UNIVERSIDAD DE CONCEPCION
FACULTAD DE CIENCIAS
FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA MATEMATICA

Cálculo Numérico (521230)

Laboratorio 8

Ecuaciones no lineales con raíces múltiples

1. La siguiente función tiene raices múltiples :

$$
\begin{aligned}
F(x) &= 823543\,x^7 - 18117946\,x^6 + 170826348\,x^5 - 894804680\,x^4 \\
&\quad + 2812243280\,x^3 - 5303087328\,x^2 + 5555615296\,x - 2494357888
\end{aligned}
$$

El método de Newton-Raphson para calcular la raíz de $f(x) = \sqrt[p]{F(x)}$, donde p es la multiplicidad de la raíz de $F(x)$, está dado por :

$$
x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{(F(x_k))^{\frac{1}{p}}}{\frac{1}{p}(F(x_k))^{\frac{1}{p}-1}F'(x_k)} = x_k - p\frac{F(x_k)}{F'(x_k)}
$$

(a) Escriba el siguiente programa (o bájelo de la página web del curso, o solicítelo al ayudante) para el cálculo de la raíz de $f(x) = \sqrt[p]{F(x)}$ mediante el *método de Newton-Raphson modificado* en términos del parámetro p :

```
function [raiz,k]=newtonm(f,Df,x0,p,tol,maxit)
k=0;
raiz=x0;
corr=tol+1;
while (k<maxit) & (abs(corr)>tol)
    k=k+1;
    xk=raiz;
    fxk=feval(f,xk);
    Dfxk=feval(Df,xk);
    if (abs(fxk)<eps)&(Dfxk==0)
        corr=0;
    else
        if (Dfxk==0)
            error('La derivada de la funcion se anula.')
        end
        corr=fxk/Dfxk;
        raiz=xk-p*corr;
    end
end
if (abs(corr)>tol)
    error('Se excedio el numero maximo de iteraciones.')
end
```

(b) Pruebe con $p = 1, 2, 3, 4, 5, 6, 7, 8, 9 \ldots, 14$ ¿para que valor de p de todos los testeados, el algoritmo utiliza el menor número de iteraciones ? deduzca la multiplicidad de la raíz de F.

(c) **Ejercicio optativo :** haga lo mismo con la función

$$
G(x) = \frac{64}{125} - \frac{48}{25}\,x\sin(x) + \frac{12}{5}\,x^2\,(\sin(x))^2 - x^3\,(\sin(x))^3
$$

Ecuaciones Diferenciales Ordinarias

El objetivo de este laboratorio es aprender técnicas para la resolución numérica de problemas de valores iniciales (P.V.I.) para ecuaciones diferenciales ordinarias (E.D.O.) y sistemas de E.D.O.

MATLAB tiene varios comandos para la resolución numérica de P.V.I. para E.D.O.:

```
Ordinary differential equation solvers.
(If unsure about stiffness, try ODE45 first, then ODE15S.)
  ode45    - Solve non-stiff differential equations, medium order method.
  ode23    - Solve non-stiff differential equations, low order method.
  ode113   - Solve non-stiff differential equations, variable order method.
  ode23t   - Solve moderately stiff differential equations, trapezoidal rule.
  ode15s   - Solve stiff differential equations, variable order method.
  ode23s   - Solve stiff differential equations, low order method.
  ode23tb  - Solve stiff differential equations, low order method.
```

Como se ve en esta lista, hay métodos para resolver E.D.O. *stiff* y no *stiff*. Además hay métodos de orden bajo, medio, alto y variable.

Todos ellos tienen una sintaxis semejante. Por ejemplo, para resolver el P.V.I.

$$\begin{cases} y' = f(t, y), \\ y(t_o) = y_o, \end{cases}$$

en el intervalo $[t_o, t_f]$ mediante el comando `ode45` en su opción más sencilla, debe ejecutarse:

```
[t,y]=ode45('f',[to tf],yo);
```

donde:

- `f` es el nombre de la función $f(t, y)$ (típicamente definida mediante un programa *function* en un archivo `f.m`);
- `to` y `tf` son los extremos del intervalo donde se desea conocer la solución;
- `yo` es el valor de la solución en `to` (es decir el valor de la condición inicial $y(t_o) = y_o$);
- `t` devuelve los valores de la variable independiente t donde el método calcula el valor de la solución;
- `y` devuelve los valores de la solución en cada uno de los puntos t.

Estos comandos no requieren como dato un paso de integración h pues todos ellos determinan de manera automática en cada paso k, el tamaño del paso de integración h_k necesario para mantener los errores por debajo de una tolerancia determinada. Los valores de t que entrega corresponden a los puntos $t_k = t_{k-1} + h_k$, $k = 1, 2, \ldots$, en los que el comando necesitó calcular el valor de $y(t_k)$.

Si se desea conocer la solución para ciertos valores de t, puede alternativamente ejecutarse:

```
[t,y]=ode45('f',tspan,yo);
```

donde `tspan` es el vector de valores donde se desea conocer la solución. Por ejemplo, `tspan=0:0.1:1`. En ese caso, la salida `t` coincide con `tspan` e `y` contiene los valores de la solución en esos puntos.

La tolerancia predeterminada de estos métodos es 10^{-3}, para el error relativo, y 10^{-6}, para el error absoluto. Si se desea calcular la solución con otras tolerancias, deben prefijarse las opciones elegidas mediante el comando `odeset`. Además, en la ejecución del comando para resolver la E.D.O., debe agregarse el parámetro adicional de opciones. La sintaxis para realizar esto es, por ejemplo:

```
options=odeset('RelTol',1e-6,'AbsTol',1.e-8);
[t,y]=ode45('f',[to tf],yo,options);
```

Si se ejecuta `options=odeset('RelTol',1e-6,'AbsTol',1.e-8)` sin el ";" puede verse que hay otras opciones que pueden prefijarse, además de las tolerancias de los errores.

Por ejemplo, si se desea resolver el P.V.I.

$$\begin{cases} y' = y, \\ y(0) = 1, \end{cases}$$

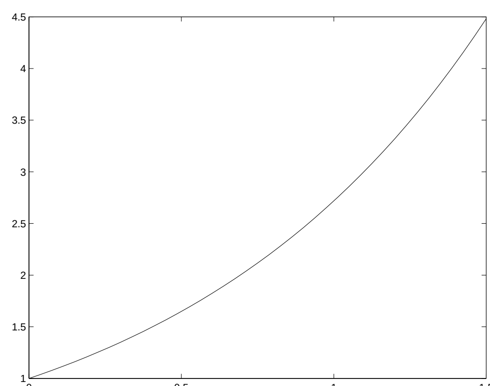
en el intervalo $[0, 1.5]$, mediante el comando `ode45` y visualizar la solución obtenida, debe crearse un fichero `f.m` como sigue:

```
function z=f(t,y)
z=y;
```

y ejecutarse:

```
[t,y]=ode45('f',[0 1.5],1);
plot(t,y)
```

Así se obtiene la siguiente gráfica:



El siguiente ejemplo resuelve la misma ecuación en los puntos `t=0:0.1:1.5`, con error absoluto menor a 10^{-6} y calcula los errores cometidos restando los valores calulados a los de la solución verdadera, que en este caso es $y(t) = \mathrm{e}^t$:

```
options=odeset('AbsTol',1.e-6);
tspan=0:.1:1.5;
[t,y]=ode45('f',tspan,1,options);
error=exp(t)-y
error =
   1.0e-06 *
         0
   -0.0003
   -0.0248
   -0.0448
   -0.0076
   -0.0415
   -0.0694
   -0.0200
   -0.0669
   -0.1056
   -0.0402
   -0.1048
   -0.1586
   -0.0721
   -0.1612
   -0.0989
```

La salida que se presenta indica que los errores son efectivamente menores en valor absoluto a 10^{-6}.

La resolución de P.V.I. para sistemas de E.D.O. se realiza mediante los mismos comandos. En tal caso, `f(t,y)` debe ser una función a valores vectoriales (es decir un vector columna de funciones) e `y` un vector columna de variables de la misma dimensión. Además, la condición inicial `yo` también debe ser un vector columna de la misma dimensión.

Por ejemplo, consideremos el P.V.I.

$$\begin{cases} x' = y, & x(0) = 1, \\ y' = -x, & y(0) = 0, \end{cases}$$

cuya solución exacta es

$$\begin{cases} x = \cos t, \\ y = \operatorname{sen} t. \end{cases}$$

Por lo tanto los puntos $(x(t), y(t))$ solución de este sistema de E.D.O, describen la circunferencia unitaria.

Este sistema escrito vectorialmente resulta:

$$\begin{cases} Y' = F(t, Y), \\ Y(0) = Y_o, \end{cases} \quad \text{con} \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(t, Y) = \begin{pmatrix} y \\ -x \end{pmatrix} = \begin{pmatrix} Y_2 \\ -Y_1 \end{pmatrix} \quad \text{e} \quad Y_o = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

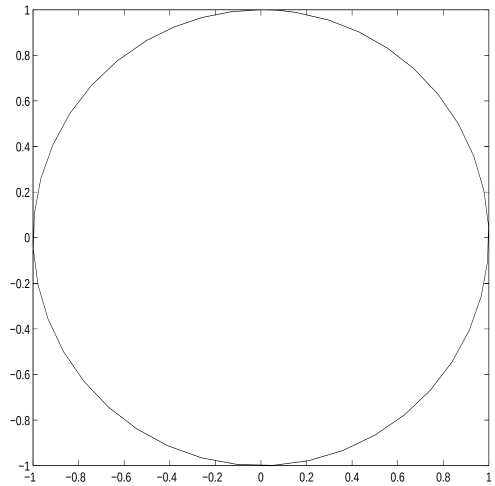
Para resolverlo debe crearse un fichero `F.m` como sigue:

```
function Z=F(t,Y)
Z=[Y(2);-Y(1)];
```

Los siguientes comandos resuelven este P.V.I. en el intervalo $[0, 2\pi]$ y grafican la curva $(x(t), y(t))$, para $0 \leq t \leq 2\pi$, que se obtiene:

```
[t,Y]=ode45('F',[0 2*pi],[1;0]);
plot(Y(:,1),Y(:,2));
```

Así se obtiene la siguiente gráfica:



2. El desplazamiento u de la posición de equilibrio de una masa m, sujeta a un resorte de constante k, inmersa en un medio viscoso que ejerce una resistencia al movimiento cu' (es decir, proporcional a la velocidad de la masa $u' = du/dt$) y sobre la que se ejerce una fuerza f, se modela mediante la E.D.O.

$$\begin{cases} mu'' + cu' + ku = f, \\ u(0) = u_0, \quad u'(0) = v_0, \end{cases}$$

donde u_0 y v_0 son el desplazamiento y la velocidad de la masa en el instante inicial $t = 0$.

(a) Hacer un cambio de variable para convertir esta E.D.O. de segundo orden en un sistema de 2×2 de primer orden

(b) Calcular y graficar el desplazamiento de la masa a lo largo de 1 minuto para los siguientes datos:

$$m = 1.2\,\mathrm{kg}, \qquad k = 15\,\frac{\mathrm{kg}}{\mathrm{s}^2}, \qquad c = 0.3\,\frac{\mathrm{kg}}{\mathrm{s}},$$

en los siguientes casos:

i. cuando la masa se desplaza $1\,\mathrm{m}$ de su posición de equilibrio y, desde el reposo, se la deja oscilar libremente;

ii. cuando desde su posición de equilibrio y en reposo, se le aplica una fuerza externa periódica $f(t) = \cos t$ (con el tiempo t medido en segundos y la fuerza f en $\mathrm{kg\,m/s^2}$);

iii. igual que el anterior pero con $f(t) = \cos \omega t$, donde $\omega = \sqrt{\frac{k}{m}}$ es la frecuencia de oscilación libre del sistema sin amortiguamiento.

MCP/RRS/GBG/MSC
http://www.ing-mat.udec.cl/pregrado/asignaturas/521230/
22/10/03