UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMÁTICA
Viernes 11/10, 2019

Métodos Numéricos 220138: Tarea 2

Fecha de entrega: Viernes 18/10, 2019.

Complete los problemas 1–5. Los problemas de desarrollo serán entregados en clase o enviados al correo pvenegas@ubiobio.cl. Enviar por correo los programas utilizados para resolver los problemas de Matlab.

Problema de desarrollo.

Problema 1: Encontrar un intervalo $[a,\, b]$, con $a, b \in \mathbb{N}$, en el cual se pueda aplicar el método de bisección para encontrar las soluciones de la ecuación $e^x - x^2 + 3x - 2 = 0$. Estimar la cantidad de iteraciones necesarias para obtener un error menor que 10^{-9} y ejecute 2 iteraciones.

Problema 2:

1. Mostrar que, calcular las raices de $f(x) = \dfrac{x^2 - 4x - 5}{x - 4}$ es equivalente a resolver el siguiente esquema de punto fijo
$$x_{n+1} = \frac{5}{x_n - 4} \qquad n = 1, 2, \ldots$$
2. Encuentre un intervalo $[a, b]$ donde se garantice la existencia del punto fijo. Ejecutar 2 iteraciones del método.

Problema de Matlab.

Problema 3: Haga un programa que calcule la raíz de una ecuación $f(x) = 0$ mediante el *método de bisección*. Los datos del programa deben ser la función f, los extremos del intervalo $[a, b]$ donde se busca la raíz, la tolerancia `tol` con la que se desea calcular ésta ($|x_{k+1} - x_k| < \texttt{tol}$) y la cantidad máxima de iteraciones a ejecutar (`maxiter`). El programa debe tener una salida de error en el caso en que la función f no cambie de signo en el intervalo inicial. (*Indicación:* utilizar como base el programa `biseccion`.)

Considere la siguiente función no lineal:

$$f(x) = \sqrt{x} - 3\,(\ln(x)\cos(x)) - \frac{1}{2}. \tag{1}$$

3.1) Escriba un programa (*function*) en MATLAB para evaluar la función dada $f(x)$.

3.2) Determine el número de raíces de la ecuación $f(x) = 0$, para $x \in [0, 40]$. Describir el procedimiento y comandos que utilizó para determinar las raices.

3.3) Use el Método de **Bisección** para aproximar todas las raíces de la ecuación no lineal dada en el intervalo $[30, 40]$, con una tolerancia de 10^{-6} y un número máximo de iteraciones de 400. Escriba: el intervalo inicial considerado para la busqueda de cada una de las raices, todas las raíces determinadas, cantidad de iteraciones y los comandos que usó.

3.4) Use el Método de **Newton** para aproximar todas las raíces de la ecuación no lineal dada en el intervalo $[15, 25]$, con una tolerancia de 10^{-6} y un número máximo de iteraciones de 400. Escriba: todas las raíces determinadas, cantidad de iteraciones, la aproximación inicial para cada raíz y los comandos que usó.

3.5) Considere la función

$$f(x) = x^2/10 - 30\,(\ln(x)\cos(x)) - 1. \tag{2}$$

Encuentre 2 raices de f con los métodos de bisección y Newton con una tolerancia de 10^{-10} y un número máximo de iteraciones de 800. Escriba: todas la raíces determinadas, cantidad de iteraciones, intervalo inicial (bisección) la aproximación inicial (Newton) para cada raíz y los comandos que usó.

Problema 4: Haga un programa que calcule la raíz de una ecuación $f(x) = 0$ mediante el *método de la secante*. Los datos del programa deben ser la función f, las approximaciones iniciales (x_0, x_1), la tolerancia `tol` con la que se desea calcular ésta ($|x_{k+1} - x_k| <$ `tol`) y la cantidad máxima de iteraciones a ejecutar (`maxiter`). (*Indicación:* utilizar como base el programa `newton`.)

4.1) Use el Método de la **Secante** para aproximar todas las raíces de la ecuación no lineal $f(x)$ en el intervalo $[30, 40]$ (f dada en (1)), con una tolerancia de 10^{-6} y un número máximo de iteraciones de 400. Escriba: todas las raíces determinadas, cantidad de iteraciones, las aproximaciónes iniciales (x_0, x_1) para cada raíz y los comandos que usó.

4.2) Hacer un programa que ejecute los primeros 20 pasos de los métodos de bisección, Newton-Raphson y secante para calcular $\sqrt[3]{2}$, comenzando con valores iniciales apropiados. Graficar simultáneamente las tres sucesiones obtenidas.

4.3) Se quiere resolver la ecuación $f(x) = 0$, donde $f(x) = e^x - 2$. Calcular los 15 primeros términos de las sucesiones generadas por los métodos de bisección, Newton y de la secante, comenzando con un intervalo adecuado para el primer metodo, los valores iniciales $x_0 = 3$ para el segundo método, y $x_0 = 3$, $x_1 = 2.3$ para el último. Graficar simultáneamente las tres sucesiones obtenidas.

Problema 5:. Haga un programa que calcule el punto fijo de una función mediante el *método de punto fijo*. Los datos del programa deben ser la función g, las approximación inicial x_0, la tolerancia `tol` con la que se desea calcular ésta ($|x_{k+1} - x_k| <$ `tol`) y la cantidad máxima de iteraciones a ejecutar `maxiter`. (*Indicación:* utilizar como base el programa `ptofijo`.)

5.1) Considere las siguientes funciones:

$$i)\quad g(x) = \frac{5}{x-4} \qquad\qquad ii)\quad g(x) = \sqrt{\frac{e^{-x}}{3}}$$

Para cada función encontrar un intervalo $[a, b]$ en el cual exista un punto fijo. Para esto, graficar $g(x)$ y la función $f(x) = x$ (en una figura) y buscar el punto de intersección. (*Indicación:* utilizar el comando `fplot`.)

5.2) Aplicar el programa para encontrar puntos fijos de las funciones del problema anterior considerando una tolerancia de 10^{-7} y un número máximo de iteraciones de 500. Escriba: la aproximación inicial, los puntos fijos determinados, cantidad de iteraciones y los comandos que usó.

Método de Newton

```
function [raiz,iter]=newton(f,Df,x0,tol,maxit)
k=0;
raiz=x0;
corr=tol+1;
while (k<maxit) & (abs(corr)>tol)
    k=k+1;
    xk=raiz;
    fxk=f(xk);
    Dfxk=Df(xk);
    if (Dfxk==0)
        error('La derivada de la funcion se anula.')
    end
    corr=fxk/Dfxk;
    raiz=xk-corr;
    iter=k;
end

if (abs(corr)>tol)
    error('Se excedio el numero maximo de iteraciones.')
end
```

Método de Bisección

```
function p = biseccion(f,a,b)
if f(a)*f(b)>0
    disp('No hay cambio de signo')
else
    p = (a + b)/2;
    err = abs(f(p));
    while err > 1e-7
   if f(a)*f(p)<0
       b = p;
   else
       a = p;
   end
    p = (a + b)/2;
   err = abs(f(p));
    end
end
```

Método de Punto Fijo

```
function x0 = ptofijo(f,a,b)
for i=1:100
xi=g(x0);
x0=xi;
end
```