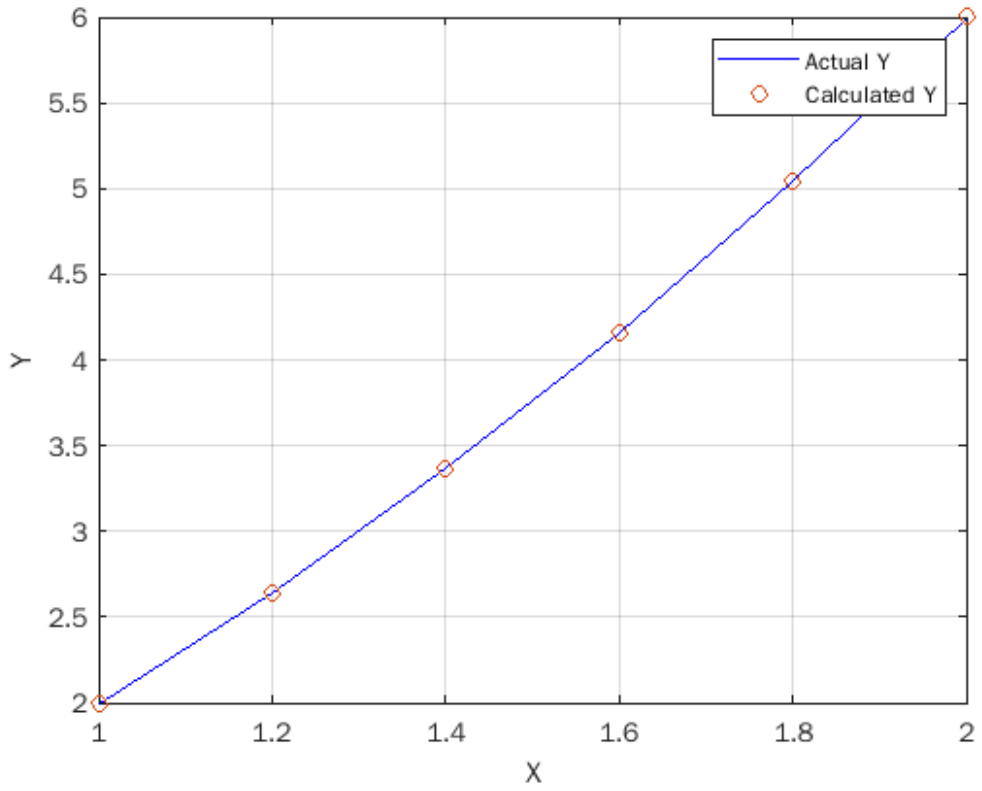
```
h = 0.2;
x = 1:h:2;
n = length(x);
y = zeros(n,1);
y(1) = 2;
y(n) = 6;
diff = (6-2)/(n-1);
for i=2:n-1
    y(i)=y(i-1)+diff;
end
y1 = zeros(n,1);
y2 = zeros(n,1);
for i=2:n-1
    y1(i) = (y(i+1)-y(i-1))/(2*h);
    y2(i) = (y(i+1)-2*y(i)+y(i-1))/(h*h);
end
eps = 0.00001;
err = 1;
while err > eps
    a = zeros(n,1);
    b = zeros(n,1);
    c = zeros(n,1);
    d = zeros(n,1);

    for i = 2:n-1
        a(i) = (y(i)/(h*h) + 1/(2*h));
        b(i) = -2*y(i)/(h*h) + y2(i);
        c(i) = (y(i)/(h*h) -1/(2*h));
        d(i) = y(i)*y2(i) + 2*x(i)^2 - 1;
    end

    d(2) = d(2)-a(2)*2;
    a(2) = 0;
    d(n-1) = d(n-1)-c(n-1)*6;
    c(n-1) = 0;
    del = zeros(n,1);

    % Thomas algorithm

    gamma = zeros(n,1);
    beta = zeros(n,1);
    gamma(2) = c(2)/b(2);
    beta(2) = d(2)/b(2);
    for i=3:n-1
        gamma(i) = (c(i))/(b(i)-a(i)*gamma(i-1));
        beta(i) = (d(i)- a(i)*beta(i-1))/(b(i)-a(i)*gamma(i-1));
    end
    del(n-1)=beta(n-1);
    err = abs(del(n-1)-y(n-1));
    y(n-1)=del(n-1);

    for i=n-2:-1:2
```

```
        del(i) = beta(i) - gamma(i)*del(i+1);
        if(abs(del(i)-y(i))>err)
            err = abs(del(i)-y(i));
        end
        y(i)=del(i);
    end
    for i=2:n-1
        y1(i) = (y(i+1)-y(i-1))/(2*h);
        y2(i) = (y(i+1)-2*y(i)+y(i-1))/(h*h);
    end
end

xmesh=linspace(1,2,n);
solinit=bvpinit(xmesh,@guess);
sol=bvp4c(@bvpfcn,@bcfcn,solinit);
plot(sol.x,sol.y(1,:),'b',x,y,'o');
grid on;
xlabel('X');
ylabel('Y');
legend('Actual Y', 'Calculated Y');

y
sol.y(1,:)

function dydx=bvpfcn(x,y)
dydx=zeros(2,1);
dydx=[y(2)
     (2*x^2-1+y(2))/y(1)];
end

function res=bcfcn(ya,yb)
res=[ya(1)-2
     yb(1)-6];
end

function g=guess(x)
g=[x^2
    0];
end
```

```
y =

    2.0000
    2.6400
    3.3600
    4.1600
    5.0400
    6.0000


ans =

    2.0000    2.6400    3.3600    4.1600    5.0400    6.0000
```



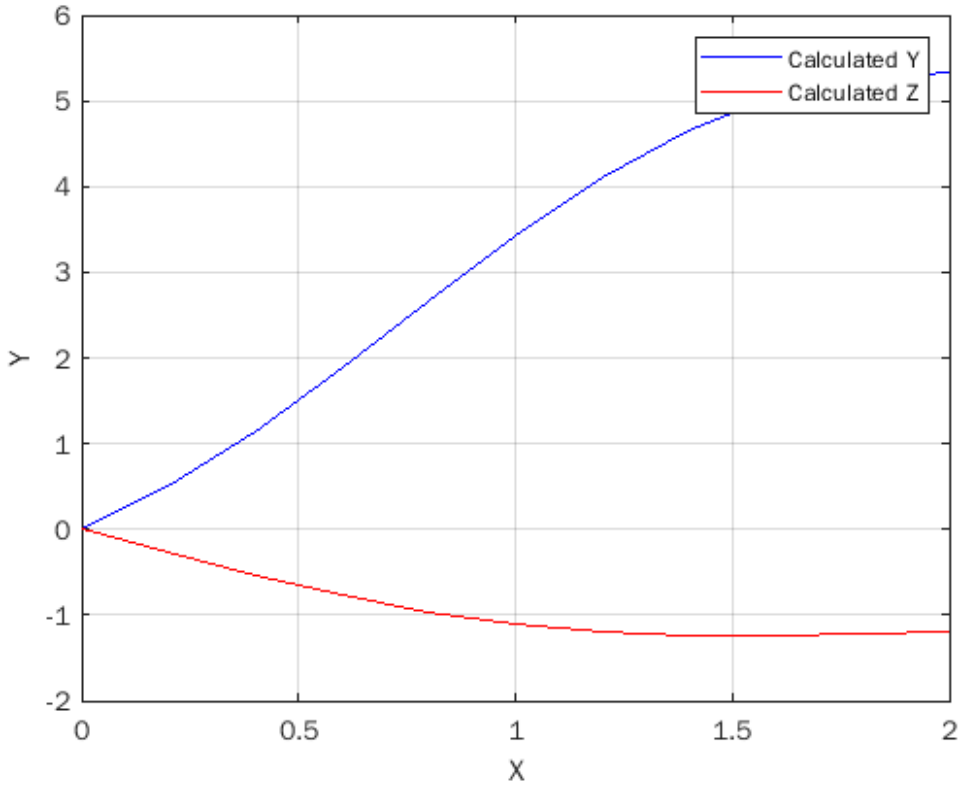
Published with MATLAB® R2021b

```
h = 0.2;
x = 0:h:2;
n = length(x);
% Finding the entries of the block tridiagonal matrix
a = eye(2);
a(1,1) = 1/(h*h)-(x(1)-2)/(2*h);
a(2,2) = 1/(h*h)+1/h;
A = a;
for i=2:n
    a(1,1) = 1/(h*h)-(x(i)-2)/(2*h);
    a(2,2) = 1/(h*h)+1/h;
    A=cat(3,A,a);
end
b = (-2/(h*h))*eye(2);
b(1,2) = -6;
b(2,1) = x(1);
B = b;
for i=2:n
    b = (-2/(h*h))*eye(2);
    b(1,2) = -6;
    b(2,1) = x(i);
    B=cat(3,B,b);
end
c = eye(2);
c(1,1) = 1/(h*h)+(x(1)-2)/(2*h);
c(2,2) = 1/(h*h)-1/h;
C = c;
for i=2:n
    c(1,1) = 1/(h*h)+(x(i)-2)/(2*h);
    c(2,2) = 1/(h*h)-1/h;
    C=cat(3,C,c);
end
d = zeros(2,1);
d(1,1) = x(1)*x(1);
d(2,1) = 4*x(1)+2;
D = d;
for i=2:n
    d(1,1) = x(i)*x(i);
    d(2,1) = 4*x(i)+2;
    D=cat(3,D,d);
end
A(1,1,2) = 0;
A(1,2,2) = 0;
A(2,2,2) = 0;
A(2,1,2) = 0;
A(:,:,n) = A(:,:,n)+C(:,:,n);
C(1,1,n) = 0;
C(1,2,n) = 0;
C(2,2,n) = 0;
C(2,1,n) = 0;
y = zeros(2,1,n);
% Thomas algorithm
```

```
gamma = zeros(2,2);
beta = zeros(2,2);
gamma = B(:,:,2)\C(:,:,2);
beta = B(:,:,2)\D(:,:,2);
for i=3:n
    gamm = (B(:,:,i)-A(:,:,i)*gamma(:,:,i-2))\C(:,:,i);
    gamma = cat(3,gamma,gamm);
    bet = (B(:,:,i)-A(:,:,i)*gamma(:,:,i-2))\(D(:,:,i)-
A(:,:,i)*beta(:,:,i-2));
    beta = cat(3,beta,bet);
end
y(:,:,n) = beta(:,:,n-1);
for i=n-1:-1:2
    y(:,:,i) = beta(:,:,i-1) - gamma(:,:,i-1)*y(:,:,i+1);
end
fprintf("%6s %20s\n",'X','Calculated value Y');
Y = zeros(n,1);
for i=1:n
    fprintf('%6.2f %20.8f\n',x(i),y(1,1,i));
    Y(i) = y(1,1,i);
end
fprintf("%6s %20s\n",'X','Calculated value Z');
Z = zeros(n,1);
for i=1:n
    fprintf('%6.2f %20.8f\n',x(i),y(2,1,i));
    Z(i) = y(2,1,i);
end
plot(x,Y,'b',x,Z,'r');
grid on;
xlabel('X');
ylabel('Y');
legend('Calculated Y','Calculated Z');
```

```
     X   Calculated value Y
  0.00           0.00000000
  0.20           0.49890557
  0.40           1.14071722
  0.60           1.88267744
  0.80           2.66739419
  1.00           3.43076270
  1.20           4.11161774
  1.40           4.66106607
  1.60           5.04940035
  1.80           5.26904166
  2.00           5.33294698
     X   Calculated value Z
  0.00           0.00000000
  0.20          -0.26676260
  0.40          -0.53189555
  0.60          -0.77240933
  0.80          -0.96966032
  1.00          -1.11223257
  1.20          -1.19762908
  1.40          -1.23242091
```

```
1.60          -1.23088328
1.80          -1.21252886
2.00          -1.19921098
```



Published with MATLAB® R2021b