

## spline2

```
function [S,S1,S2]=spline2(x,y,z)
% Questa funzione calcola la Spline Cubica Naturale relativa ai nodi di
% interpolazione x equidistanti con ordinata y e la valuta in tutti i punti
% contenuti nel vettore z. Per risolvere il sistema lineare con matrice
% tridiagonale necessario per determinare i coefficienti della spline
% utilizza l'algoritmo di Thomas modificato.

n=length(x)-1; % i nodi equidistanti sono n+1
h=x(2)-x(1);
mu=1/2*ones(n-1,1); lambda=mu;
d=3*(y(3:n+1)-2*y(2:n)+y(1:n-1))/(h^2);

lambda0=0;d0=0;mun=0;dn=0; %spline naturale

M=modthomas([2*ones(n+1,1)],[mu;mun],[lambda0;lambda],[d0;d';dn])';
C=(y(2:n+1)-y(1:n))/h-h*(M(2:n+1)-M(1:n))/6;
Ctilde=y(1:n)-M(1:n)*h^2/6;

J=2;
for I=1:length(z)
    while z(I)>x(J) & J<n+1 %ovvero se z non e' in [x(j-1),x(j)]
        J=J+1;
    end
    S(I)=M(J-1)*(x(J)-z(I))^3/(6*h)+M(J)*(z(I)-x(J-1))^3/(6*h)+...
        C(J-1)*(z(I)-x(J-1))+Ctilde(J-1);
    S1(I)=-M(J-1)*(x(J)-z(I))^2/(2*h)+M(J)*(z(I)-x(J-1))^2/(2*h)+C(J-1);
    S2(I)=M(J-1)*(x(J)-z(I))/h+M(J)*(z(I)-x(J-1))/h;
end
```