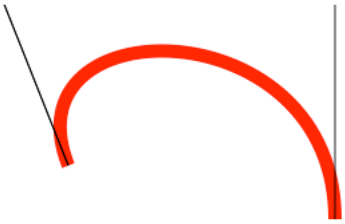
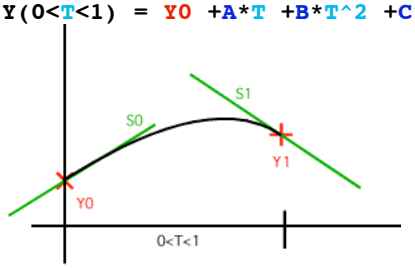
==========Popular_extrapolation_methods==========



Bézier Splines

Popular extrapolation methods being used by the graphic industry are Bézier Splines.
This type of curve fit is usually applied to two dimensions.
The simplest one dimensional curve fit method is shown below.
A single value Y will be made to follow a simple polynomial equation.

Y(0<T<1) = Y0 +A*T +B*T^2 +C*T^3



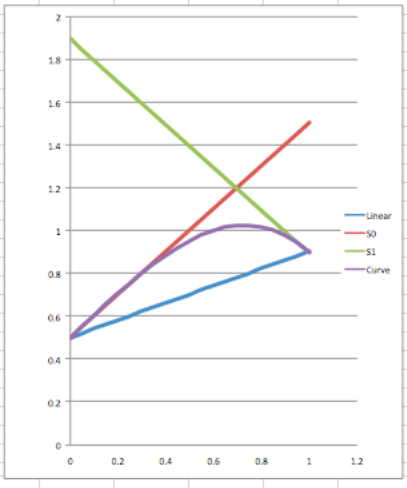
The assumption is that y has a start and stop point, and also has a start and stop slope.
The spread sheet pages below show the equations for all terms (See source zip file).
The value T always varies from 0 to 1.
But the value for Y0, Y1, S0, and S1 can be changed to anything.
And the results to all curves can be plotted.

Y = Y0 + S0*T + B*T^2 + C*T^2

Y0	0.5	
Y1	0.9	
S0	1	
S1	-1	
B	0.2	= 3*(Y1-Y0) -2*S0 -S1
C	-0.8	= 2*(Y0-Y1) +S0 +S1

Y = Y0 + S0*T + B*T^2 + C*T^2

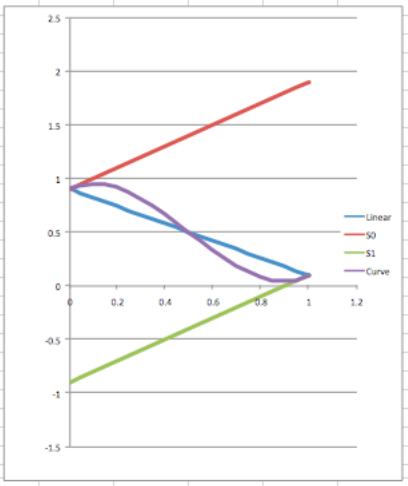
T	Linear	S0	S1	Curve
0	0.5	0.5	1.9	**0.5**
0.05	0.52	0.55	1.85	**0.5504**
0.1	0.54	0.6	1.8	**0.6012**
0.15	0.56	0.65	1.75	**0.6518**
0.2	0.58	0.7	1.7	**0.7016**
0.25	0.6	0.75	1.65	**0.75**
0.3	0.62	0.8	1.6	**0.7964**
0.35	0.64	0.85	1.55	**0.8402**
0.4	0.66	0.9	1.5	**0.8808**
0.45	0.68	0.95	1.45	**0.9176**
0.5	0.7	1	1.4	**0.95**
0.55	0.72	1.05	1.35	**0.9774**
0.6	0.74	1.1	1.3	**0.9992**
0.65	0.76	1.15	1.25	**1.0148**
0.7	0.78	1.2	1.2	**1.0236**
0.75	0.8	1.25	1.15	**1.025**
0.8	0.82	1.3	1.1	**1.0184**
0.85	0.84	1.35	1.05	**1.0032**
0.9	0.86	1.4	1	**0.9788**
0.95	0.88	1.45	0.95	**0.9446**
1	0.9	1.5	0.9	**0.9**



Y0	0.9
Y1	0.1
S0	1
S1	1
B	-5.4 = 3*(Y1-Y0) -2*S0 -S1
C	3.6 = 2*(Y0-Y1) +S0 +S1

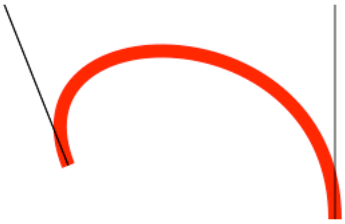
Y = Y0 + S0*T + B*T^2 + C*T^2

T	Linear	S0	S1	Curve
0	0.9	0.9	-0.9	**0.9**
0.05	0.86	0.95	-0.9	**0.937**
0.1	0.82	1	-0.8	**0.9496**
0.15	0.78	1.05	-0.8	**0.9407**
0.2	0.74	1.1	-0.7	**0.9128**
0.25	0.7	1.15	-0.7	**0.8688**
0.3	0.66	1.2	-0.6	**0.8112**
0.35	0.62	1.25	-0.6	**0.7429**
0.4	0.58	1.3	-0.5	**0.6664**
0.45	0.54	1.35	-0.5	**0.5846**
0.5	0.5	1.4	-0.4	**0.5**
0.55	0.46	1.45	-0.4	**0.4155**
0.6	0.42	1.5	-0.3	**0.3336**
0.65	0.38	1.55	-0.3	**0.2572**
0.7	0.34	1.6	-0.2	**0.1888**
0.75	0.3	1.65	-0.2	**0.1313**
0.8	0.26	1.7	-0.1	**0.0872**
0.85	0.22	1.75	-0	**0.0593**
0.9	0.18	1.8	0	**0.0504**
0.95	0.14	1.85	0.05	**0.0631**
1	0.1	1.9	0.1	**0.1**



The two screens shots above show all the details for different endpoints and slopes. Notices how the interpolation process resembles a digital filter.

==========Done_In_Javascript==========



```
<!DOCTYPE HTML>
<html>
<head>
<script>
window.onload = function(){
    var canvas            = document.getElementById("myCanvas");
    var context           = canvas.getContext("2d");
    var startX            = 188;
    var startY            = 130;
    var controlX1         = 140;
    var controlY1         = 10;
    var controlX2         = 388;
    var controlY2         = 10;
    var endX              = 388;
    var endY              = 170;
    context.moveTo(         startX,    startY);
    context.bezierCurveTo( controlX1, controlY1, controlX2, controlY2, endX, endY);
    context.lineWidth     = 10;
    context.strokeStyle   = "red"; // line color
    context.stroke();
    context.beginPath();   // view curve points
    context.moveTo(         startX, startY);
    context.lineTo(         controlX1, controlY1);
    context.lineWidth     = 1;
    context.strokeStyle   = "black"; // line color
    context.stroke();
    context.moveTo(         controlX2, controlY2);
    context.lineTo(         endX, endY);
    context.lineWidth     = 1;
    context.strokeStyle   = "black"; // line color
    context.stroke();
};
</script>
</head>
<body>
<canvas id="myCanvas" width="578" height="200">
</canvas>
</body>
</html>
```

8.18.11_3.20PM
dsauersanjose@aol.com
Don Sauer