The Need for Spline Interpolation.

NOTE: This worksheet demonstrates the use of Maple to show why we need to understand spline interpolation. It illustrates how interpolation using splines can be more accurate when compared to interpolation using polynomials.

⊟ Introduction

The following example illustrates the need for spline interpolation as opposed to using polynomial interpolation. In the year 1901, Runge tried to explain that higher order interpolation is a bad idea.

He took a simple function $f(x) = \dfrac{1}{(1 + 25x^2)}$ and chose equidistantly spaced data points to interpolate this function. The same function is used in the following example and is interpolated using polynomial interpolation and cubic spline interpolation.

```
> restart;
```
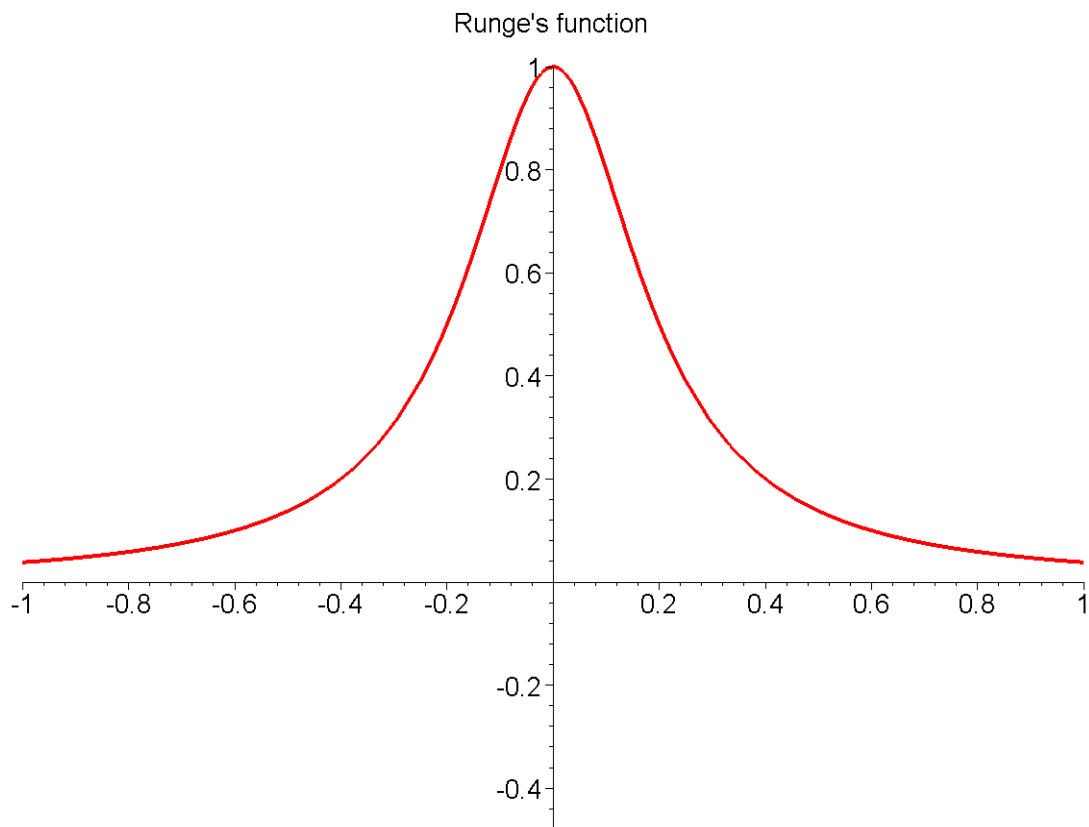Warning, the protected names norm and trace have been redefined and unprotected

⊟ Section I : Data.

The points chosen in this example are 9 equidistant points in [-1,1]. They are [-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1].

Plotting Runge's Function:

```
> fRunge:=x->1/(1+25*x^2);
```

$$fRunge := x \rightarrow \frac{1}{1 + 25\,x^2}$$

```
> plot(fRunge,-1..1,-0.5..1,thickness=4,title="Runge's function");
```

Runge's function

Section II: Polynomial Interpolation.

The function is interpolated using 9 equidistant data points in [-1,1] to obtain an 8th order polynomial.   By looking at the plot of the original function and the 8th order polynomial, you can see that the polynomial interpolation does not accurately represent the function. One may think that choosing more points would help in alleviating this problem, but in fact it makes it worse.

```
> poly_fn:=interp([-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1],[fRunge
  (-1),fRunge(-0.75),fRunge(-0.5),fRunge(-0.25),fRunge(0),fRunge(
  0.25),fRunge(0.5),fRunge(0.75),fRunge(1)],t);
```

$poly\_fn := 53.68930043\ t^8 - 0.13\ 10^{-6}\ t^7 + 1.000000000 - 102.8150104\ t^6 + 0.16\ 10^{-8}\ t$
$\quad + 0.19\ 10^{-6}\ t^5 - 13.20303455\ t^2 + 61.36720611\ t^4 - 0.3\ 10^{-7}\ t^3$

```
> poly_fn:=t->interp([-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1],[fRu
  nge(-1),fRunge(-0.75),fRunge(-0.5),fRunge(-0.25),fRunge(0),fRun
  ge(0.25),fRunge(0.5),fRunge(0.75),fRunge(1)],t):
```

```
> plot([fRunge,poly_fn],-1..1,-1..1,thickness=4,color=[red,green]
  ,legend=["Runge's Function","8th order Polynomial"]);
```
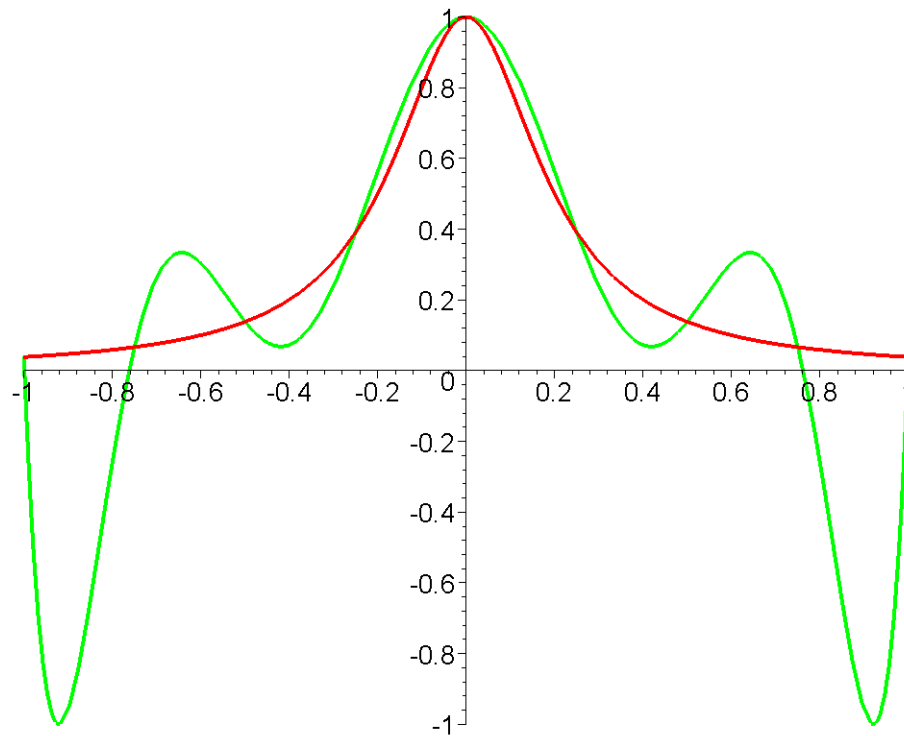
Runge's Function

8th order Polynomial

Section III: Cubic Spline Interpolation.

The function is now interpolated using cubic splines for the same 9 equidistant data points in the range of [-1,1] as used for polynomial interpolation.

```
> fcubic:=spline([-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1],[fRunge(
  -1),fRunge(-0.75),fRunge(-0.5),fRunge(-0.25),fRunge(0),fRunge(0
  .25),fRunge(0.5),fRunge(0.75),fRunge(1)],t,cubic);
```

$fcubic := \{$

$\quad 0.1009729044 + 0.0625113659000000044 \, t + 0.787242339299999982 \, (t+1)^3 , t < \text{-}0.75$

$\quad 0.2239795199 + 0.210119304600000001 \, t + 0.590431754647422702 \, (t+0.75)^2$

$\quad - 1.14501233899999999 \, (t+0.75)^3 , t < \text{-}0.5$

$\quad 0.2832537187 + 0.290645368299999984 \, t - 0.268327499789690670 \, (t+0.5)^2$

$\quad + 12.5710076599999994 \, (t+0.5)^3 , t < \text{-}0.25$

$\quad 1.018630291 + 2.51354555399999979 \, t + 9.15992824451133992 \, (t+0.25)^2$
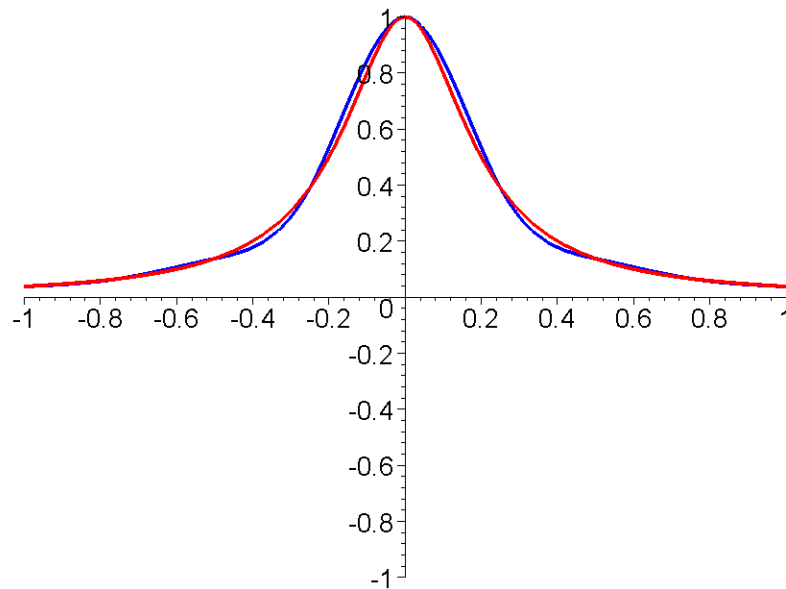
$\quad - 37.8320515999999998 \, (t+0.25)^3 , t < 0$

$\quad 1. - 19.2141104622556718 \, t^2 + 37.8320515999999998 \, t^3 , t < 0.25$

$1.018630291 - 2.51354555499999988\,t + 9.15992824451134170\,(t - 0.25)^2$

$\quad - 12.5710076599999994\,(t - 0.25)^3\ ,\ t < 0.5$

$0.2832537186 - 0.290645368199999976\,t - 0.268327499789691504\,(t - 0.5)^2$

$\quad + 1.14501233889999999\,(t - 0.5)^3\ ,\ t < 0.75$

$0.2239795199 - 0.210119304499999993\,t + 0.590431754647422924\,(t - 0.75)^2$

$\quad - 0.787242339299999982\,(t - 0.75)^3\ ,\ otherwise$

```
> fcubic:=t->spline([-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1],[fRun
  ge(-1),fRunge(-0.75),fRunge(-0.5),fRunge(-0.25),fRunge(0),fRung
  e(0.25),fRunge(0.5),fRunge(0.75),fRunge(1)],t,cubic):
```

```
> plot([fRunge,fcubic],-1..1,-1..1,thickness=4,color=[red,blue],l
  egend=["Runge's Function","Cubic Spline"]);
```


Section IV: Comparison.

Below is a plot to compare the interpolated functions obtained using an 8th order polynomial and a cubic spline with the actual Runge's Function :

```
> plot([fRunge,poly_fn,fcubic],-1..1,-1..1,thickness=4,color=[red
  ,green,blue],title="Comparison of the 8th Order Polynomial and
  Cubic Spline with Runge's function",legend=["Runge's
  Function","8th order Polynomial","Cubic Spline"]);
```

Comparison of the 8th Order Polynomial and Cubic Spline with Runge's function
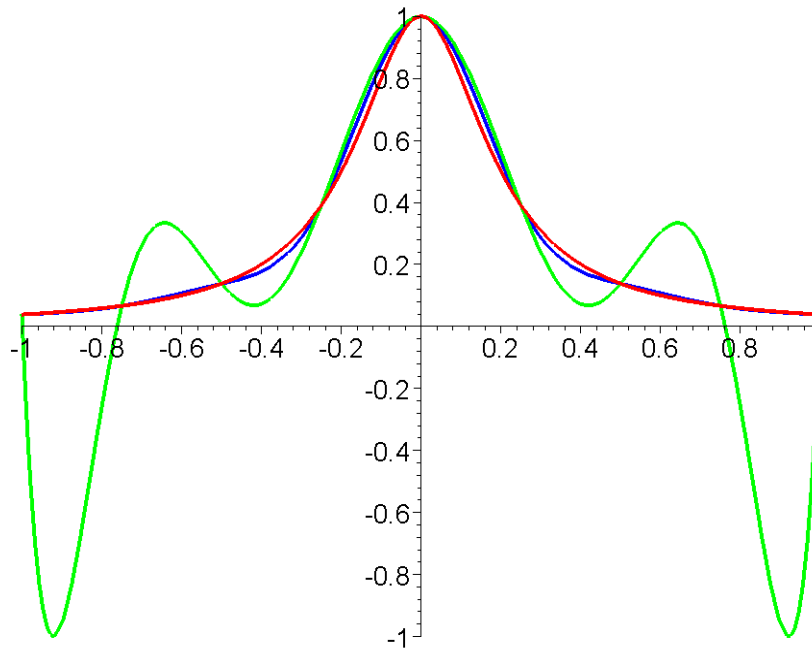
	Runge's Function
	8th order Polynomial
	Cubic Spline

To better understand the difference between the 8th order polynomial and the cubic spline, let us compare the value of the functions with the original Runge's function at any point x which is not specified, say, x = 0.6.

Value from Runge's function at x=0.6
```
> fRunge(0.6);
```
$$0.1000000000$$

Value from polynomial interpolant at x=0.6
```
> poly_fn(0.6);
```
$$0.3049344276$$

Value from cubic spline interpolant at x=0.6
```
> fcubic(0.6);
```
$$0.1073282350$$

```
> 
```

Section V: Conclusion.

Maple helped us to apply our knowledge of numerical methods of interpolation to illustrate that for approximating Runge's function, cubic spline interpolation is better than polynomial interpolation.

Can you repeat the example by choosing 20 equidistant data points in [-1,1] and compare the results obtained from polynomial interpolation and spline interpolation for a value of x = -0.45?

References:

[1] *Autar Kaw, Holistic Numerical Methods Institute, See http://numericalmethods.eng.usf.edu/mws/ind/05inp/mws_ind_inp_spe_needspline.pdf*