

The Effect of Choice of Points on Polynomial Interpolation.

© 2003 Nathan Collier, Autar Kaw, Jai Paul, University of South Florida, kaw@eng.usf.edu, <http://numericalmethods.eng.usf.edu/mws>.

NOTE: This worksheet demonstrates the use of Maple to show how the choice of points for interpolation affects the polynomial interpolation of a function.

- Introduction

The following example illustrates the difference in interpolation curves due to the selection of points. In 1901, Carl Runge published his work on dangers of higher order interpolation. He took a

simple looking function, $f(x) = \frac{1}{(1+25x^2)}$ on the interval $[-1,1]$. He took points equidistantly

spaced in $[-1,1]$ and interpolated the points with polynomials. He found that as he took more points, the polynomials and the original curve differed considerably. However, he also discovered that if he took data points close to the ends of the interval $[-1,1]$, the problem of large differences between interpolated and actual values was less pronounced. This simulation shows you this phenomena.

> **restart;with(linalg):**

Warning, the protected names norm and trace have been redefined and unprotected

- Section I : Data.

Let us first chose points which are equidistantly placed in $[-1,1]$, and interpolate it. We will chose 11 points, $[-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1]$.

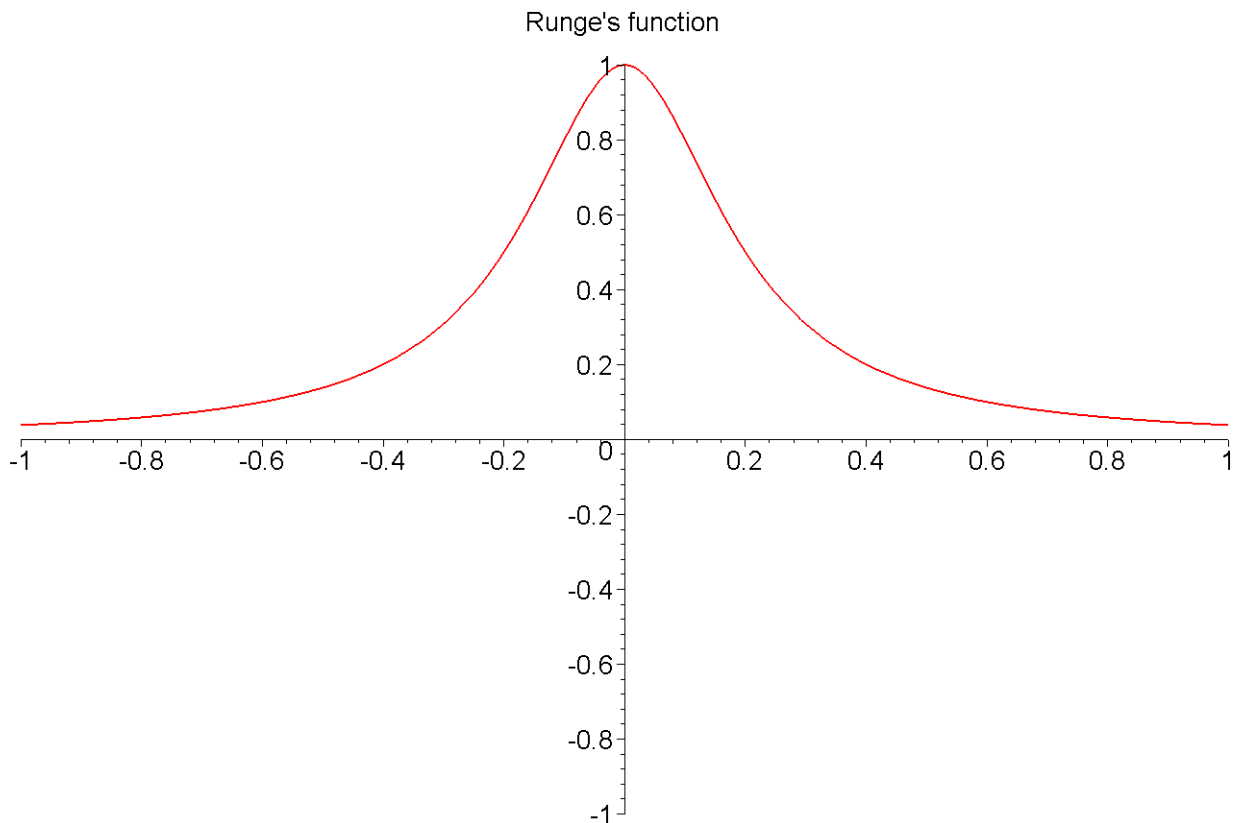
Runge's Function is given by:

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

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

Plotting the data:

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



Section II: Polynomial Interpolation with equidistantly spaced points.

Let us now interpolate Runge's function using polynomial interpolation in $[-1,1]$, choosing 11 equidistantly spaced data points, $[-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1]$. This will give us a 10th order polynomial.

```
> eq_poly:=interp([-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6,
0.8,
1], [fRunge(-1), fRunge(-0.8), fRunge(-0.6), fRunge(-0.4), fRunge(-0.2), fRunge(0), fRunge(0.2), fRunge(0.4), fRunge(0.6), fRunge(0.8), fRunge(1)], t);
```

```
eq_poly := -220.9417439 t10 - 0.12 10-5 t9 + 1.000000000 + 494.9095054 t8 - 0.20 10-8 t
+ 0.22 10-5 t7 - 16.85520363 t2 - 381.4338247 t6 + 0.11 10-6 t3 - 0.7 10-6 t5 + 123.3597288 t4
```

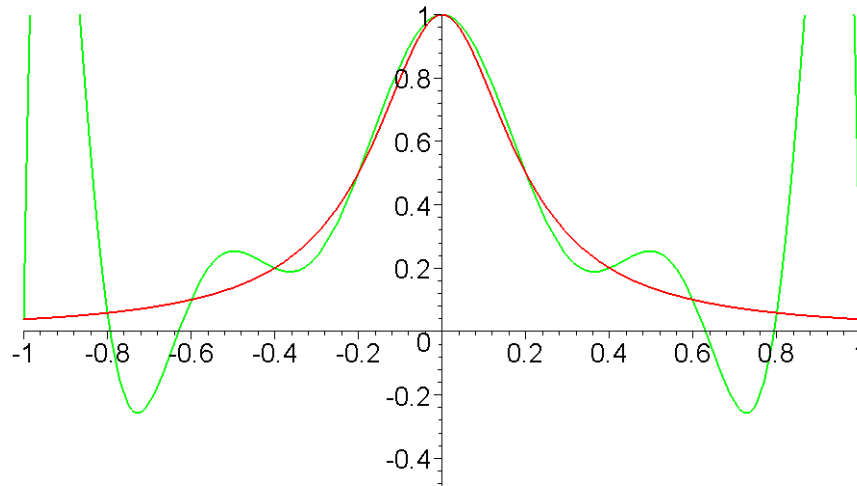
We will now plot this function against Runge's function.

```
> eq_poly:=t->interp([-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4,
0.6, 0.8,
1], [fRunge(-1), fRunge(-0.8), fRunge(-0.6), fRunge(-0.4), fRunge(-0.2), fRunge(0), fRunge(0.2), fRunge(0.4), fRunge(0.6), fRunge(0.8), fRunge(1)], t);
```

```
Runge(1), t):
```

```
> plot([fRunge,eq_poly],[-1..1,-0.5..1,thickness=2,color=[red,green],title="Plot of Runge's Function and Polynomial interpolated with equidistantly spaced points",legend=["Runge's function","Polynomial with equidistantly spaced points"]);
```

Plot of Runge's Function and Polynomial interpolated with equidistantly spaced points



— Runge's function
— Polynomial with equidistantly spaced points

Section III: Polynomial with more points at the end.

The idea of this function is to place more points at the ends of the interval than in the middle. To do this, first the half interval from -1 to 1 is divided to give 'n' points in [-1,1] ('n' needs to be odd). Then the number of divisions is (n-1), and the number of divisions are divided equally, that is (n-1)/2, between [-1,0] and [0,1]. For points from -1 to 0, starting with -1 as the first point, and assuming the next points are $-1+l$, $-1+2l$, $-1+4l$,, 0, where the distance between -1 and the next point is l , and the distance between consecutive points from -1 to 0 gets doubled after each point, the value of l is given by $l=1/[2^{\{(n-1)/2\}}-1]$.

The points from 0 to 1 are mirror images about the y-axis of points from -1 to 0.

```
> n:=11:  
> Xb:=matrix(n,1,0):  
d:=0:  
for i from 2 to ((n-1)/2)+1 do  
d:=d+2^(i-2):  
end do:  
l:=1/d:  
Xb[1,1]:=-1:  
for i from 2 to ((n-1)/2)+1 do
```

```

Xb[i,1]:=Xb[i-1,1]+2^(i-2)*1:
end do:
for i from (n-1)/2+2 to n do
Xb[i,1]:=Xb[i-1,1]+(2^(n-i))*1:
end do:

```

```

> Yb:=matrix(n,1,0):
for i from 1 to n do
Yb[i,1]:=fRunge(Xb[i,1]):
end do:

```

When x and y data and order is given, the following constructs the matrix whose inverse is needed to find coefficients of the polynomial which approximates the data.

```

> A:=matrix(n,n,0):
for i from 1 to n do
for j from 1 to n do
A[i,j]:=Xb[i,1]^(j-1):
end do:
end do:

```

This generates the coefficients for the polynomial that approximates the x and y data.

```

> M:=evalm(inverse(A) &* Yb):

```

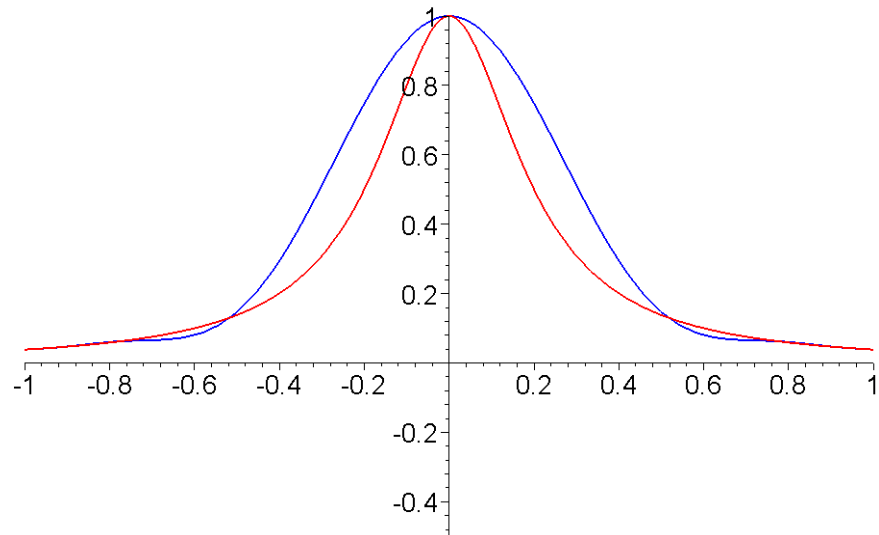
When given the polynomial order and specific x, this procedure uses the above calculated coefficients to calculate the approximated $f(x)$

```

> f2:=proc(x)
local i,c,d;
c:=0:
for i from 1 to n do
d:=M[i,1]*x^(i-1)+c:
c:=d:
end do:
d;
end proc:
> plot([fRunge,f2],-1..1,-0.5..1,thickness=2,color=[RED,BLUE],title="Runge's function and interpolated polynomial with more points near the ends of the interval [-1,1]",legend=["Runge's function","Polynomial with more points at the end"]);

```

Runge's function and interpolated polynomial with more points near the ends of the interval [-1,1]



— Runge's function
— Polynomial with more points at the end

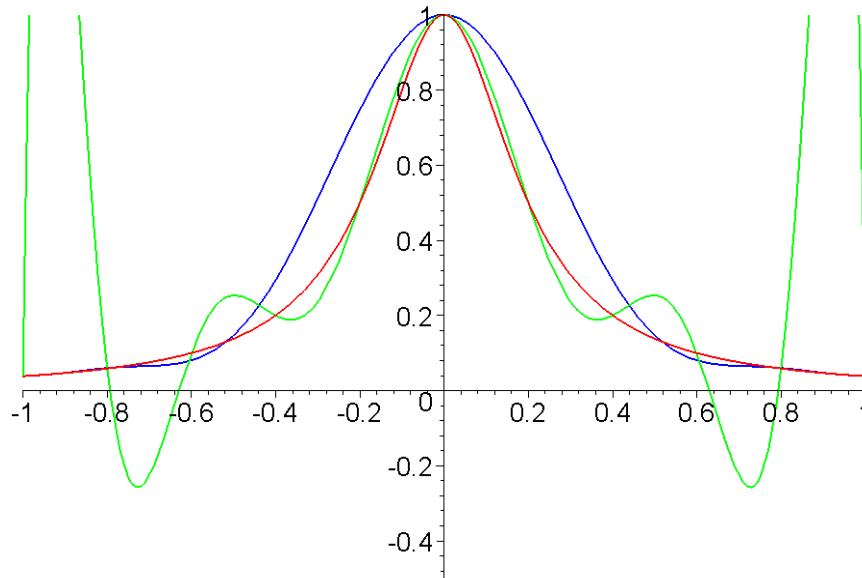
[>

Section IV: Comparison.

Below is a plot to compare the polynomial obtained by interpolating using equidistantly placed data points and the polynomial obtained by interpolation using more data points near the end points of -1 and 1.

```
> plot([fRunge,eq_poly,f2],-1..1,-0.5..1,thickness=2,color=[RED,G  
REEN,BLUE],title="Runge's function and Interpolated polynomials  
with equidistant data points and with more data points near the  
ends of the interval [-1,1]",legend=["Runge's  
function","Polynomial with equidistantly spaced data  
points","Polynomial with more points at the end"]);
```

Runge's function and Interpolated polynomials with equidistant data points and with more data points near the



— Runge's function
 — Polynomial with equidistantly spaced data points
 — Polynomial with more points at the end

To better understand the difference between the polynomial obtained by interpolating with equidistantly spaced data points and the polynomial obtained by interpolating more data points at the end of the interval, let us find the value of these functions and the actual value from Runge's function for any x that is not specified, say, $x = 0.5$:

Value of the original function at $x=0.5$

```
> fRunge(0.5);
```

0.1379310345

Value of the polynomial interpolant at $x=0.5$ with equidistant points chosen for interpolation

```
> eq_poly(0.5);
```

0.2537554594

Value of the polynomial interpolant at $x=0.5$ with more points chosen close to the end points.

```
> f2(0.5);
```

0.1463771727

Section VI: Conclusion.

Maple helped us to apply our knowledge of numerical methods of interpolation to find a better interpolant for Runge's function by polynomial interpolation by choosing more data points at the

end of the interval as opposed to using equidistantly spaced data points. Although there is no rule for choosing points for interpolation for a better approximation, rules can be derived for particular functions (Davis, 1963; Kahaner, Moler and Nash, 1989)

Can you repeat the example by finding two polynomials choosing 21 points. The points for the first polynomial are obtained by using equidistantly spaced points and the points for the second polynomial are obtained using points concentrated towards the end in $[-1,1]$ domain. Compare the results for the two polynomials for a value of $x = -0.7$?

References:

[1] *Autar Kaw, Holistic Numerical Methods Institute, See*

http://numericalmethods.eng.usf.edu/mws/ind/05inp/mws_ind_inp_spe_choiceofpoints.pdf

[2] P. Davis, *Interpolation and Approximation*, Balisdell, New York, 1963.

[3] D. Kahaner, C. Moler, S. Nash, *Numerical Methods and Software*, Prentice Hall, New York, 1989.

Disclaimer: While every effort has been made to validate the solutions in this worksheet, University of South Florida and the contributors are not responsible for any errors contained and are not liable for any damages resulting from the use of this material.