Primera práctica voluntaria — Métodos Matemáticos y Técn. Comp.
Profesor Francisco R. Villatoro — 27 de Octubre de 1999

Repaso de álgebra y espacios vectoriales de polinomios.

Se recomienda estudiar el tutorial de Matlab entregado a los alumnos de la asignatura de Técnicas Numéricas de 3er. curso.

1. Escribe un algoritmo en Matlab para rellenar una matriz `A` de $n \times n$ tridiagonal con valores $\langle -1, 3, -1 \rangle$, donde el 3 está en la diagonal. Mira la ayuda de la función de Matlab `diag`. Ahora escribe un algoritmo para rellenar dicha matriz `As` considerada como *sparse* utilizando la función `spdiags`. Escribe un algoritmo para rellenar un vector `b` $= (2, 1, \ldots, 1, 2)^T$ de $n \times 1$. Calcula la solución del problema $Ax = b$ mediante:

   a) Matlab directamente para matrices densas `x = A\b;`

   b) Matlab directamente para matrices *sparse* `x = As\b;`

   c) Calculando la inversa para matrices densas `x = inv(A)*b;`

   d) Calculando la inversa para matrices *sparse* `x = inv(As)*b;`

   e) Implementando en Matlab el algoritmo de Thomas (Gauss para tridiagonales, visto en Técnicas Numéricas). Hazlo.

   Compara el coste computacional de los tres procedimientos. Para ello dibuja una gráfica del número de operaciones flotantes en función de n (la dimensión de la matriz). Comenta los resultados que obtienes. Realiza un ajuste mínimo-cuadrático con polinomios de grado 1, 2 y 3 de los datos de dichas figuras. ¿Qué conclusiones obtienes?

2. Escribe la siguiente función en un fichero `matriz.m`,

```
function H = matriz(n)
%   C. Moler, 6-22-91.
%   Copyright (c) 1984-97 by The MathWorks, Inc.
%   $Revision: 5.5 $  $Date: 1997/04/08 05:49:32 $
J = 1:n;
J = J(ones(n,1),:);
I = J';
E = ones(n,n);
H = E./(I+J-1);
```

¿Cómo son las matrices que genera dicha función? Representa gráficamente el número de condición de dichas matrices en función de n usando `cond` y `condest`. Compara los resultados de estas dos funciones. Comenta los resultados. Calcula la inversa de las matrices `matriz(n)` utilizando `inv`. Compara `inv(matriz(10))*matriz(10)` con el resultado de `inv(matriz(11))*matriz(11)`. Comenta y justifica los resultados que obtienes y que deberías obtener. Estudia el comportamiento de `invhilb(n)*matriz(n)` para varios n.

3. Para interpolar una serie de puntos (valores de una función) se puede utilizar tanto polinomios globales como locales (a trozos). El polinomio de Lagrange para interpolación global se puede calcular mediante el algoritmo de Aitken. Escribe la funciones de Matlab `aitken.m` y `aitkenvector.m`.

```
function Q=aitken(x,y,xval)
%% Q=aitken(x,y,xval)
%% Calcula el valor segun interpolacion polinomica (Lagrange)
%%  x, y son vectores con los datos a interpolar
%%  xval es el punto donde se desea el valor interpolado
%%  Q es el resultado del interpolante
%% Nota: si x,y tienen n puntos, es un interpolante de grado n-1
n=length(x); P=zeros(n);
P(1,:)=y;
for j=1:n-1,
  for i=j+1:n,
    P(j+1,i) = (P(j,i)*(xval-x(j))-P(j,j)*(xval-x(i)))/(x(i)-x(j));
  end
end
Q=P(n,n);
```

```
function Q=aitkenvector(xx,yy,x)
%% Q=aitken(xx,yy,x)
%% Calcula el valor segun interpolacion polinomica (Lagrange)
%%  xx, yy son vectores con los datos a interpolar
%%  x vector donde se calcularan los puntos a interpolar
%%  Q es el vector resultado del interpolante Q(x)
```

```
%% Nota: si x,y tienen n puntos, es un interpolante de grado n-1
n=length(x); Q=zeros(size(x));
for j=1:n,
  Q(j) = aitken(xx,yy,x(j));
end
```

Vamos a estudiar las diferencias entre la interpolación polinómica global y la local. Escojamos una función gaussiana,

```
%%  Gaussian
x = -5:0.1:5; y = exp(-x.^2); plot ( x, y)
%%  Some points
xx = -5:1:5; yy = exp(-xx.^2); plot ( x,y, xx,yy,'*')
```

Ahora lo interpolaremos con polinomios globales y locales a trozos lineales y cúbicos:

```
%%  Global interpolation
yipol = aitkenvector ( xx,yy,x); plot ( x,y,x,yipol)
%%  Linear Local interpolation
yilin = interp1 ( xx,yy,x,'linear'); plot ( x,y,x,yilin)
%%  Cubic Local interpolation
yicub = interp1 ( xx,yy,x,'cubic'); plot ( x,y,x,yicub)
%%  Spline (cubic) Local interpolation
yispline = interp1 ( xx,yy,x,'spline'); plot ( x,y,x,yispline)
%%  Error behaviour
plot ( x, y-yipol', x,y-yilin',x,y-yicub',x,y-yispline')
```

Comenta los resultados que has obtenido. Cuál es la más precisa de todas las interpolaciones. Realiza un estudio de cómo varía el error de interpolación global conforme el número de puntos aumenta. Haz lo mismo con la interpolación local lineal. Comenta tus conclusiones y presenta las gráficas que consideres oportunas para confirmar tus conclusiones.

4. Matlab permite la resolución de ecuaciones diferenciales tanto numérica (directamente) como simbólica (usando Maple). Pongamos un ejemplo. Sea la ecuación diferencial

$$x''(t) + 2x(t) = 0, \qquad x(0) = 0, \quad x'(0) = 1,$$

cuya solución exacta es fácil de calcular. Detalla su cálculo a mano. Matlab también nos la permite calcular

```
dsolve('D2x + 2*x = 0', 'x(0)=0, Dx(0)=1').
```

También se pueden resolver problemas de contorno, por ejemplo,

$$x''(t) + 2x(t) = 0, \qquad x(0) = 0, \quad x(3) = 1.$$

Detalla el cálculo de la solución a mano. Matlab nos confirma el resultado obtenido

```
dsolve('D2x + 2*x = 0', 'x(0)=0, x(3)=1').
```

¿Cómo dibujarías en Matlab la gráfica de las soluciones obtenidas?

Consideremos otro ejemplo. Sea la ecuación del péndulo no lineal forzado

$$x'' + 0{,}1x' + \sin x = 0{,}02\cos t, \qquad x(0) = 0, \quad x'(0) = 1.$$

¿`dsolve` es capaz de resolver esta ecuación? Podemos calcular la solución numéricamente mediante el método de Runge-Kutta de orden 4(5) de paso adaptativo `ode45`. Primero escribimos la ecuación como un sistema de primer orden y el término no homogéneo lo metemos en una función `pendulo.m`,

```
function yprima = pendulo(t,y);
% función vectorial de dos componentes en t
yprima = [ y(2) ; -0.1*y(2)-sin(y(1))-0.02*cos(t)];
```

Ahora aplicamos la función de Matlab y dibujamos el espacio de fases

```
[t, y] = ode45 ('pendulo', 0, 3, [0:1]);
plot(y(:,1),y(:,2))
```

Dibuja el espacio de fases para varias condiciones iniciales, y comenta el comportamiento de la solución que observas.