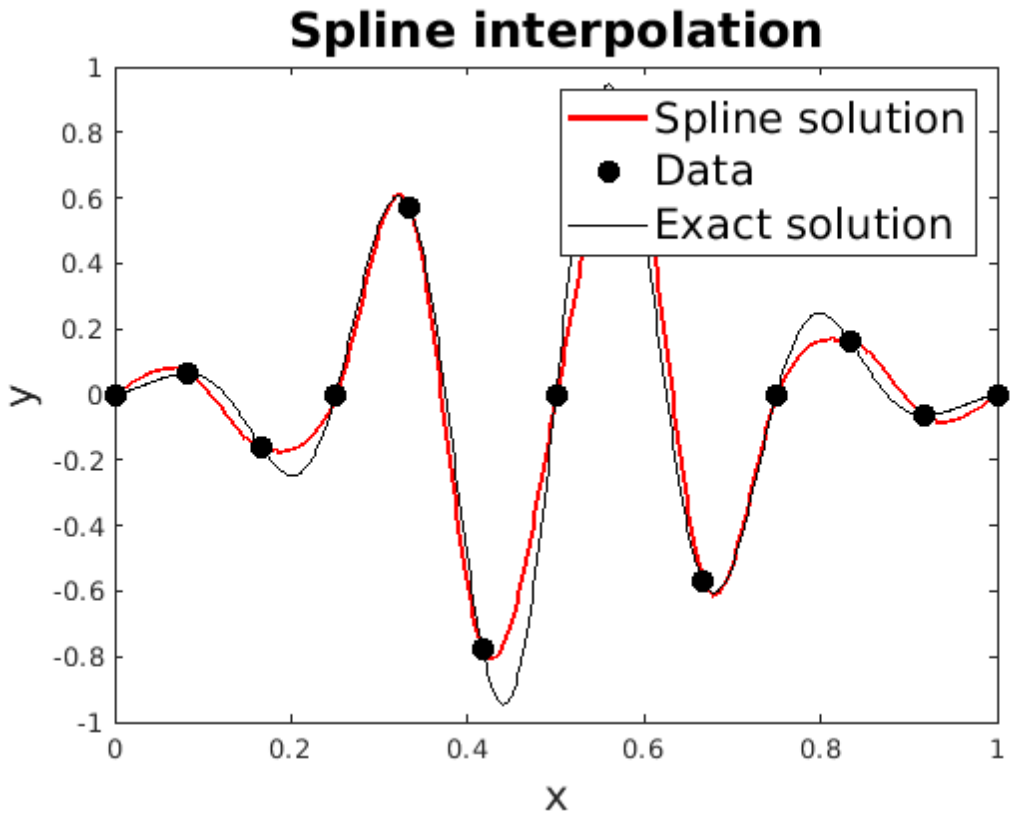
```
function spline_test()

% Function to interpolate
a = 15;
m = 8;
b = 1/2;
f = @(x) exp(-a*(x-b).^2).*sin(m*pi*x);
% Derivative (ne
fp = @(x) exp(-a*(x-b).^2).*(m*pi*cos(m*pi*x) - 2*a*(x-b)*sin(m*pi*x));

% Data at equispaced points
N = 12;
xdata = linspace(0,1,N+1);
ydata = f(xdata);

% Build the spline.  Currently, the derivatives at nodes are all set to
% zero.  Your job is to come up with a nicer spline by modifying
% the routines below.
math465 = false;
if (math465)
    pp = math465_build_spline(xdata,ydata);
else
    end_cond = 'natural';                          % 'natural' or 'clamped'
    end_data = [fp(xdata(1)); fp(xdata(end))]';   % For 'clamped' endpoint condition

    pp = math565_build_spline(xdata,ydata,end_cond,end_data);
end

% Evaluate the spline at points used for plotting
xv = linspace(0,1,500);
yv = ppval(pp,xv);       % Matlab function

% Plot results
figure(2)
clf;

% Plotting
plot(xv,yv,'r','linewidth',2);
hold on;
plot(xdata,ydata,'k.','markersize',30);
plot(xv,f(xv),'k');
legend('Spline solution','Data','Exact solution','fontsize',16);

xlabel('x','fontsize',16);
ylabel('y','fontsize',16);
title('Spline interpolation','fontsize',18);

shg

end
```



Published with MATLAB® R2020a