Universidad Rey Juan Carlos

SEMINARIOS REALIZADOS EN AULAS DE INFORMÁTICA CON OCTAVE/MATLAB

MÉTODOS MATEMÁTICOS APLICADOS A LA INGENIERÍA

Asignaturas:
Métodos matemáticos aplicados a la Ing. de Materiales
Métodos numéricos en el Máster en Ing. Industrial

A. I. Muñoz Montalvo
Septiembre 2022

Publicado en: https://burjcdigital.urjc.es

http://hdl.handle.net/10115/20113

EJERCICIOS PROPUESTOS Y SOLUCIONES

Observaciones: Toda línea que comience por % indicará un comentario y lo que siga al símbolo prompt del sistema, >, es una línea de comandos.

Los códigos utilizados han sido creados por los profesores A.I. Muñoz, A. Nolla y E. Schiavi y están publicados en https://burjcdigital.urjc.es, siendo la mayor parte de ellos, adaptaciones de las funciones del libro *"Cálculo científico con MATLAB y Octave"* de A. Quarteroni, F. Saleri, que se pueden obtener en https://mox.polimi.it/qs/.

SEMINARIO 1.

1. Sea dada la función
$$f(x) = x^3 + 4x^2 - 10,$$
   (a) Aplicar el algoritmo de bisección para calcular todas las soluciones de la ecuación $f(x) = 0$, trabajando con una tolerancia de 10^{-3} y un número máximo de 100 iteraciones.
   (b) Aplicar el método de Newton para resolver la misma ecuación con los mismos valores de tolerancia y número máximo de iteraciones.

Solución. Escribimos en un script el siguiente texto:

% Apartado a)

```
> fecu=@(x) x.^ 3+4.*x.^ 2-10;
> a=0;b=2;maxitera=100;errorper=1e-3;
```

% Comprobamos que hemos elegido bien los extremos del intervalo:

```
> fecu(a); fecu(b)
```

% y vemos que toman signos distintos, -10 y 14.

% Hacemos la llamada al código que resuelve el problema por bisección,
% metbiseccion.m:

> `[sol,itera]=metbiseccion(fecu,a,b,errorper,maxitera);`

% Apartado b). A continuación resolvemos el problema por el método de Newton-Raphson:

% sólo queda por definir:

> `dfecu=@(x) 3.*x.^ 2+8.*x; x0=1;`

% y llamamos al código metnewton1ecu.m

> `[sol2,itera2]=metnewton1ecu(fecu,dfecu,x0,...`

> `errorper,maxitera)`

%Fin del script.

Las soluciones obtenidas son: sol=1.3643, itera=10, con el método de bisección y sol2=1.352, itera2=4, con el método de Newton.

2. Sea dada la función

$$f(x) = e^x - 3x^2.$$

   (a) Considerando el esquema numérico asociado al método de Newton como un esquema de punto fijo, definir la función $g(x)$ asociada al método y utilizar el algoritmo de punto fijo para calcular las soluciones de la ecuación $f(x) = 0$, trabajando con una tolerancia de 10^{-6} y un número máximo de 1000 iteraciones.

Solución. Escribimos en un script el siguiente texto:

> `f=@(x) exp(x)-3.*x.^ 2;`

% Comprobamos que en el intervalo [0,1] hay una solución

> `a=0;b=1; fun(a),fun(b)`

% fun(a)=1, fun(b)=-0.2817

% En efecto, la función toma signos distintos en 0 y en 1: 1 y -0.28172, resp.

%Definimos g(x)=x-f/f'

```
> g=@(x) x-(exp(x)-3.*x.^ 2).*(exp(x)-6.*x).^ (-1);
```

```
> x0=0.5;errorper=1e-6;maxitera=1000;
```

% Hacemos la llamada al código metpuntofijo.m

```
> [x,itera]=metpuntofijo(g,x0,errorper,maxitera)
```

La solución obtenida es: x=0.91001 e itera=5.

3. Resolver el sistema de ecuaciones no lineales:

$$\begin{cases} \ln(x^2+y^2) - \sin(xy) - (\ln(2) + \ln(\pi)) = 0 \\ e^{x-y} + \cos(xy) = 0. \end{cases}$$

utilizando el método de Newton y trabajando con una tolerancia de 10^{-6}. Utilizar como semilla $(x_0, y_0) = (2, 2)$.

Solución.

Primero definimos los scripts con las funciones fecusistema.m y jacobiana.m:

```
> function F=fecusistema(x,y)

> F(1,1)=log(x.^ 2+y.^ 2)-sin(x.*y)-(log(2)+log(pi));
> F(2,1)=exp(x-y)+cos(x.*y);
> end

> function J=jacobiana(x,y)
> J(1,1)=2.*x.*(x.^ 2+y.^ 2).^ (-1)-y.*cos(x.*y);
> J(1,2)=2.*y.*(x.^ 2+y.^ 2).^ (-1)-x.*cos(x.*y);
> J(2,1)=exp(x-y)-y.*sin(x.*y);
> J(2,2)=-exp(x-y)-x.*sin(x.*y);
> end
```

A continuación, llamamos al código metnewtonsistemas.m, para ellos escribimos un nuevo script con los comandos:

```
> vectorx0=[2;2]; errorper=1e-6;maxitera=100;
> [x,res,niter]=metnewtonsistema(@fecusistema,...
```

> `@jacobiana,vectorx0,errorper,maxitera)`

La solución obtenida es: x=1.77245, y=1.77245, itera=6.

SEMINARIO 2.

1. Se considera el problema de valor inicial:

$$\begin{cases} y^{'} = y - \sin(t) + \cos(\mathrm{t}), \\ y(0) = 1. \end{cases}$$

En el intervalo temporal $[0, 2]$, utilizar los esquemas numéricos de Euler explícito, Euler implícito y Crank-Nicolson para obtener el valor aproximado de la solución en $t = 1$ tomando un paso de discretización $h = 0.1$. Sabiendo que la solución exacta es $y(t) = e^t + sin(t)$, dibujar las soluciones obtenidas con los tres métodos y la solución exacta en un mismo plot. Comparar los resultados obtenidos. Escribir los valores de las distintas soluciones en $t = 0.5$ y $t = 1.5$. .

Solución. Abrimos un nuevo script para escribir el siguiente código y luego ejecutarlo:

> `f=@(t,y) y-sin(t)+cos(t);`

> `intiempo=[0 2]; valorini=1;npasos=20;`

% Hacemos una llamada a los códigos correspondientes:

> `[solt,solye]=eulerexpicito(f,intiempo,valorini,npasos);`

> `[solt,solyi]=eulerimplicito(f,intiempo,valorini,npasos);`

> `[solt,solyc]=cranknicolson(f,intiempo,valorini,npasos);`

% Observa que llamamos de forma distinta a los resultados para y.

> `solexac=@(t) exp(t)+sin(t); figure;`

% Dibujamos las soluciones junto con los valores de la solución exacta para comparar

> `plot(solt,solye,'ro',solt,solyi,'b+',...`

> `solt,solyc,'g*',solt,solexac(solt),'k^')`

> `t1=min(find(soluciont>=0.5));`

% también t1=find(soluciont==0.5);

> `t2=min(find(soluciont>=1.5));`

% también t2=find(solución t==1.5);

> `u1(t1),u2(t1),u3(t1), u1(t2),u2(t2),u3(t2)`

Las soluciones son: para t=0.5, con EE 2.096, con EI 2.1646 y con CN 2.1283. Para t=1.5, con EE 5.2516, con EI 5.7585 y con CN, 5.4825.

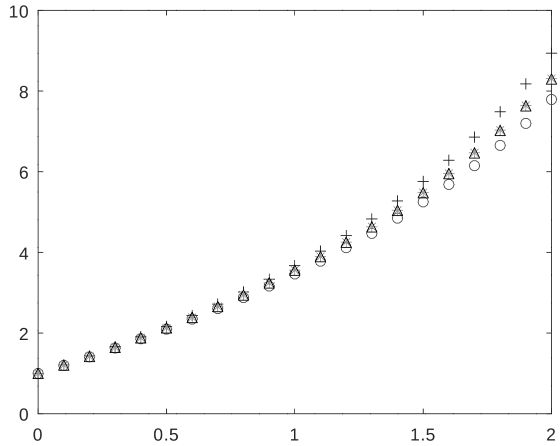

Figure 1: Resultados obtenidos con los 3 métodos junto con la exacta (en círculos rojos EE, en + azules EI, en * verdes CN, y la exacta en triángulos negros).

2. Se considera el problema de valor inicial:

$$\begin{cases} y' = -2ty^2, \\ y(0) = 1. \end{cases}$$

Aplicar los métodos tipo Runge-Kutta dados por los algoritmos de Heun (heun.m) y Simpson (rungekuttao3.m), para resolver el PVI anterior en el intervalo temporal [0,1], tomando como paso de discretización $h = 0.01$.

Escribir los valores obtenidos con los dos métodos para los tiempos $t = 0.1$, $t = 0.5$ y $t = 1$.

Solución. Abrimos un script con el siguiente código :

```
> f=@(t,y) -2.*t.*y.^ 2; intiempo=[0 1];

> npasos=100;valorini=1;

> [solt,soly1]=heun(f,intiempo,valorin,npasos);

> [solt,soly2]=rungekuttao3(f,intiempo,valorin,npasos);
```

% Dibujamos los resultados obtenidos con los dos métodos:

```
> plot(solt,soly1,'ro',solt,soly2,'b*')
```

% Calculamos los valores numéricos en los tiempos pedidos:

```
> t1=min(find(tt>=0.1));
> t2= min(find(tt>=0.5));
> t3=min(find(tt>=1));
> u2(t1),u2(t2),u2(t3) u3(t1),u3(t2),u3(t3)
```

Los resultados obtenidos con Heun son: en t=0.1, 0.9954, en t=0.5, 0.8518, en t=1, 0.5174. Los resultados obtenidos con Simpson son: en t=0.1, 0.9901, en t=0.5, 0.8006, en t=1, 0.5014.

SEMINARIO 3.

1. Sea dado el Problema de Transporte Difusivo y Convectivo definido por el PVC:

$$\begin{cases} -u^{''} - 4u^{'} = -16x^3 + 34x - 1, \quad \forall x \in (0, 2), \\ u(0) = 4, \quad u(2) = 2, \end{cases}$$

cuya solución analítica es:

$$sol(x) = x^4 - x^3 - 3.5x^2 + 2x + 4,$$

siendo u la concentración de un contaminante.

(a) Aplicar el algoritmo bvpdirichlet.m para calcular la solución en el intervalo [0, 2] con paso de discretización $h = 0.125$.

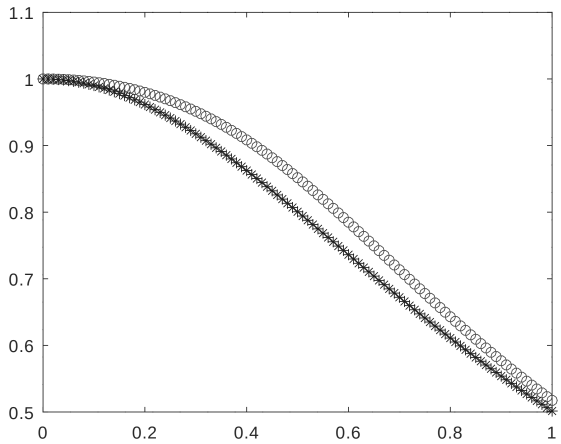

Figure 2: Resultados obtenidos con el método de Heun en rojo y con Simpson en azul (queda por encima la solución obtenida con Heun).

(b) Dibujar la solución analítica junto con la solución numérica.

(c) Determinar los valores máximos y mínimos de contaminación así como su localización utilizando la solución numérica y la analítica. Calcular los errores cometidos.

(d) Determinar la región máxima de seguridad (donde la concentración del contaminante es menor que uno, es decir, $u(x) < 1$). Utilizar la solución analítica.

Solución Creamos un script con el siguiente código para luego ejecutarlo:

```
> a=0; b=2; D=1; V=-4; Q=0;

> f=@(x) -16.*x.^ 3+34.*x-1; ua=4; ub=2; N=15;

> [xh,uh]=bvpdirichlet(a,b,N,D,V,Q,f,ua,ub);

> solexac=xh.^ 4-xh.^ 3-3.5.*xh.^ 2+2.*xh+4;

> figure; plot(xh,uh,'r',xh,solexac,'g')

> nmax=max(uh),nmin=min(uh)

> emin=min(solexac),emax=max(solexac)
```

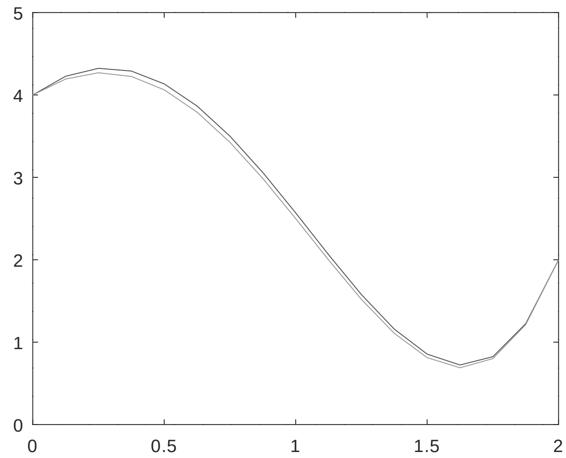

Figure 3: Resultados obtenidos con el método numérico en rojo, y la solución exacta en verde.

Los resultados son: nmax=4.3235, nmin=0.7233, emin=0.6897, emax=4.2695.

% Dibujamos el error cometido:

`> figure; err=uh-solexac; plot(xh,err)`

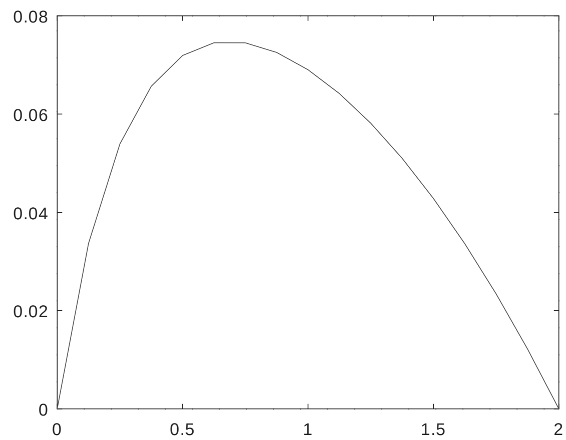

Figure 4: Error cometido.

% Vemos dónde se alcanzan el valor máximo y el mínimo

`> xm=find(uh>=nmax), xmin=find(uh<=nmin), xh(xm),xh(xmin)`

Los valores obtenidos (componentes) son: xm=3 y xmin=14. Por tanto, el valor máximo se alcanza en x=0.25, y el valor mínimo en x=1.6250.

2. Sea dado el PVIC:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \\ u(x,t) = g(x,t), \quad x = 0, \quad x = 1, \\ u(x,0) = u_0(x), \quad x \in (0,1) \end{cases}$$

donde

$$f(x,t) = 2t - x^2 + 10xt,$$

$$cc(x,t) = x^2(x-1),$$

$$u_0(x) = 0.$$

Aplicar el algoritmo de ecucalor.m para calcular la solución en el intervalo temporal [0,1], con paso de discretización espacial, $\Delta x = 0.05$ y de discretización temporal, $\Delta t = 0.02$. Dibujar la solución obtenida para $\theta = 0$ (Euler Explícito). ¿Puedes justificar la gráfica obtenida ? Determinar el paso de discretización temporal máximo que asegura estabilidad y calcular la solución para ese paso temporal. ¿Cuánto vale la solución en el punto $x = 0.85$?

Solución. Abrimos un nuevo script con el siguiente código:

```
> intespacio=[0 1]; intiempo=[0 1];

> pasosespacio=20,pasostiempo=100;

> theta=0; c=1; u0=@(x) 0.*x;

> cc=@(t,x) (x.^2).*(x-1);

> f=@(t,x) 2.*t-x.^ 2+10.*x.*t;

> [x,u]=ecucalor(intespacio,intiempo,pasosespacio,...

> pasostiempo,theta,c,u0,cc,f);

> figure; plot(x,u)
```

% Se puede observar que el plot no aparece puesto que hay valores
% que son del orden de 1e+11

% Esto se debe a la inestabilidad asociada al método explícito

% Para que sea inestable, debemos reducir el paso temporal

> x1=min(find(x>=0.85)); u(x1)

% Se tiene que x1=18 y u(18)=3.6504e+11

> dx=0.05; dt=(0.05)^ 2; pasost=1/(0.05)^ 2 ;

> pasosespacio=20; pasosespacio=800;

> [x,u1]=ecucalor(intespacio,intiempo,pasosespacio,pasostiempo,...

> theta,c,u0,cc,f);

> figure;

> plot(x,u1), x1=min(find(x>=0.85)), u1(x1)

El resultado obtenido es u1(18)=0.4496.

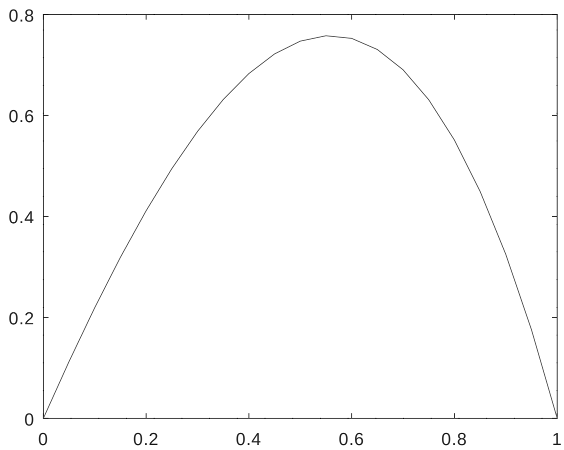

Figure 5: Resultados obtenidos cumpliendo el criterio de estabilidad.

EJERCICIOS QUE DEBE REALIZAR EL ALUMNO

Y ENTREGAR A LA PROFESORA

1. Sea dada la función
$$f(x) = e^x - 15 - arctg(x).$$
   (a) Aplicar el algoritmo de bisección para calcular la solución de la ecuación $f(x) = 0$, trabajando con una tolerancia de 10^{-3} y un número máximo de 100 iteraciones.

   (b) Aplicar el método de Newton para resolver la misma ecuación con los mismos valores de tolerancia y número máximo de iteraciones.

2. Sea dada la función
$$f(x) = \sqrt{x} sin(x) - x^3 + 2.$$
   (a) Considerando el esquema numérico asociado al método de Newton como un esquema de punto fijo, definir la función $g(x)$ asociada al método y utilizar el algoritmo de punto fijo para calcular la solución de la ecuación $f(x) = 0$, trabajando con una tolerancia de 10^{-6} y un número máximo de 1000 iteraciones.

   (b) Define un esquema de punto fijo diferente al considerado en el apartado anterior justificando la elección, para calcular la solución de la ecuación $f(x) = 0$ trabajando con una tolerancia de 10^{-6} y un número máximo de 1000 iteraciones.

3. Resolver el sistema de ecuaciones no lineales:
$$\left\{ \begin{array}{l} -x - y^3 + x^2 - 2 = 0, \\ 1 - 2x + y - y^2 = 0. \end{array} \right.$$
   utilizando el método de Newton y trabajando con una tolerancia de 10^{-6}. Utilizar como semilla $(x_0, y_0) = (-0.8, -0.4)$.

4. Se considera el problema de valor inicial:
$$\left\{ \begin{array}{l} y' = -kt^\beta y, \\ y(0) = 1, \end{array} \right.$$
   donde $k = 4$ y $\beta = 0.25$. Resolver el PVI en el intervalo $[0, 4]$, utilizando los métodos de Euler explícito y Euler implícito. Analizar la estabilidad de los dos métodos utilizando para ello distintos pasos de discretización, por ejemplo $h_1 = 0.5$, $h_2 = 0.4$, $h_3 = 0.2$ y $h_4 = 0.1$.

5. Se considera el problema de valor inicial:
$$\left\{ \begin{array}{l} y' = \frac{2y}{t+1}, \\ y(0) = 1. \end{array} \right.$$

En el intervalo temporal $[0, 1]$, utilizar los esquemas numéricos de Euler explícito, Euler implícito y Crank-Nicholson para obtener el valor aproximado de la solución en $t = 1$, tomando un paso de discretización $h = 0.1$. Sabiendo que la solución exacta es $y(t) = (t + 1)^2$, dibujar las soluciones obtenidas con los tres métodos y la solución exacta en un mismo plot. Comparar los resultados obtenidos. Escribir los valores de las distintas soluciones en $t = 0.5$ y $t = 0.75$. Encontrar los intervalos temporales para los cuales las distintas soluciones numéricas alcanzan valores superiores a tres.

6. Se considera el problema de valor inicial:

$$\begin{cases} y^{'} = -y(y-1), \\ y(0) = 2. \end{cases}$$

Aplicar los métodos tipo Runge-Kutta dados por los algoritmos heun.m y rungekuttao3.m, para resolver el PVI anterior en el intervalo temporal [0,1], tomando como paso de discretización $h = 0.01$ Escribir los valores obtenidos con los dos métodos para los tiempos $t = 0.1$, $t = 0.5$ y $t = 1$.

7. Sea dado el Problema de Transporte Difusivo y Convectivo definido por el PVC:

$$\begin{cases} -u^{''} + u^{'} = -2e^{-x}, \quad \forall x \in (0, 1), \\ u(0) = 0, \quad u(1) = 2cosh(1), \end{cases}$$

cuya solución analítica es:

$$sol(x) = e^x + e^{-x}$$

.

   (a) Aplicar el algoritmo bvpdirichlet.m para calcular la solución en el intervalo [0, 1] con paso de discretización $h = 0.01$.
   (b) Dibujar la solución analítica junto con la solución numérica.
   (c) Determinar los valores máximos y mínimos de u y su localización, para la solución numérica y la solución analítica. Calcular los errores cometidos.

8. Sea dado el PVIC:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x, t), \\ u(x, t) = g(x, t), \quad x = 0, \quad x = 2, \\ u(x, 0) = u_0(x), \quad x \in (0, 2), \end{cases}$$

donde

$$f(x, t) = tsin(x),$$

$$g(x, t) = 0,$$

$$u_0(x) = x, \text{ para } 0 \leq x \leq 1 \quad \text{y} \quad u_0(x) = 2 - x, \text{ para } 1 < x \leq 2.$$

Aplicar el algoritmo de ecucalor.m para calcular la solución en el intervalo temporal [0,1], con paso de discretización espacial $\Delta x = 0.05$ y discretización temporal, $\Delta t = 0.02$. Dibujar la solución obtenida para $\theta = 0.5$ en $t = 1$. ¿Cuánto vale la solución en el punto $x = 1$?