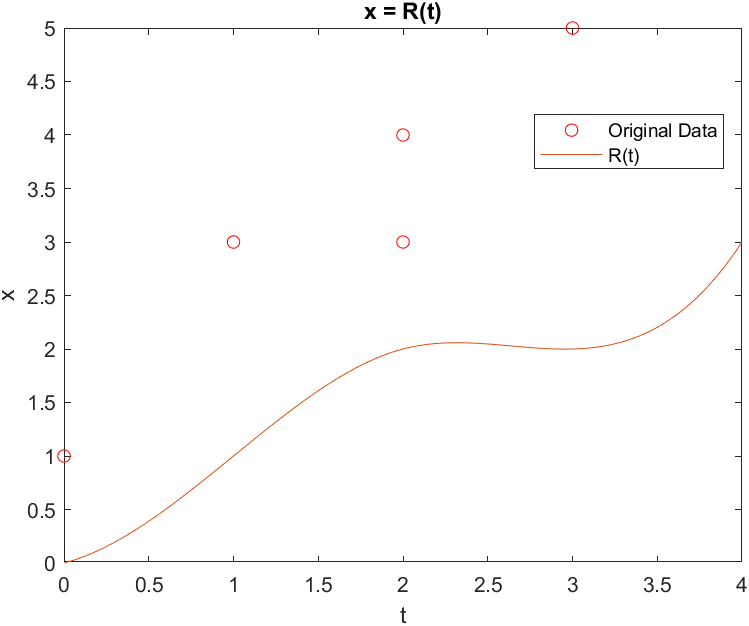
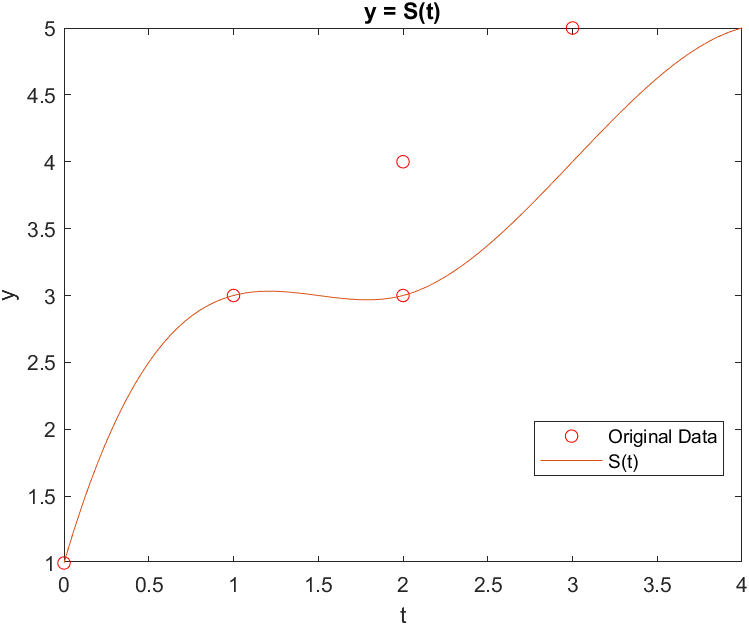
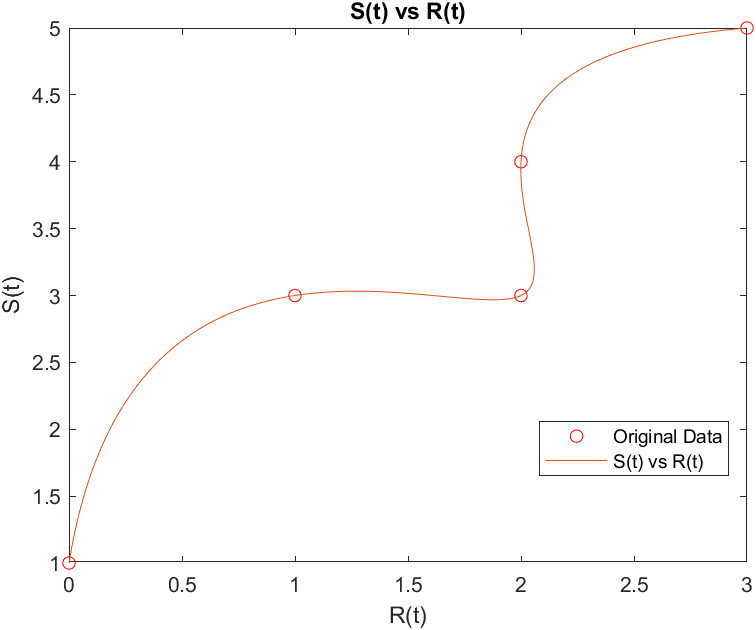
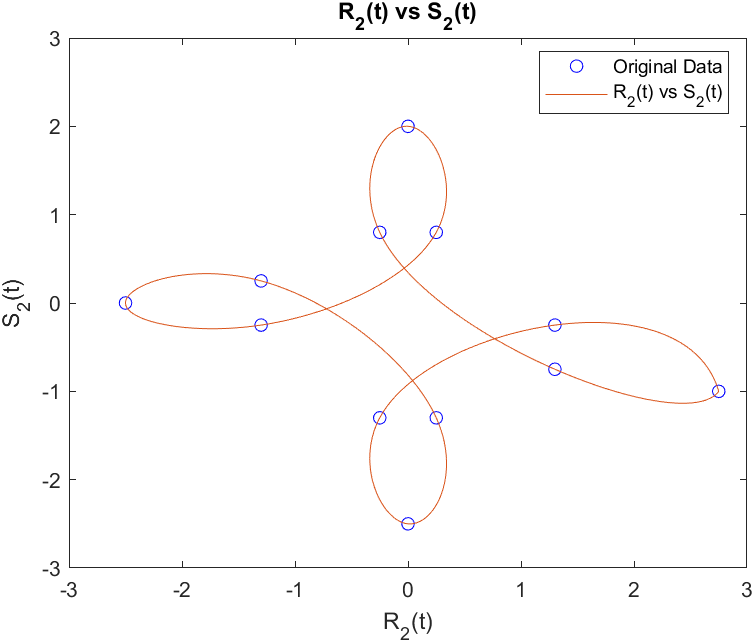
**MACM 316 – Assignment 7**

a)

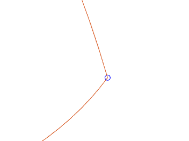
A plot of y = S(t), which passes thru all of the desired points (t,y).

A plot of x = R(t), which passes thru all of the desired points (t,x).





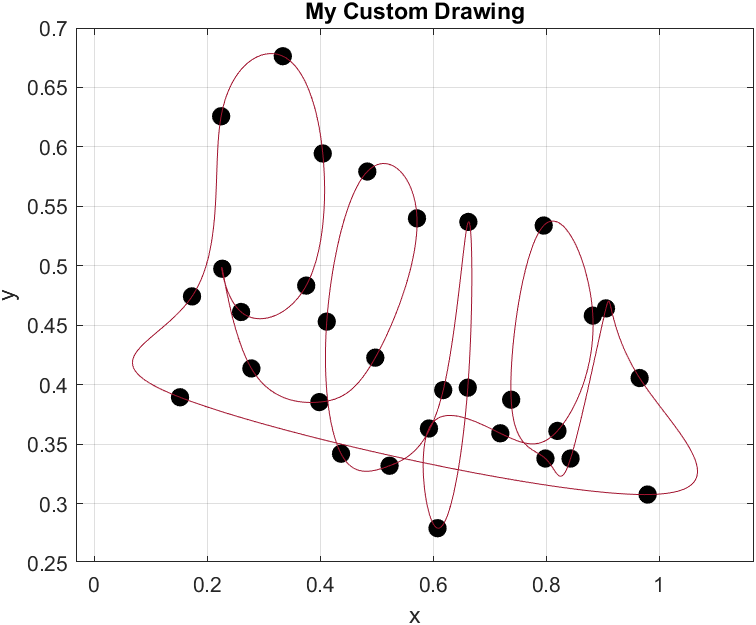
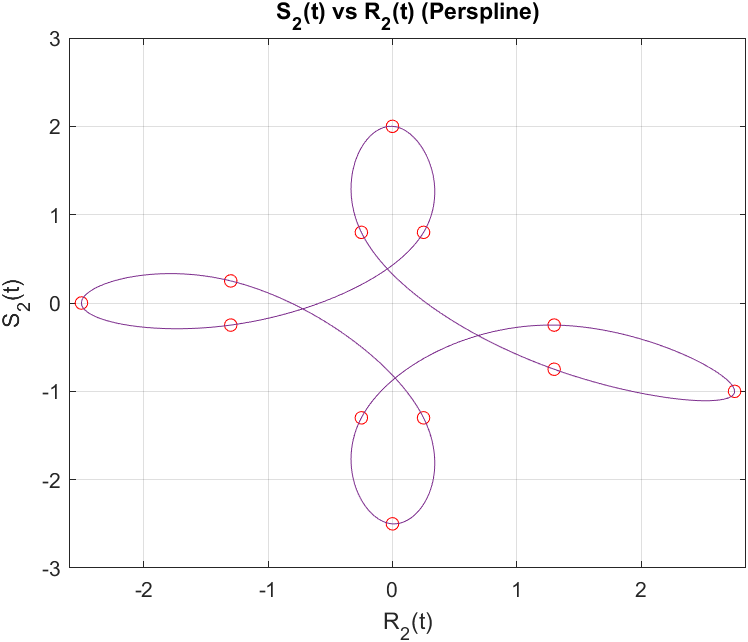
We can see that there is indeed a cusp on the right leaf.



Zoomed

b)

A plot of S(t) vs R(t), which accurately interpolates our original function.

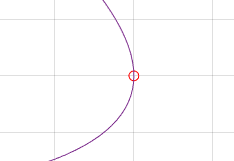


c) d)

Zoomed

We can see that the cusp at the given point has been eliminated.

Attempted to handwrite my name, resulting in the following periodic spline.



% a)

t = [0 1 2 3 4];

x = [0.0 1.0 2.0 2.0 3.0];

y = [1.0 3.0 3.0 4.0 5.0];

xx = linspace(0, 4, 1000);

grid on

pp1 = spline(t, y);

St = ppval(pp1, xx);

pp2 = spline(t, x);

Rt = ppval(pp2, xx);

plot(x, y, 'ro', xx, Rt)

title 'x = R(t)'

% PERSPLINE: Perform cubic spline interpolation on a given set

% of data points, using periodic end-point conditions.

% NOTE: Must have y(1)=y(end)!! So this is a modified version

% of the data used for the other spline examples.

function [ylist] = perspline(x,y, dots)

x = x';

y = y';

n = length(x) - 1;

ylist = [];

x2 = [];

y2 = [];

% Set up the matrix

h = diff(x);

diag0 = [1; 2\*(h(1:end-1)+h(2:end)); 2\*h(end)];

A = spdiags([[h;0], diag0, [0;h]], [-1, 0, 1], n+1, n+1);

% Then do a little surgery on the first/last rows ...

A(1,2) = 0;

A(1,end) = -1;

A(end,1) = 2\*h(1);

A(end,2) = h(1);

dy = diff(y);

% ... and the RHS vector

rhs = 6\*[0; diff(dy./h); dy(1)/h(1)-dy(end)/h(end)];

m = A \ rhs; % Solve for slopes, m\_i=S''(x\_i)

% Compute the cubic polynomial coefficients

a = y;

b = dy./h - h.\*m(1:end-1)/2 - h.\*diff(m)/6;

c = m(1:end-1)/2;

d = diff(m)./h/6;

% Plot each spline along with the data

for i = 1 : n

xx = linspace(x(i), x(i+1), 100);

yy = a(i) + b(i)\*(xx-x(i)) + c(i)\*(xx-x(i)).^2 ...

+ d(i)\*(xx-x(i)).^3;

ylist = [ylist, yy];

plot(xx, yy, 'r-')

hold on

end

if dots == 1

plot(x,y,'k.', 'MarkerSize', 30)

hold off

set(gca, 'XLim', [min(x)-0.1, max(x)+0.1])

xlabel('t'), ylabel('R(t)')

title 'x = R(t)'

grid on, shg

print -djpeg 'perspline.jpg'

elseif dots == 0

for i = 1:100:n

x2 = [x2, x(i)];

y2 = [y2, y(i)];

end

plot(x2,y2,'k.', 'MarkerSize', 30)

plot(x,y)

hold on

set(gca, 'XLim', [min(x)-0.1, max(x)+0.1])

xlabel('x'), ylabel('y')

title 'My Custom Drawing'

grid on, shg

print -djpeg 'perspline.jpg'

end

xlabel('t')

ylabel('x')

legend ('Original Data','R(t)', 'location', 'best')

plot(x, y, 'ro', xx, St)

title 'y = S(t)'

xlabel('t')

ylabel('y')

legend ('Original Data','S(t)', 'location', 'best')

plot(x, y, 'ro', Rt, St)

title 'S(t) vs R(t)'

xlabel('R(t)')

ylabel('S(t)')

legend ('Original Data','S(t) vs R(t)', 'location', 'best')

% b)

x2 = [2.75 1.3 -0.25 0.0 0.25 -1.3 -2.5 -1.3 0.25 0.0 -0.25 1.3 2.75];

y2 = [-1.0 -0.75 0.8 2.0 0.8 -0.25 0.0 0.25 -1.3 -2.5 -1.3 -0.25 -1.0];

t2 = [-3 -2.5 -2 -1.5 -1 -0.5 0 0.5 1 1.5 2 2.5 3];

xx2 = linspace(-3, 3, 1000);

pp3 = spline(t2, x2);

Rt2 = ppval(pp3, xx2)

pp4 = spline(t2, y2);

St2 = ppval(pp4, xx2)

plot(x2, y2, 'bo', xx, Rt2)

title 'x = R\_2(t)'

xlabel('t')

ylabel('x')

legend ('Original Data','R\_2(t)', 'location', 'best')

plot(x2, y2, 'bo', xx, St2)

title 'y = S\_2(t)'

xlabel('t')

ylabel('y')

legend ('Original Data','S\_2(t)', 'location', 'best')

plot(x2, y2, 'bo', Rt2, St2)

title 'R\_2(t) vs S\_2(t)'

xlabel('R\_2(t)')

ylabel('S\_2(t)')

legend ('Original Data','R\_2(t) vs S\_2(t)', 'location', 'best')

% c)

[ylist1] = perspline(t2, x2, 1);

[ylist2] = perspline(t2, y2, 1);

[ylist3] = perspline(ylist1, ylist2, 0);

% d)

%figure('position', get(0,'screensize')) % biggest window possible

%axes('position', [0 0 1 1]) % domain [0,1] x [0,1]

%axis square % x,y axes equal

%[x,y] = ginput; % record mouse clicks

%close % get rid of huge window

%save mydatafile.mat x y % save the data points

% I renamed the variables x and y to xd and yd in the variable editor

td = linspace(0, 1, length(xd))

[ylist1] = perspline(td, xd', 1);

[ylist2] = perspline(td, yd', 1);

[ylist3] = perspline(ylist1, ylist2, 0);