `derivative` y `plot` deberán estar en un módulo llamado `calculus`

- **Hito 7 :** Series de potencias

  Aproximar mediante un desarrollo truncado en serie de Taylor de la forma:

  $$f(x) = \sum_{k=0}^{M} a_k \ (x - x_0)^k, \qquad\qquad a_k = \frac{f^{(k)}(x_0)}{k!},$$

  para las funciones $f : \mathbb{R} \to \mathbb{R}$, siguientes:

  1. $f(x) = e^x$,
  2. $f(x) = \sin(x)$,
  3. $f(x) = \cosh(x)$,
  4. $f(x) = \dfrac{1}{1-x}$.

  Para cada desarrollo en serie se pide:

  1. Calcular el desarrollo para un valor de M genérico ($M = 1, 2, 3, 4, 5, \ldots$) mediante un bucle desde $k = 0$ hasta $k = M$.
  2. Implementar un desarrollo para infinitos términos que para de sumar cuando se alcance la máxima precisión. Realizar la implementación mediante una sentencia de control `while`.
  3. Comparar con las funciones intrínsecas correspondientes.
  4. Discutir los resultados obtenidos.

- **Hito 8 :** Sentencias de entrada y salida.

  Crear los ficheros de datos ForTran con nombres `data1.txt` e `data2.txt` con la información siguiente:

  Contenido del fichero de entrada `data1.txt` :

```
VARIABLES=t, x, y, z, u
1.2,      3.4,     6.2,    -14.0,    0.1
-25.2,    -8.6,    5.1,     9.9,    17.0
-1.0,     -2.0,    -5.4,    -8.6,    0.0
3.14,     -11.9,   -7.0,    -12.1,   9.2
6.66,    5.32,    0.001,   0.2,     0.001
```

  Contenido del fichero de entrada `data2.txt` :

```
VARIABLES=t, x1, x2, x3, x4, x5, x6
1.2,      3.4,     6.2,    -14.0,    0.1,      4.89,   7.54
4-25.2,    -8.6,    5.1,    12.0,     9.9,      12.24,  17.0
0.0,      34.5,    -1.0,   -2.0,     -43.04,   -8.6,    0.0
3.14,     -11.9,   71.0,   7.0,     17.0,      -12.1,   9.2
 6.66,     5.32,    0.001,   0.2,     0.001,    0.008,   -0.027
54.0,     77.1,    -9.002,  -13.2,   0.017,    65.53,   -0.021
 23.04,    -51.98,  -34.2,   9.99,    5.34,     8.87,    3.22
```

  Escribir una función `load_matrix` que permita crear una matriz A con las dimensiones del fichero de datos y que almacene los valores de dicho fichero mediante asignaciones del tipo:

```
A = load_matrix('data1.txt')
A = load_matrix('data2.txt')
```

  La función `load_matrix` deberá permitir la carga de datos con cualquier formato. Una vez cargados los valores de la matriz A, crear dos vectores U y V que contengan la primera y segunda columnas de la matriz A.