Computacion Científica I
Práctica final

Abril de 2011

1. Enunciado del problema

La práctica final consistirá en el análisis y solución de un problema. La práctica debe realizarse por parejas, siendo asignada a cada una de ellas uno de los dos temas propuestos: Resolución de sistemas de ecuaciones y búsqueda de raíces de un polinonio.

1.1. Sistema de ecuaciones

En este trabajo se pretende encontrar una solución adecuada a un sistema de ecuaciones, en términos de tiempo y precisión. Para ello, se pueden utilizar los métodos vistos en la teoría de la asignatura, así como cualquier otro método que los estudiantes consideren adecuado. Además, los estudiantes pueden combinar varios métodos o adaptarlos a su conveniencia.

El método debe funcionar de la siguiente manera:

- Recibe como entrada una matriz de $n \times (n+1)$ elementos, donde n representa el número de ecuaciones del sistema. Todos los sistemas a resolver tendrán el mismo número de ecuaciones como de incógnitas. La última columna de la matriz representa las soluciones de las ecuaciones.
- Devuelve en un vector los valores que deben valer cada una de las incógnitas.

De esta forma, si el sistema de ecuaciones que queremos resolver viene representado de la forma $AX = B$, donde

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

el método implementado debe recoger como parámetro de entrada una matriz de la forma

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{pmatrix}$$

y debe devolver el vector X.

Todos los sistemas presentados deben ser determinados, por lo que el algoritmo no debe estar preparado para ningún otro tipo de sistema.

1.2. Raíces en polinimios

En este trabajo se deben encontrar TODAS las raíces de un polinimio en un intervalo dado. Para ello, se pueden utilizar los métodos vistos en la teoría de la asignatura, así como cualquier otro método que los estudiantes consideren adecuado. Además, los estudiantes pueden combinar varios métodos o adaptarlos a su conveniencia.

El método debe funcionar de la siguiente manera:

- Recibe como entrada dos vectores: un vector de m componentes, donde $m-1$ es el grado del polinimio, y un vector de dos posiciones $[\underline{i}, \overline{i}]$, con $\underline{i} < \overline{i}$, representando el intervalo de búsqueda de raíces.
- Devuelve en un vector todos los puntos que constituyen una raíz del polinomio.

Por ejemplo, si queremos encontrar las ráices del polinomio $2x^2+6x+3$ en el intervalo $[-4, 6]$, el método implementado recibe como entrada los vectores [2 6 3] y $[-4\ 6]$ y devuelve el vector $[-2{,}3660\ \ -0{,}6340]$.

2. Estudio a realizar

En esta práctica se debe realizar un estudio completo del problema asignado. En este estudio se deben incluir los siguientes apartados:

- Estudio teórico: Se debe describir el problema solucionado, explicando detalladamente la solución final implementada. En este sentido, se debe especificar cómo funciona exactamente el método, así como argumentar cualquiera de las modificaciones que se hayan realizado sobre un método existente. Este estudio debe tener una extensión de entre 2 y 4 páginas.
- Estudio práctico: Se debe presentar el estudio práctico que se ha realizado sobre el problema. En él se deben incluir los siguientes aspectos:
  - Precisión/Tiempo: A la hora de resolver el problema, es necesario tener en cuenta la precisión obtenida en la solución, así como el tiempo que se tarda en obtener dicha solución. En este apartado, se debe realizar un estudio de del comportamiento del algoritmo en función de estos dos puntos de vista.

Eventualmente, se pueden comparar diferentes versiones o parametrizacione propuestas y elegir aquella que resuelva el problema compensándolos. Para ello, se pueden utilizar uno o varios métodos de los estudiados en clase, así como cualquier otro método que se crea conveniente. Además, sobre ellos se puede realizar cualquier modificación que haga obtener una mejor solución.

- Ruido gausiano: Se debe estudiar el comportamiento (robustez) del método ante la contaminación de los datos de entrada con ruido Gausiano (el ruido gausiano viene dado en función del parámetro σ). Se medirá la respuesta del algoritmo propuesto ante la contaminación con diferentes fuentes, definidas por su desviaciónt típica (σ).
- Tipos de datos: En MatLab existen diferentes tipos de datos numéricos, y dependiendo de con cual de ellos se trabaje se pueden obtener resultados más o menos precisos, pera también más o menos rápidos. En este apartado se debe realizar un estudio de como varía la precisión y el tiempo de ejecución del programa al modificar el tipo de datos. La comparación debe incluir al menos 3 tipos de datos.
- Pivoteo (sólo para el problema de sistemas de ecuaciones): A la hora de resolver un sistema de ecuaciones, la mayoría de los métodos trabajan operando entre las diferentes filas de la matriz. En este apartado, se debe hacer un estudio de cómo elegir las filas seleccionadas en cada paso, para intentar optimizar la solución (tiempo de ejecución). En este sentido, es posible que reordenar las ecuaciones del sistema (reordenar las filas de la matriz) de un modo dado, haga que la solución se obtenga en un tiempo menor, o con una precisión mayor.
- Tolerancia (sólo para el problema de raíces de polinimios): Los métodos iterativos para obtener raíces de polinomios trabajan con una cierta tolerancia. En este apartado se debe estudiar cómo varían los resultados del problema dependiendo de la tolerancia elegida. Esta tolerancia se puede expresar de dos formas: mediante cercanía de la aproximación al eje de abscisas o a realizando un estudio de la convergencia de las aproximaciones.

3. Competición

Para cada uno de los dos ejercicios se realizará una competición. En ellas, cada uno de los algoritmos implementados en MatLab deberá solucionar una cierta cantidad de problemas generados de manera aleatoria. La puntuación de cada uno de las trabajos en esta competición vendrá determinada por la combinación del error de la solución (distancia a la solución ideal) y el tiempo de ejecución. Todas las pruebas se ejecutarán al finalizar el plazo de entrega, en una de las máquinas del laboratorio. Las fórmulas de puntuación en función del tiempo y el error están bajo estudio, y se presentarán proximamente.

4. Puntuación de la práctica

Sobre un total de 10 puntos, estos se reparten de la siguiente forma:

- Estudio teórico: 2 puntos
- Estudio práctico: 5 puntos
  - Precisión / Tiempo: 2 puntos
  - Ruido Gausiano: 1 punto
  - Tipo de datos: 1 punto
  - Pivoteo / Tolerancia: 1 punto
- Competición: 3 puntos

Los tres puntos asignados a competición se repartirán de la siguiente forma. Todos los estudiantes que hayan trabajado sobre el mismo tema mostrarán su ejecución sobre un conjunto de datos aleatorio, igual para todos los grupos. Haciendo una medición referente a tiempo de ejecución y precisión obtenida, se ordenarán las soluciones obtenidas por todos los estudiantes, asignándoles los puntos correspondientes según la ecuación:

$$puntos = 3 - \frac{i-1}{N-1} \cdot 3$$

donde i es la posición obtenida por la solución y N es el número de grupos que han solucionado ese problema.

5. Material de la práctica

- Guión de la práctica.
- Reglas de formato para la memoria de prácticas.
- Funciones de MatLab para el cálculo de la puntuación de un resultado en un experimento.
- Cabecera de las funciones de MatLab que se deberán entregar para la competición.
- Datos de prueba en formato `.mat` para la realización de pruebas.