PRÁCTICAS DE CÁLCULO NUMÉRICO I

PRÁCTICA 2: resolución de ecuaciones no lineales

El objetivo de esta práctica es el desarrollo de algoritmos para la resolución de ecuaciones no lineales, implementando los métodos discutidos en clase. En particular, se desarrollarán programas para los métodos de bisección, de la secante y de Newton-Raphson, que se utilizarán para investigar su comportamiento en diversos ejemplos propuestos en las clases de problemas.

1 Programas a realizar

Las indicaciones que se dan a continuación habrán de tenerse en cuenta a la hora de elaborar los programas. Se trata de programar los métodos de bisección, de la secante y de Newton, así como de realizar una rutina para estimar el orden de convergencia de una sucesión x_n (las sucesivas aproximaciones de los métodos) que converge a un determinado valor α (la solución buscada de la ecuación no lineal). Llamaremos a estas funciones Matlab **bisec.m**, **secant.m**, **newton.m**, **orden.m**. Las funciones deben tener la siguiente sintaxis:

1. **bisect.m**: la llamada a la rutina será como sigue:

   ```
   it=bisect(a,b,epsi);
   ```

   donde $[a, b]$ es el intervalo donde se busca el cero de $f(x) = 0$ (debiéndose cumplir que $f(a)f(b) < 0$ y epsi es la precisión relativa. Tomaremos como precisión relativa

   $$\epsilon_n = (b_n - c_n)/c_n \, , \, c_n = (b_n + a_n)/2$$

   siendo $[a_n, b_n]$ los sucesivos intervalos de acotación de la raíz. Cuando se alcance un n tal que $\epsilon_n < epsi$, el algoritmo debe parar.

   La función problema, cuyo cero queremos calcular, estará definida en un fichero que llamaremos **f.m**.

   Se sugiere utilizar algún ejemplo concreto como prueba del funcionamiento del algoritmo.

   La salida del programa es un vector (**it**) que contiene los sucesivos valores de c_n generados por el algoritmo.

2. **secant.m**: la llamada será:

   ```
   it=secant(x0,x1,epsi);
   ```

   $x0$ y $x1$ son los dos valores iniciales para iniciar el método de la secante y $epsi$ es la precisión relativa con la que se quiere calcular el cero de $f(x) = 0$ (en el fichero $f.m$). Como error relativo en cada paso, tomaremos $\epsilon_n = |(x_n - x_{n-1})/x_n|$.

   La salida del programa es un vector (**it**) que contiene los sucesivos valores de x_n generados por el algoritmo.

3. **newton.m**: la llamada será:

   ```
   it=newton(x0,epsi);
   ```

$x0$ es el valor inicial para el método de Newton. Editaremos un fichero, **fn.m**, conteniendo tanto la función problema como su derivada (ambas necesarias para aplicar el método de Newton). *epsi* es el error relativo, para el que utilizaremos el mismo criterio que para el método de la secante.

La salida de la rutina es un vector (**it**) que contiene los sucesivos valores de x_n generados por el algoritmo.

4. **orden.m**

   La sintaxis ser:

```
[or,erro]=orden(it);
```

   donde **it** será una sucesión de valores (las salidas de cualquiera de los tres métodos anteriores). Este algoritmo tendrá como salida:

   (a) **erro**: si **it** es un vector de longitud n (se aconseja utilizar el comando **length** para averiguar la longitud de un vector), **erro** es el vector que almacena las diferencias $\epsilon_i \equiv \mathrm{it}(i) - \mathrm{it}(n)$ para $i = 1, ..., n$, es decir, los errores absolutos (con signo).

   (b) **or**: vector que devuelve las estimaciones del orden de convergencia en cada paso:

$$p_i = \log(|\epsilon_{i+1}/\epsilon_{i+2}|)/\log(|\epsilon_i/\epsilon_{i+1}|)\,, i = 1, ..., n-2$$

2 Actividades

1. Evaluar el cero de la ecuación $e^x \ln x + x^3 - 2 = 0$ con al menos 10 dígitos exactos utilizando el método de bisección, de la secante y de Newton. Obtener una estimación experimental del orden de convergencia utilizando el programa **orden.m** descrito en el guión de la práctica. ¿Coincide con lo esperado?.

2. Considerar el cálculo del cero de la ecuación $x + e^{-Bx^2} cos(x) = 0$ con 8 dígitos exactos para los valores $B = 5, 9$; utilizar el método de Newton y el valor inicial $x_0 = 0$. Construir un fichero **graf.m** que proporcione como salida la representación del vector **erro** generado al ejecutar

```
>> it=Newton(0,1.d-8);
>> [or,erro]=orden(it);
```

   Discutir los resultados obtenidos. ¿Qué ocurre al tomar $B = 10$?; ¿por qué?.

3. Utilizar el método de Newton para evaluar el cero positivo de la función

$$f(x) = (1 + \Delta)x - \arcsin(x)$$

   para valores $0 < \Delta << 1$. Probar por ejemplo, con $\Delta = 10^{-3}, 10^{-5}, 10^{-9}$ y precisiones relativas $\epsilon = 1.e-6, 1.e-10, 1.e-14$. Explicar lo que ocurre y considerar alguna mejora para evaluar $f(x)$ cerca de su cero de forma más precisa.