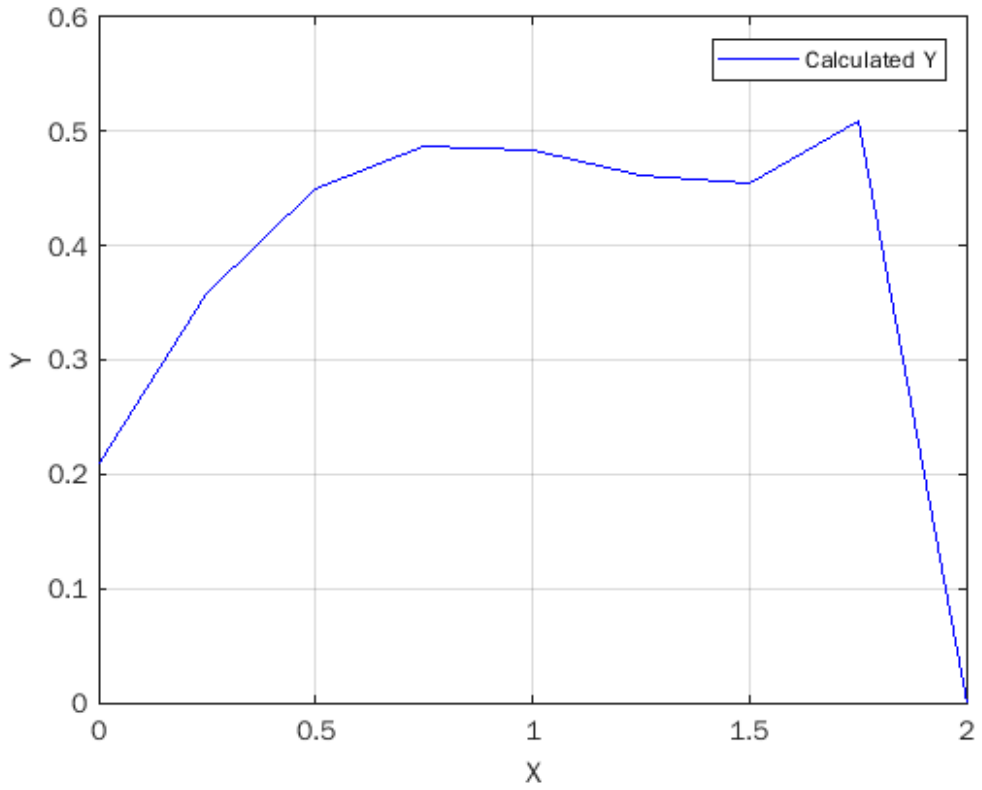
```
h = 0.25;
x = 0:h:2;
n = length(x);
% Finding the entries of the block tridiagonal matrix
a = (1/(h*h))*eye(2);
a(2,2) = a(2,2) + (1/h);
A = a;
for i=2:n
    A=cat(3,A,a);
end
b = (-2/(h*h))*eye(2);
b(1,2) = -1;
b(2,1) = 1;
B = b;
for i=2:n
    B=cat(3,B,b);
end
c = (1/(h*h))*eye(2);
c(2,2) = c(2,2) - (1/h);
C = c;
for i=2:n
    C=cat(3,C,c);
end
d = zeros(2,1);
d(2,1) = -(x(1)^4)+1;
D = d;
for i=2:n
    d(2,1)= -(x(i)^4)+1;
    D=cat(3,D,d);
end
C(2,2,1) = C(2,2,1)+A(2,2,1);
A(2,2,1)=0;
A(1,1,1) = 0;
A(2,2,n)=A(2,2,n)+C(2,2,n);
D(2,1,n) = D(2,1,n)-(2*h)*C(2,2,n);
D(1,1,n) = D(1,1,n)-C(1,1,n);
C(2,2,n)=0;
C(1,1,n) = 0;
y = zeros(2,1,n);
% Thomas algorithm
gamma = zeros(2,2);
beta = zeros(2,2);
gamma = B(:,:,1)\C(:,:,1);
beta = B(:,:,1)\D(:,:,1);
for i=2:n
    gamm = (B(:,:,i)-A(:,:,i)*gamma(:,:,i-1))\C(:,:,i);
    gamma = cat(3,gamma,gamm);
    bet = (B(:,:,i)-A(:,:,i)*gamma(:,:,i-1))\(D(:,:,i)-
A(:,:,i)*beta(:,:,i-1));
    beta = cat(3,beta,bet);
end
y(:,:,n) = beta(:,:,n);
```

```
for i=n-1:-1:1
    y(:,:,i) = beta(:,:,i) - gamma(:,:,i)*y(:,:,i+1);
end
fprintf("%6s %20s\n",'X','Calculated value');
Y = zeros(n,1);
for i=1:n-1
    fprintf('%6.2f %20.8f\n',x(i),y(1,1,i));
    Y(i) = y(1,1,i);
end
% Plotting graph to compare actual and expected values
plot(x,Y,'b');
grid on;
xlabel('X');
ylabel('Y');
legend('Calculated Y');
```

```
     X     Calculated value
  0.00           0.20933410
  0.25           0.35840066
  0.50           0.44874394
  0.75           0.48625904
  1.00           0.48331674
  1.25           0.46156277
  1.50           0.45455609
  1.75           0.50831699
```



Published with MATLAB® R2021b