Computación Numérica

Segunda Práctica de Fortran. Curso 2008 – 2009

Sistemas de ecuaciones lineales. Ecuaciones no lineales.

En esta práctica se propone la programación de los métodos numéricos **LU, Jacobi** y **relajación** para resolver sistemas de ecuaciones lineales y del método de **Newton** para aproximar una raíz de una ecuación escalar numérica.

1. Sistemas de ecuaciones lineales

Método LU

- Construye una subrutina para realizar la factorización LU.
- Mediante sendas subrutinas, resuelve los sistemas triangulares del método LU.
- Mediante otra subrutina, escribe en un fichero las matrices L y U y por pantalla la solución obtenida.

Jacobi y relajación

- Se pedirán por teclado: el número máximo de iteraciones y la tolerancia de error.
- Elabora dos subrutinas, una para cada algoritmo, que calculen las aproximaciones de cada método.
- La escritura de resultados en un fichero presentará los datos empleados: vector inicial, el número máximo de iteraciones y la tolerancia. Además, se mostrarán: iteración, aproximación y el error relativo asociado para cada iteración, empleando la norma 2.

Cuestiones generales

1. Construye una subrutina para leer de un fichero: n (número de ecuaciones del sistema), la matriz de coeficientes y el vector del segundo miembro.
2. El programa principal dispondrá de un menú de selección del método de resolución: LU, Jacobi o relajación.
3. Las subrutinas pueden ser externas o ir incluidas en módulos.
4. Verifica el funcionamiento de los algoritmos con ejemplos de solución conocida.

2. Ecuaciones no lineales. Método de Newton

Para aproximar una raíz de una ecuación no lineal, emplearemos el algoritmo de Newton. Para ello, introduciremos en el código la función que define la ecuación y su derivada.

- Se pedirán por teclado: la aproximación inicial, el número máximo de iteraciones y la tolerancia de error.

- La escritura de resultados presentará los datos empleados: aproximación inicial, el número máximo de iteraciones y la tolerancia. Además, se mostrará para cada iteración, la aproximación obtenida y el error relativo asociado.

Memoria
Consta de los siguientes apartados, debidamente comentados:

1. Código fuente. Listado de los programas elaborados.

2. Tests, donde figuren los diferentes tests realizados para validar los algoritmos y las soluciones exactas.

3. Tabla resumen de la ejecución de la aplicación mediante todos los métodos implementados.

Aplicación

Para calcular la temperatura (^oC) en los puntos del interior de una placa, denotada por T_{ij} en cada punto, es necesario resolver el siguiente sistema lineal:

$$\begin{pmatrix} S & -I & 0 \\ -I & S & -I \\ 0 & -I & S \end{pmatrix} \begin{pmatrix} T_{11} \\ T_{12} \\ T_{13} \\ T_{21} \\ T_{22} \\ T_{23} \\ T_{31} \\ T_{32} \\ T_{33} \end{pmatrix} = \begin{pmatrix} 75 \\ 75 \\ 175 \\ 0 \\ 0 \\ 100 \\ 50 \\ 50 \\ 150 \end{pmatrix}$$

donde

$$S = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

1. Resuelve el sistema mediante los métodos de LU, Jacobi y relajación con los valores $\omega = 1$, $\omega = 1.5$ y $\omega = 2.0$.

2. El polinomio característico de la matriz S viene dado por

   $$p(\lambda) = -\lambda^3 + 12\lambda^2 - 46\lambda + 56.$$

   Calcula los tres autovalores de la matriz mediante el algoritmo de Newton, partiendo de diferentes aproximaciones iniciales, con un error inferior a 10^{-10}.

TIEMPO DE REALIZACIÓN: 5 semanas