Prácticas de Laboratorio

1. Aritmética de Punto Flotante

a) Realizar un programa para el cálculo de la sucesión:

```
x(0) = 1
x(1) = 1 / 3
x(n) = 13 * x(n - 1) / 3 – 4 * x(n - 2) / 3        Para n ≥ 2
```

Comparar y explicar los resultados, al ejecutar el programa para los primeros 50 valores, empleando números de precisión simple (`float`) y números de precisión doble (`double`).

b) Realizar un programa que permita conocer la forma en que los números de **punto flotante** se almacenan en la computadora, empleando la **Norma IEEE 754-1985** para números de **precisión sencilla**, en ambos sentidos: **Decimal Û IEEE**.

c) Realizar un programa para encontrar él **épsilon de la máquina**.

2. Algoritmos para búsqueda de raíces

a) Implementar los algoritmos de **Bisección, Punto Fijo, Newton Raphson** y **Secante**, para encontrar las raíces de una ecuación no lineal de una variable. Comparar los cuatro métodos, calculando el número de iteraciones necesarias para encontrar una solución para una ecuación particular.

b) Hacer un programa que permita encontrar las raíces de un polinomio por el **método de Horner**.

3. Algoritmos de Interpolación

Programar los métodos de interpolación **lineal**, de **Newton**, de **Lagrange** y de **Neville**. Discutir cual es más eficiente en cuanto al número de iteraciones y al resultado obtenido, probándolo con al menos tres tablas de valores diferentes.

4. Ajuste de Curvas

Realizar un programa para hallar la ecuación que mejor se ajuste a un conjunto de valores en forma de tabla. Emplear el **método de mínimos cuadrados** para un grado de **1** a **n**. Probar con diferentes tablas de valores, que correspondan al menos a polinomios de orden **2, 3, 4** y **5**, para comprobar su funcionalidad.

5. Derivación

Programar los algoritmos de **derivación numérica** para la primera derivada por aproximación por diferencias hacia **delante** (al menos tres fórmulas), para la primera derivada por aproximación por diferencias hacia **atrás** (al menos tres fórmulas) y para la primera derivada por aproximación por diferencias **centrales** (al menos dos fórmulas). Compare cual es la mas precisa con tres pruebas de funciones conocidas.

6. Integración

a) Programar los algoritmos de **integración simple** por la formula del **trapecio** y de **Simpson**, los métodos de **integración compuesta** del **trapecio** y de **Simpson**. Discutir sus diferencias y determinar cual es la más eficiente.

b) Programar las fórmulas de integración de **Newton-Cotes** para **N = 1** a **10 cerradas** y para **N = 1** a **6 abiertas**. Discutir cuales son mejores y en que circunstancias, aplicándolas a al menos tres integrales conocidas.

7. Álgebra lineal

a) Programar los algoritmos de **solución de ecuaciones simultáneas lineales** directos:
- Método de eliminación Gaussiana.
- Método de Gauss-Jordan.

b) Métodos iterativos:
- Método de Jacobí.
- Método de Gauss-Seidel.

Compararlos y observar su comportamiento, determinar cual es más eficiente y en que circunstancia. Probar con al menos tres sistemas de ecuaciones cada uno.

8. Solución de Ecuaciones Diferenciales Ordinarias

Programar y comparar los siguientes métodos para resolver ecuaciones diferenciales ordinarias:

a) Métodos de Taylor
- Método de Euler
- Método de Euler-Gauss

b) Algoritmos de Runge-Kutta
- Runge-Kutta de orden 2
- Runge-Kutta de orden 4
- Runge-Kutta-Fehlberg

Realizar al menos tres ejemplos en cada método.