Higher Order Interpolation is a Bad Idea !

NOTE: This worksheet demonstrates the use of Maple to show how higher interpolation can be a bad idea. It illustrates how choosing more data points can result in highly oscillatory polynomial functions.

Introduction

The following example illustrates why higher order polynomial interpolation, that is, interpolating using high number of data points, is a bad idea. In 1901, Carl Runge published his work on dangers of higher order polynomial interpolation. He took a simple looking function $f(x) = \frac{1}{(1+25x^2)}$ on the interval of [-1,1]. He took data points equidistantly spaced in [-1,1], and then interpolated the data points with polynomials. He found that as he took more points, the polynomials and the original curve differed considerably in their value from each other.

```
> restart;
```

Section I : Data.

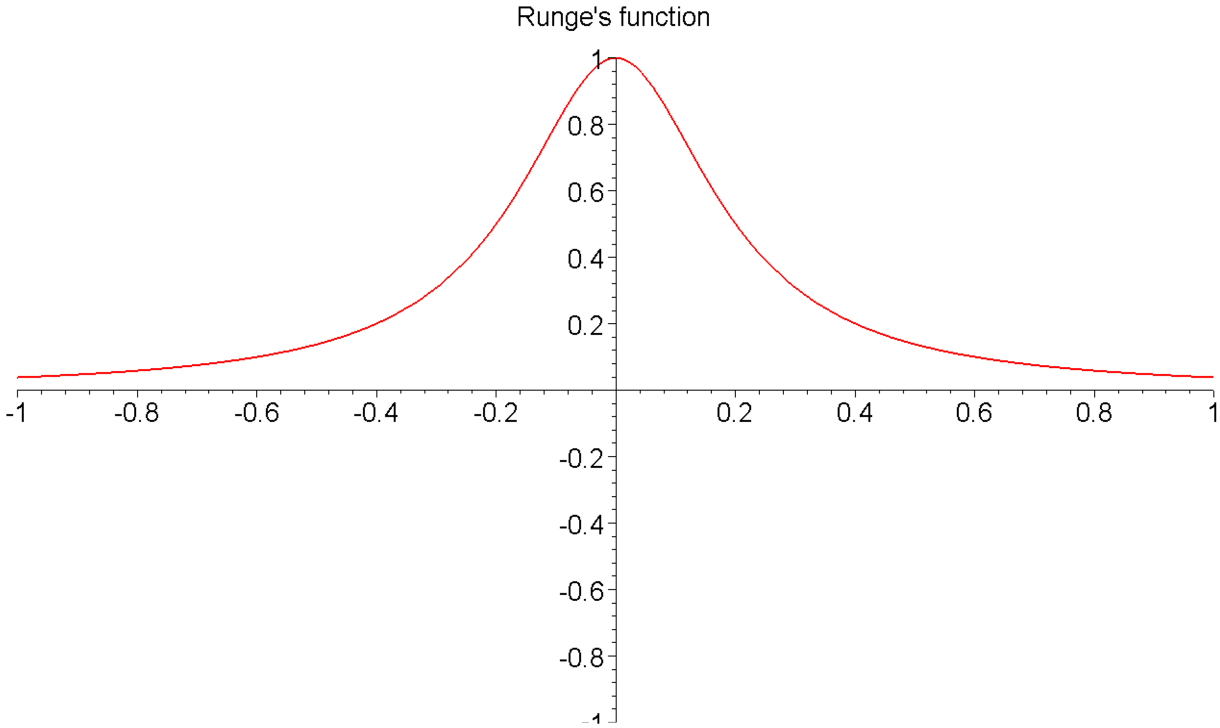
Runge's function is given by:

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

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

Plotting Runge's Function:

```
> plot(fRunge,-1..1,-1..1,thickness=2,title="Runge's function");
```



Section II: Polynomial interpolation for 5 data points.

Let us first interpolate approximate Runge's function using polynomial interpolation. For interpolation 5 equidistant data points, [-1, -0.5, 0, 0.5, 1] are chosen in [-1,1] . This will give us a 4th order polynomial.

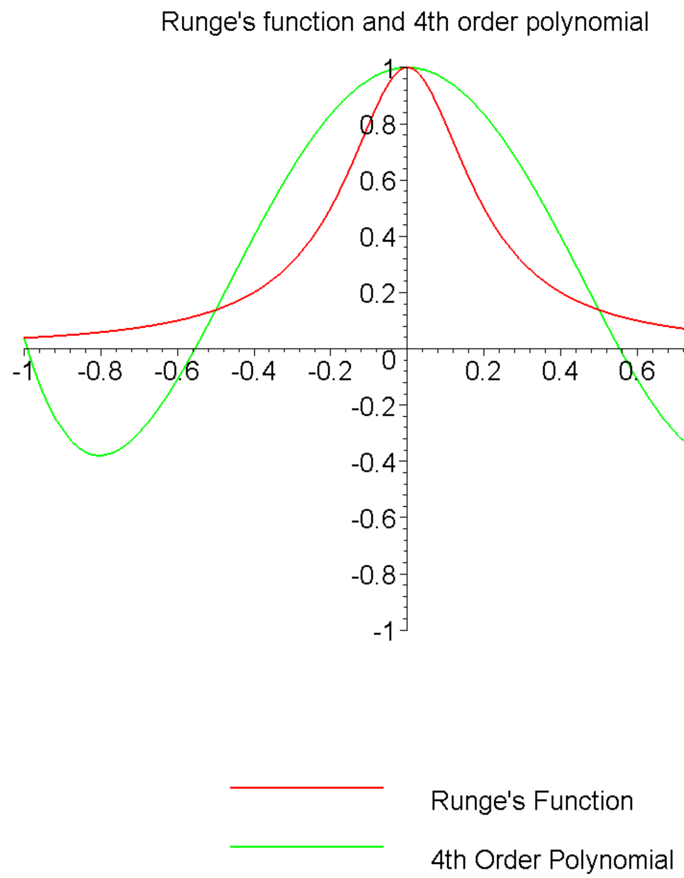
```
> poly5:=interp([-1, -0.5, 0, 0.5, 1],[fRunge(-1),
  fRunge(-0.5),fRunge(0),fRunge(0.5),fRunge(1)],t);
```

$$poly5 := 3.315649864\ t^4 - 0.2\ 10^{-8}\ t^3 + 1.000000000 - 4.277188326\ t^2 + 0.16\ 10^{-8}\ t$$

```
> poly5:=t->interp([-1, -0.5, 0, 0.5, 1],[fRunge(-1),
  fRunge(-0.5),fRunge(0),fRunge(0.5),fRunge(1)],t):
```

let us now plot it against the actual Runge's function.

```
> plot([fRunge,poly5],-1..1,-1..1,thickness=2,title="Runge's
  function and 4th order polynomial",legend=["Runge's
  Function","4th Order Polynomial"]);
```



Section III: Polynomial interpolation for 9 data points.

Runge' function is now approximated by choosing 9 equidistant data points, [-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1] in [-1,1]. Interpolating with these values would give us an 8th order polynomial.

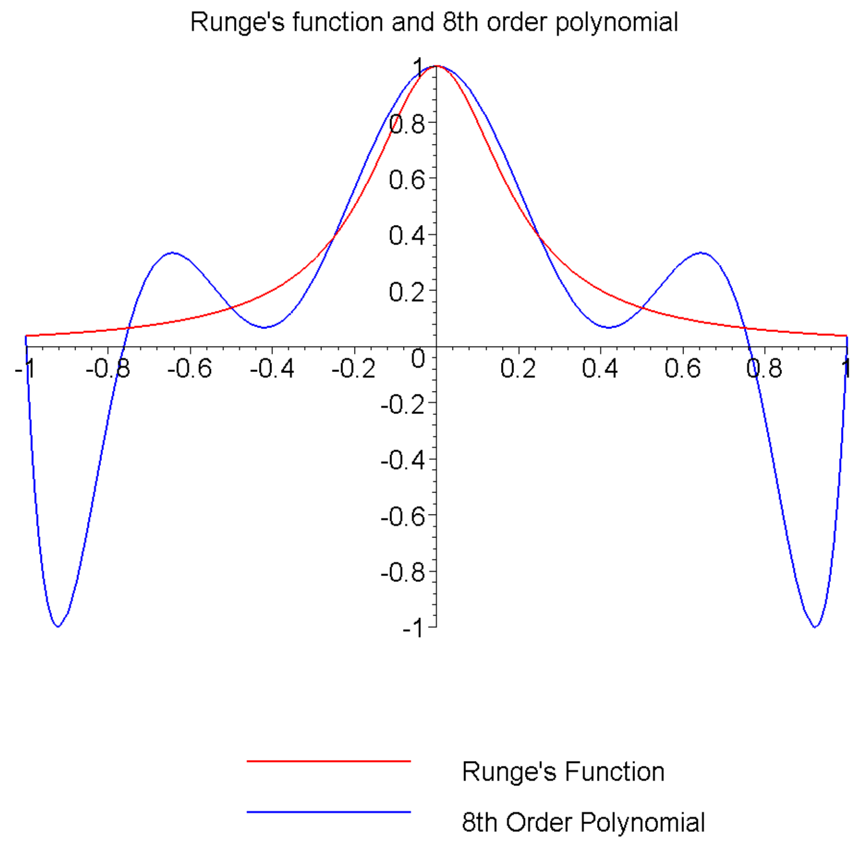
```
> poly9:=interp([-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1],[fRunge(-1),fRunge(-0.75),fRunge(-0.5),fRunge(-0.25),fRunge(0),fRunge(0.25),fRunge(0.5),fRunge(0.75),fRunge(1)],t);
```

$$poly9 := 53.68930043\, t^8 - 0.13\ 10^{-6}\, t^7 + 1.000000000 - 102.8150104\, t^6 + 0.16\ 10^{-8}\, t + 0.19\ 10^{-6}\, t^5 - 13.20303455\, t^2 + 61.36720611\, t^4 - 0.3\ 10^{-7}\, t^3$$

```
> poly9:=t->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):
```

Plotting the 8th order polynomial against Runge's function,

```
> plot([fRunge,poly9],-1..1,-1..1,thickness=2,color=[red,blue],title="Runge's function and 8th order polynomial",legend=["Runge's Function","8th Order Polynomial"]);
```



Section IV: Polynomial interpolation for 21 data points.

We will now interpolate Runge's function in [-1,1], by choosing 21 equidistant points, [-1,-0.9,-0.8,-0.7,-0.6,-0.5,-0.4,-0.3,-0.2,-0.1,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]. This will give us a 20th order polynomial.

```
> poly21:=interp([-1,-0.9,-0.8,-0.7,-0.6,-0.5,-0.4,-0.3,-0.2,-0.1,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],[fRunge(-1),fRunge(-0.9),fRunge(-0.8),fRunge(-0.7),fRunge(-0.6),fRunge(-0.5),fRunge(-0.4),fRunge(-0.3),fRunge(-0.2),fRunge(-0.1),fRunge(0),fRunge(0.1),fRunge(0.2),fRunge(0.3),fRunge(0.4),fRunge(0.5),fRunge(0.6),fRunge(0.7),fRunge(0.8),fRunge(0.9),fRunge(1)],t);
```

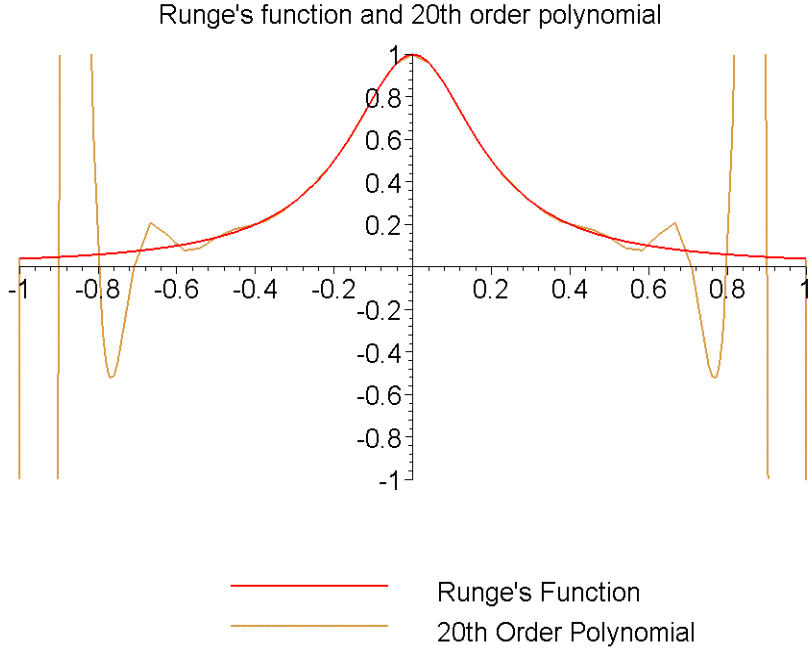
$$poly21 := -6119.218320\, t^6 + 0.0006977\, t^5 + 49317.51886\, t^8 - 0.010697\, t^7 + 0.1639175200\ 10^7\, t^{16} + 0.7207\, t^{17} - 0.2032\, t^{19} + 1.000000000 + 260178.2112\, t^{20} - 0.1012093365\ 10^7\, t^{18} + 757286.3442\, t^{12} + 0.7680\, t^{13} - 0.1442943287\ 10^7\, t^{14} - 1.0511\, t^{15} + 0.08308\, t^9 - 245249.1005\, t^{10} - 0.3397\, t^{11} + 0.1186\ 10^{-6}\, t - 0.00001826\, t^3 + 470.8461839\, t^4 - 24.14347937\, t^2$$

```
> poly21:=t->interp([-1,-0.9,-0.8,-0.7,-0.6,-0.5,-0.4,-0.3,-0.2,-0.1,0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1],[fRunge(-1),fRunge
```

```
(-0.9),fRunge(-0.8),fRunge(-0.7),fRunge(-0.6),fRunge(-0.5),fRun
ge(-0.4),fRunge(-0.3),fRunge(-0.2),fRunge(-0.1),fRunge(0),fRung
e(0.1),fRunge(0.2),fRunge(0.3),fRunge(0.4),fRunge(0.5),fRunge(0
.6),fRunge(0.7),fRunge(0.8),fRunge(0.9),fRunge(1)],t):
```

Plotting this polynomial against Runge's function,

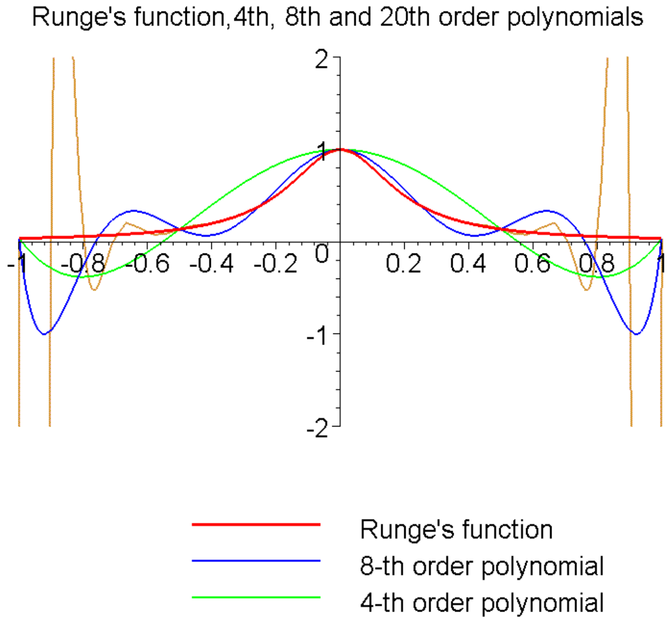
```
> plot([fRunge,poly21],-1..1,-1..1,thickness=2,color=[red,gold],t
itle="Runge's function and 20th order
polynomial",legend=["Runge's Function","20th Order
Polynomial"]);
```



Section V: Comparison.

Below is a plot to compare the interpolated polynomials obtained using 5, 9 and 21 data points, respectively, with the actual Runge's function :

```
> plot([fRunge,poly9,poly5,poly21],-1..1,-2..2,thickness=[3,2,2,2
],color=[RED,BLUE,GREEN,GOLD],title="Runge's function,4th, 8th
and 20th order polynomials",legend=["Runge's function","8-th
order polynomial","4-th order polynomial","20-th order
polynomial"]);
```



>

>

Section VI: Conclusion.

Maple helped us to apply our knowledge of numerical methods of interpolation to see that higher order interpolation is a bad idea. As the interpolant order becomes higher, the difference between the values of the original function and the interpolated function are very different, especially close to the end points of -1 and +1.

Choose even order of polynomials of your own choice, and see the effect of the higher order interpolation at x=0.8 and x=0.

References:

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