Tercera práctica voluntaria Métodos Matemáticos y Técn. Comp.
Francisco R. Villatoro y Carmen M. García 16 de Enero de 2001

SOLUCIÓN

Vamos a resolver el problema parabólico de reacción-difusión siguiente

$$\frac{\partial u(x,t)}{\partial t} - \frac{\partial}{\partial x}\left(a(x)\,\frac{\partial u(x,t)}{\partial x}\right) + b(x)\,u(x,t) = f(x), \qquad t>0, \quad x\in(0,1),$$

$$u(x,0) = u_0(x), \quad x\in(0,1), \qquad u(0,t) = \frac{\partial u}{\partial x}(1,t) = 1, \quad t>0,$$

donde $a(x) = 1+x$, $b(x) = x^2$, $f(x) = x^3$ y $u_0(x) = x^2 - x + 1$.

Primero, vamos a obtener la solución en el espacio de polinomios a trozos lineales continuos en una malla del intervalo [0,1] arbitraria.

Las funciones base del espacio $\mathcal{P}^{(1)}(I_i)$, en el intervalo $I_i = [x_{i-1}, x_i)$ se calculan fácilmente con

```
function res = philineal (z,i,j)
% function res = philineal (z,i,j)
%  Dado el intervalo I_j = [x_{j-1}, x_j)
%  se definen las funciones
%    i=0, phi(x_{j-1})=1, phi(x_{j-1/2})=0, phi(x_j)=0
%    i=1, phi(x_{j-1})=0, phi(x_{j-1/2})=0, phi(x_j)=1
%  luego
%    i \in {0,1}
%    z \in [0,1],
%    j \in {1, ..., length(x)}
%  calcula la base en psi_j(x) definida en intervalos
%  I_{j-1} y I_j

global x; % el vector que define la malla
xj = x(j); xjm1 = x(j-1);

if (i==0),     res = (z-xj)/(xjm1-xj);
elseif (i==1), res = (z-xjm1)/(xj-xjm1);
else           error('i no valido en phi(z,i,j)');
end
```

Cuyas derivadas se calculan fácilmente.

```
function res = philinealprima (z,i,j)
% ...
global x; % el vector que define la malla
xj = x(j); xjm1 = x(j-1);

if (i==0),     res = z*0+(1/(xjm1-xj));
elseif (i==1), res = z*0+(1/(xj-xjm1));
else           error('i no valido en philinealprima(z,i,j)');
end
```

Las funciones base del espacio $V_h^{(1)}$ se obtienen "pegando"las funciones anteriores de la siguiente forma.

```
function res = psilineal (z,j)
% function res = psilineal (z,j)
%    z \in [0,1],
%    j \in 1, 2, 3,... length(x)
%  Para j entero calcula la base psi_j(x)
%    definida en los intervalos I_{j-1} y I_j, con psi(x_j)=1

global x; % el vector que define la malla
N=length(x);

if     (j==1), res = philineal(z,0,j+1).*(z>=x(j)).*(z<x(j+1));
elseif (j==N), res = philineal(z,1,j).*(z>x(j-1)).*(z<=x(j));
else           res = philineal(z,1,j).*(z>x(j-1)).*(z<=x(j)) + ...
                     philineal(z,0,j+1).*(z>x(j)).*(z<x(j+1));
end
```

Cuyas derivadas se calculan fácilmente sustituyendo `philineal` por `philinealprimea`.

```
function res = psilinealprima (z,j)
% ...
```

Podemos dibujar fácilmente estas funciones para comprobar su corrección.

```
global x;
```

```
x=0:0.1:1; N=length(x); y=0:0.01:1;

% Dibuja algunas funciones base
subplot (2,2,1); plot ( y, psilineal(y,1),x,psilineal(x,1),'*');
 title('psi_1(x)');xlabel('x');
subplot (2,2,2); plot ( y, psilineal(y,2),x,psilineal(x,2),'*');
 title('psi_2(x)');xlabel('x');
subplot (2,2,3); plot ( y, psilineal(y,3),x,psilineal(x,3),'*');
 title('psi_3(x)');xlabel('x');
subplot (2,2,4); plot ( y, psilineal(y,N),x,psilineal(x,N),'*');
 title('psi_N(x)');xlabel('x');
```

Para que nuestra solución sea más general, definiremos funciones Matlab para cada una de las funciones de los datos.

```
function res = a (x)
% function res = a (x)
```

```
%    a = 1+x

res = 1 + x;

function res = b (x)
% function res = b (x)
%    b = x^2

res = x.^2;

function res = f (x)
% function res = f (x)
%    f = x^3

res = x.^3;

function res = u0 (x)
% function res = u0 (x)
%    u_0 = x^2-x+1

res = x.^2-x+1;
```

El método de elementos finitos conduce a la ecuación diferencial ordinaria

$$M\,\frac{dc(t)}{dt} + A\,c(t) = b,$$

donde

$$(M)_{ij} = \int_0^1 psi_i(z)\,psi_j(z)\,dz, \qquad i,j = 1,2,\ldots N,$$

$$(A)_{kl} = \int_0^1 a(z)\,psi_i'(z)\,psi_j'(z)\,dz + \int_0^1 b(z)\,psi_i(z)\,psi_j(z)\,dz, \qquad i,j = 1,2,\ldots N,$$

$$(b)_i = \int_0^1 f(z)\,psi_i(z) - 2\,\delta_{iN} + \delta_{i1}\,\left(\int_0^1 a(z)\,psi_i'(z)\,psi_0'(z)\,dz + \int_0^1 b(z)\,psi_i(z)\,psi_0(z)\,dz\right), \qquad i = 1$$

Para calcular estas integrales usaremos `quad` que requiere que definamos funciones para los integrandos. Sean las siguientes.

```
function res = Mij (z)
% function res = Mij (z)
```

```
%   es la función integrando para calcular
% M es una matriz (N-1)x(N-1)
%  (M)_{ij} = \int_0^1 psi_{i+1}\,psi_{j+1}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global i j;

res = psilineal(z,i).*psilineal(z,j);

function res = Aij (z)
% function res = Aij (z)
%   es la función integrando para calcular
% A, que es una matriz (N-1)x(N-1)
%  (A)_{kl} = \int_0^1 a(x)\,psi'_{i+1}\,psi'_{j+1}
%             +\int_0^1 b(x)\,psi_{i+1}\,psi_{j+1}

global i j;

res = a(z).*psilinealprima(z,i).*psilinealprima(z,j)+...
      b(z).*psilineal(z,i).*psilineal(z,j);

function res = bi (z)
% function res = bi (z)
%   es la función integrando para calcular
% b es un vector (N-1)
%  (b)_{i} = \int_0^1 psi_{i+1}\,f(x)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global i;

res = psilineal(z,i).*f(z);
```

Ahora podemos calcular directamente las matrices M y A, y el vector b.

```
global i j;
M = zeros (N-1,N-1);
A = zeros (N-1,N-1);
```

```
for i=2:N; for j=2:N;
  M(i-1,j-1)= quad8('Mij',0,1);
  A(i-1,j-1)= quad8('Aij',0,1);
end; end
%% bb en lugar de b, para confundir con "b.m"
bb = zeros (N-1,1);
for i=2:N;
  bb(i-1)= quad8('bi',0,1);
end;
%% Falta añadir los términos debidos a las Cond. Contorno
i=1; j=2; bb(1)=bb(1) - quad8('Aij',0,1);
bb(N-1)=bb(N-1) + 2;
```

NOTA: la integración se realiza en todo el intervalo $[0, 1]$. Es fácil cambiar el código para que la integración sólo se realice en el intervalo de solapamiento.

Para integrar estas ecuaciones podemos usar un método θ (p.ej. Crank-Nicolson con $\theta = 1/2$), de la forma siguiente.

```
%  M dc(t)/dt + A c(t) = bb
%
% theta-metodo
th = 1/2;
dt = 0.1;
T = 10; Nk = round(T/dt);

cn = u0(x)'; plot (x,cn);hold on;
for k=1:Nk,
 cn(2:N) = (M+dt*th*A)\( (M-dt*(1-th)*A)*cn(2:N) + dt*bb );
 if (rem(k,10)==0), plot (x,cn); end
end
```

El resultado aparece en la siguiente figura.