

```
function Y=cubicspline(x,f,X)

n = length(x);

b = zeros(4*(n-1),1);
A = zeros(4*(n-1),4*(n-1));

% solves for a not a knot case;
% I decided to go for not a knot, knot just because it followed the example
% code you provided us with, but because i think it is more prudent to focus
% on the the center portions of the interval
for i = 1:n-1
    % next two lines are setting up the basic polynomial system via monomial
    A(2*i-1, 4*(i-1)+1:4*(i)) = [1,x(i),x(i).^2,x(i).^3];
    A(2*i, 4*(i-1)+1:4*(i)) = [1,x(i+1),x(i+1).^2,x(i+1).^3];
    % these two lines
    b(2*i-1) = f(i);
    b(2*i) = f(i+1);
    if i==n-1
        A(4*(n-1)-1,1:8) = [0 0 0 6 0 0 0 -6];
        A(4*(n-1),4*(n-3)+1:4*(n-1)) = [0 0 0 6 0 0 0 -6];
    else
        A(2*(n-1)+i,4*(i-1)+1:4*(i+1)) = [0,1,2*x(i+1),3*x(i+1).^2,0,-1,-2*x(i+1),-3 *x(i+1).^2];
        A(3*(n-1)+i-1,4*(i-1)+1:4*(i+1)) = [0,0,2,6*x(i+1),0,0,-2,-6*x(i+1)];
    end
end

C = A\b;

m = length(X);
Y = zeros(1,m);

for i = 1:m
    for j = 1:n-1
        if X(i) <= x(j+1) && X(i) >= x(j)
            Y(i) = C(1+4*(j-1)) + C(2+4*(j-1))*(X(i)) + C(3+4*(j-1))*(X(i)).^2 + C(4+4*(j-1))*(X(i)).^3;
        else
            end
        end
    end

% plot(X,Y,x,f,'o')
```