

Some Interpolation Examples

Mth 351 Aug 7 2001

Bent E. Petersen

Filename: 351u2001_interp_examples.mws

In this worksheet we illustrate some of Maple's interpolation facilities.

```
> restart;
```

Example: Cubic Interpolation Spline for sin(x) and cos(x)

```
> XX1 := [0, Pi/4, Pi/2, 3*Pi/4, Pi, 5*Pi/4, 3*Pi/2, 7*Pi/4, 2*Pi];
```

$$XX1 := \left[0, \frac{1}{4}\pi, \frac{1}{2}\pi, \frac{3}{4}\pi, \pi, \frac{5}{4}\pi, \frac{3}{2}\pi, \frac{7}{4}\pi, 2\pi \right]$$

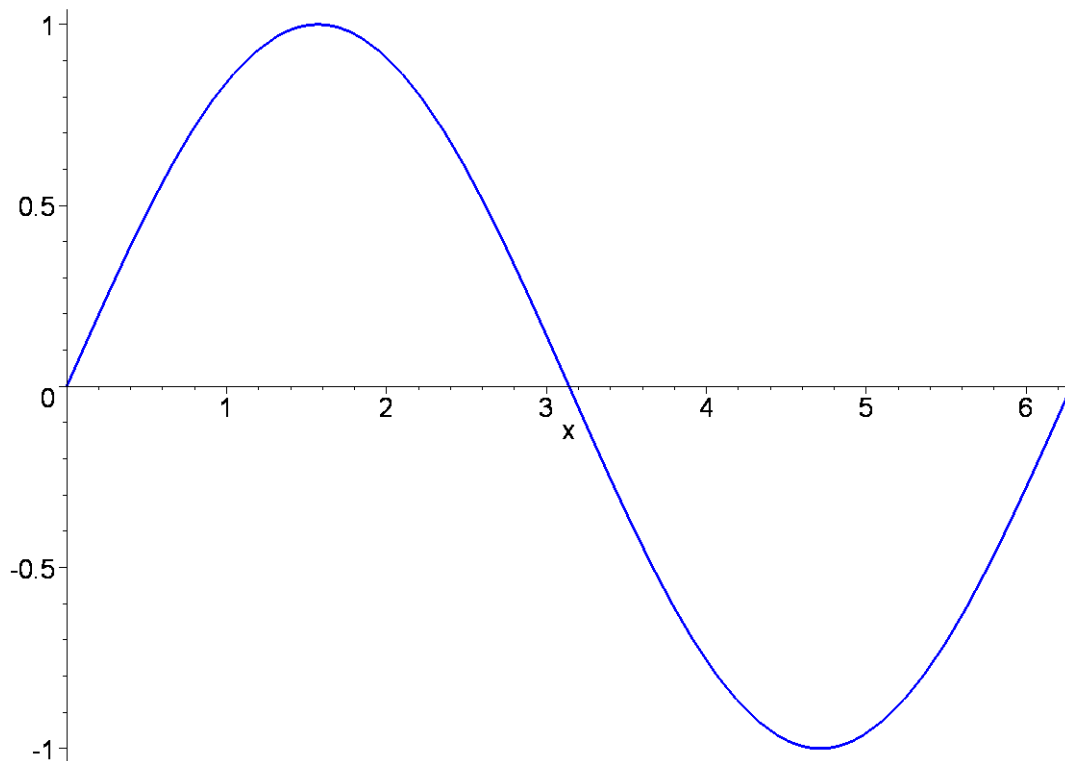
```
> YY1 := map(sin, XX1);
```

$$YY1 := \left[0, \frac{1}{2}\sqrt{2}, 1, \frac{1}{2}\sqrt{2}, 0, -\frac{1}{2}\sqrt{2}, -1, -\frac{1}{2}\sqrt{2}, 0 \right]$$

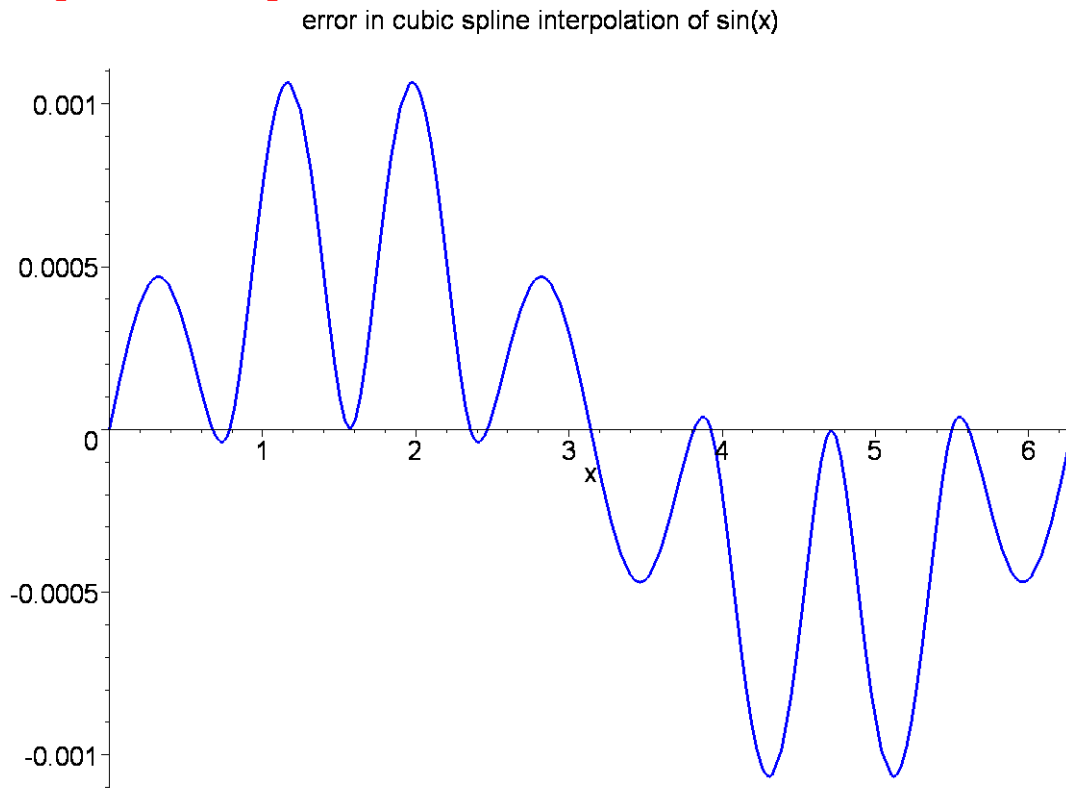
```
> ex1 := spline(XX1, YY1, x, cubic);
```

```
> plot(ex1, x=0..2*Pi, thickness=3, color=blue, title="cubic spline interpolation of sin(x)");
```

cubic spline interpolation of sin(x)



```
> plot(sin(x)-ex1,x=0..2*Pi,thickness=3,color=blue, title="error in  
cubic spline interpolation of sin(x)");
```



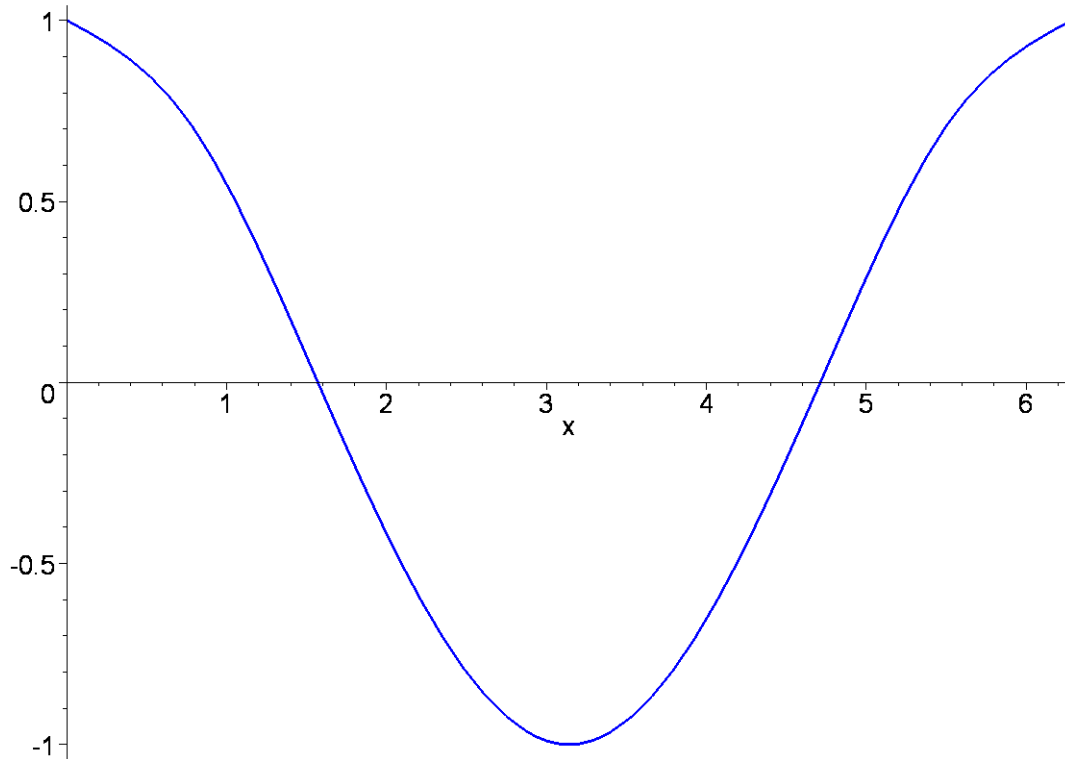
```
> YY2:=map(cos,XX1);
```

$$YY2 := \left[1, \frac{1}{2}\sqrt{2}, 0, -\frac{1}{2}\sqrt{2}, -1, -\frac{1}{2}\sqrt{2}, 0, \frac{1}{2}\sqrt{2}, 1 \right]$$

```
> ex2:=spline(XX1,YY2,x,cubic):
```

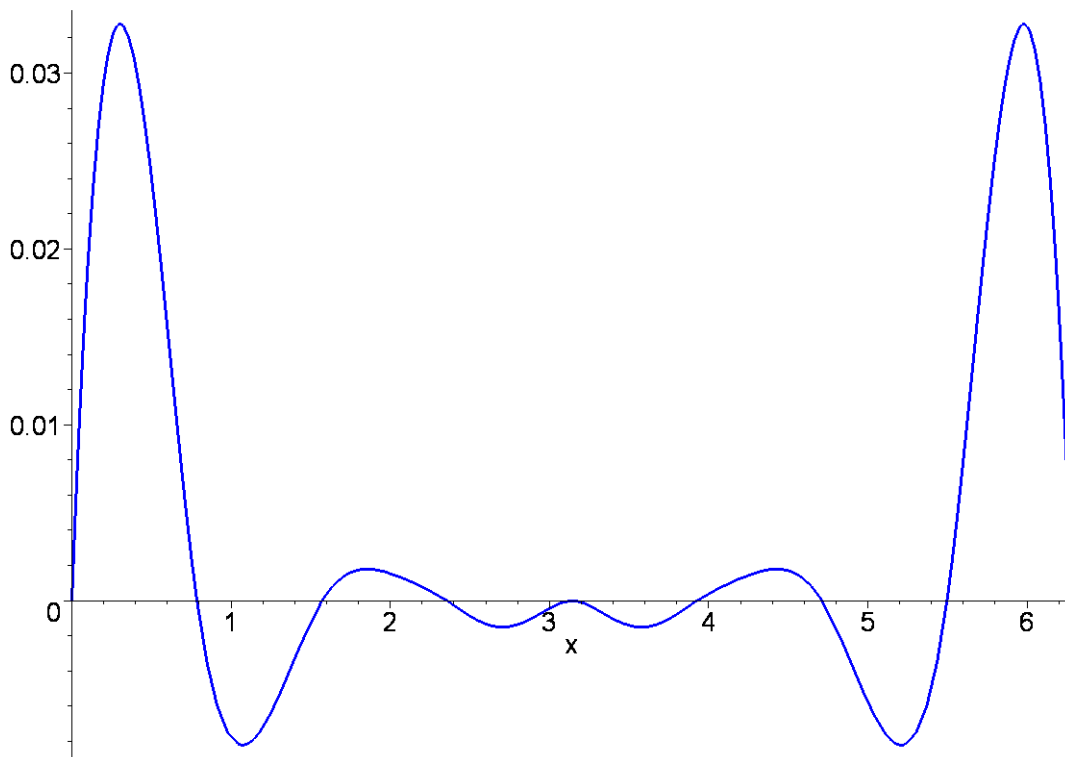
```
> plot(ex2,x=0..2*Pi,thickness=3,color=blue,title="cubic spline  
interpolation of cos(x)");
```

cubic spline interpolation of $\cos(x)$



```
> plot(cos(x) - ex2, x=0..2*Pi, thickness=3, color=blue, title="error in  
cubic spline interpolation of cos(x)");
```

error in cubic spline interpolation of $\cos(x)$



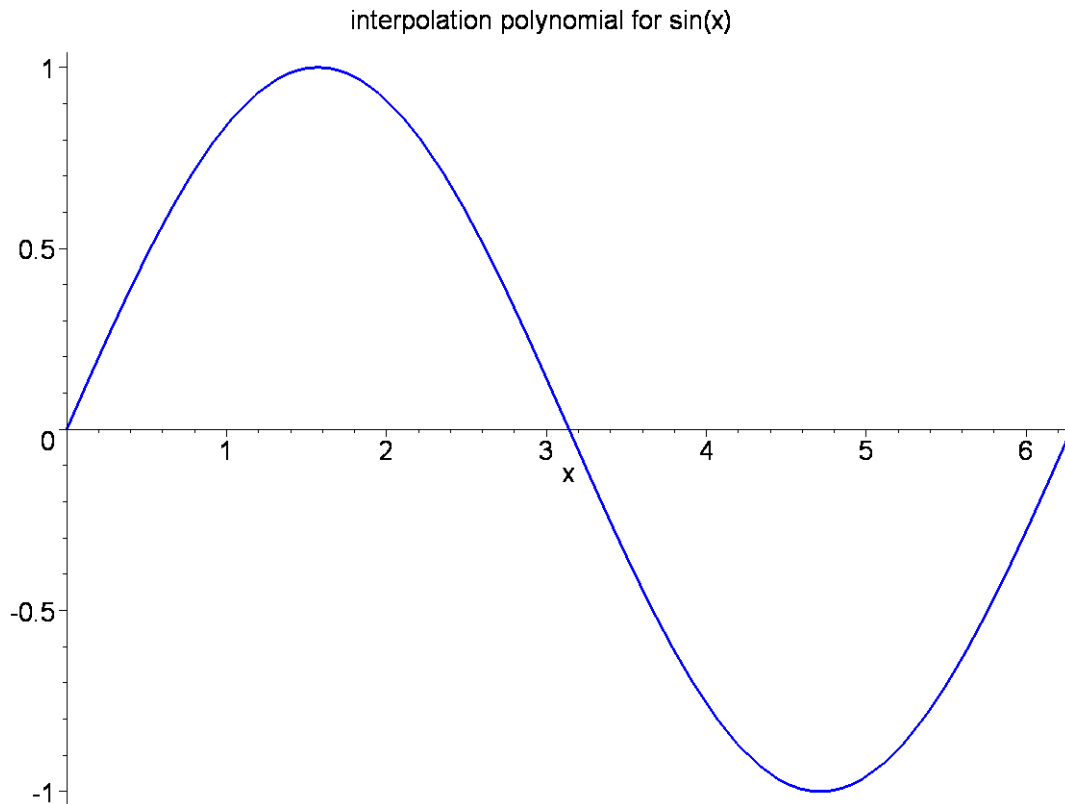
Note because the second derivative of $\sin(x)$ vanishes at 0 and at π we expect the natural cubic

spline for $\sin(x)$ to have somewhat smaller error than the natural cubic spline for $\cos(x)$. That is what we observe above.

Example: Interpolation Polynomial for $\sin(x)$ and $\cos(x)$

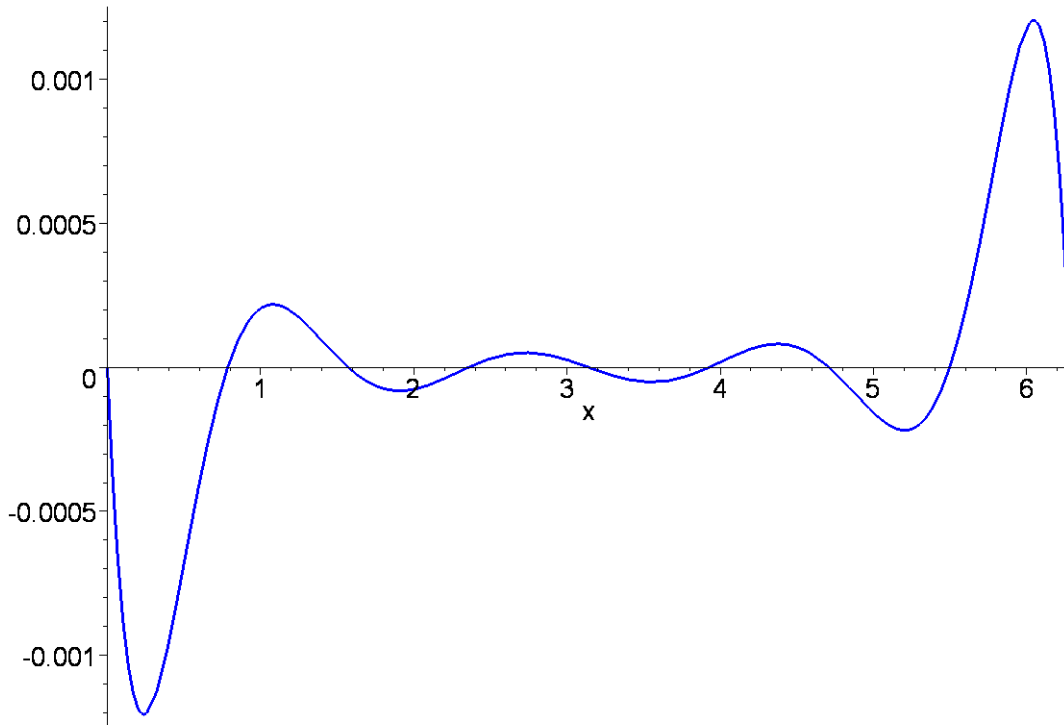
```
> exlp:=interp(XX1,YY1,x) :
```

```
> plot(exlp,x=0..2*Pi,thickness=3,color=blue, title="interpolation  
polynomial for sin(x)");
```



```
> plot(sin(x)-exlp,x=0..2*Pi,thickness=3,color=blue, title="error in  
interpolation polynomial for sin(x)");
```

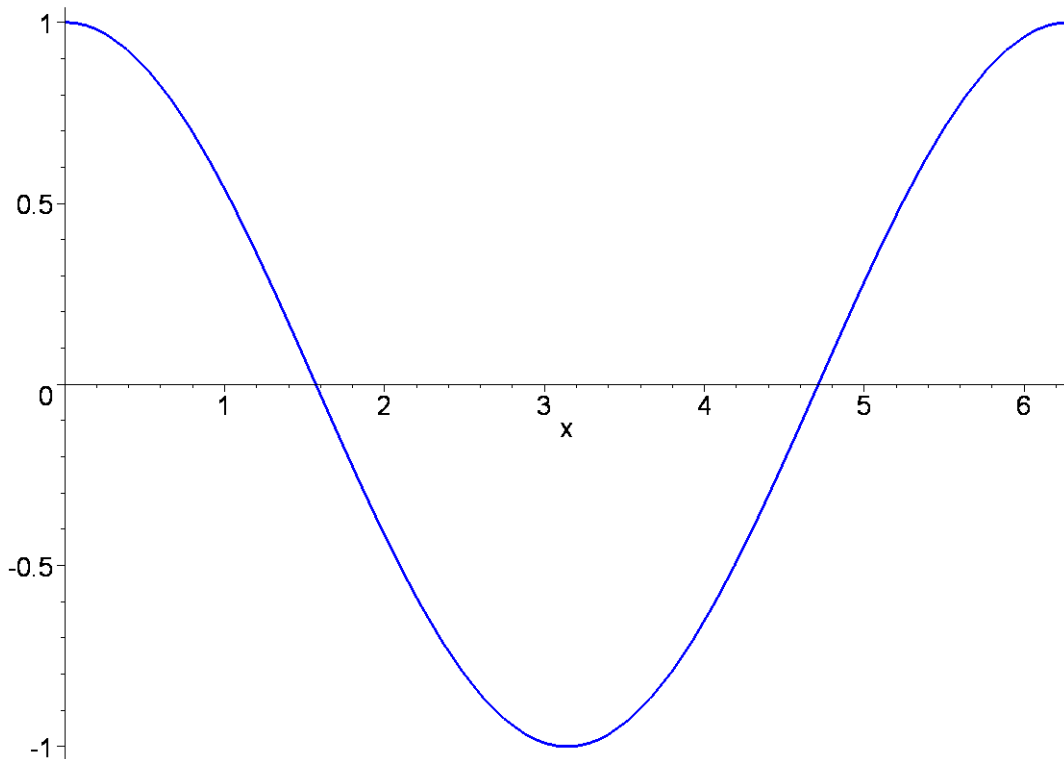
error in interpolation polynomial for sin(x)



```
> ex2p:=interp(XX1,YY2,x):
```

```
> plot(ex2p,x=0..2*Pi,thickness=3,color=blue, title="interpolation  
polynomial for cos(x)");
```

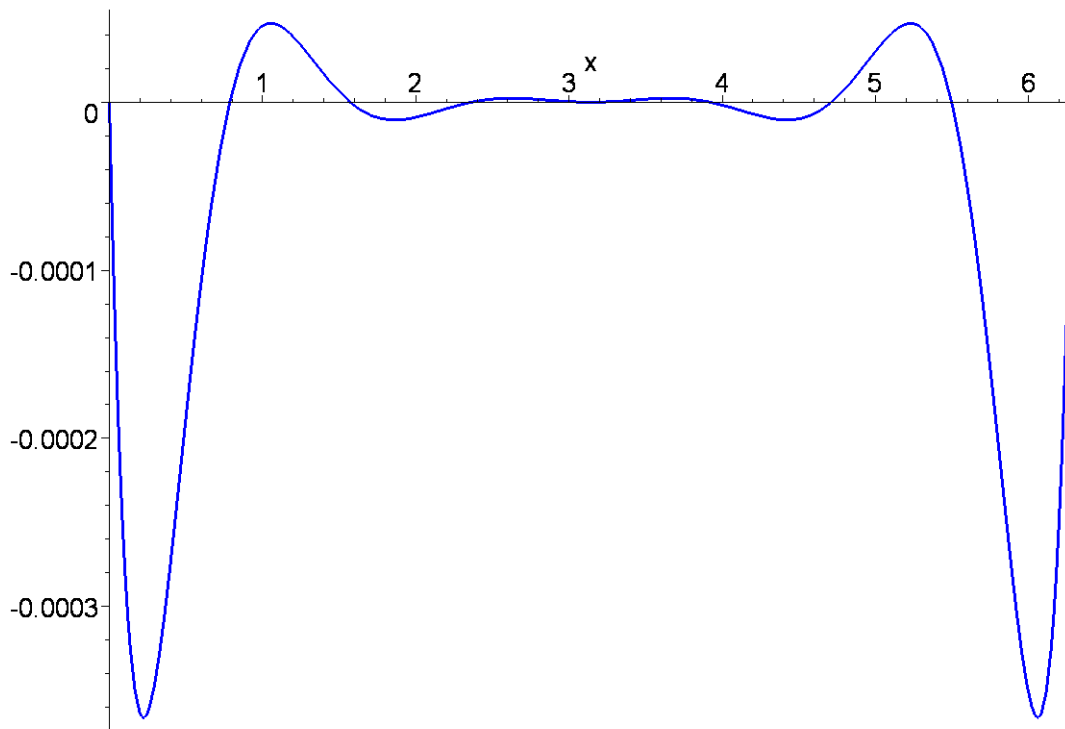
interpolation polynomial for cos(x)



```
> plot(cos(x)-ex2p,x=0..2*Pi,thickness=3,color=blue, title="error in
```

```
interpolation polynomial for cos(x)");
```

error in interpolation polynomial for cos(x)



In the above examples we see the interpolation polynomial has smaller error than the interpolation spline. The next example will demonstrate this is not always the case. One problem with interpolation polynomials is that they are of high degree if there are many nodes. Evaluating a polynomial of high degree runs a serious risk of loss of significance errors due to roundoff.

Example. A function pathological for equispaced interpolation

We use a do-loop to build a list of 21 equispaced nodes in $[-1,1]$

```
> a:=-1: XX2:=[]:for k from 1 to 21 do  
  XX2:=[op(XX2),a];a:=a+1/10;od:
```

Let's see what XX2 looks like

```
> XX2;
```

$$\left[-1, \frac{-9}{10}, \frac{-4}{5}, \frac{-7}{10}, \frac{-3}{5}, \frac{-1}{2}, \frac{-2}{5}, \frac{-3}{10}, \frac{-1}{5}, \frac{-1}{10}, 0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1 \right]$$

Now here's the standard example of a function whose interpolation polynomials with equispaced nodes

yield very bad approximation

```
> f:=x->1/(1+20*x^2);
```

$$f := x \rightarrow \frac{1}{1 + 20x^2}$$

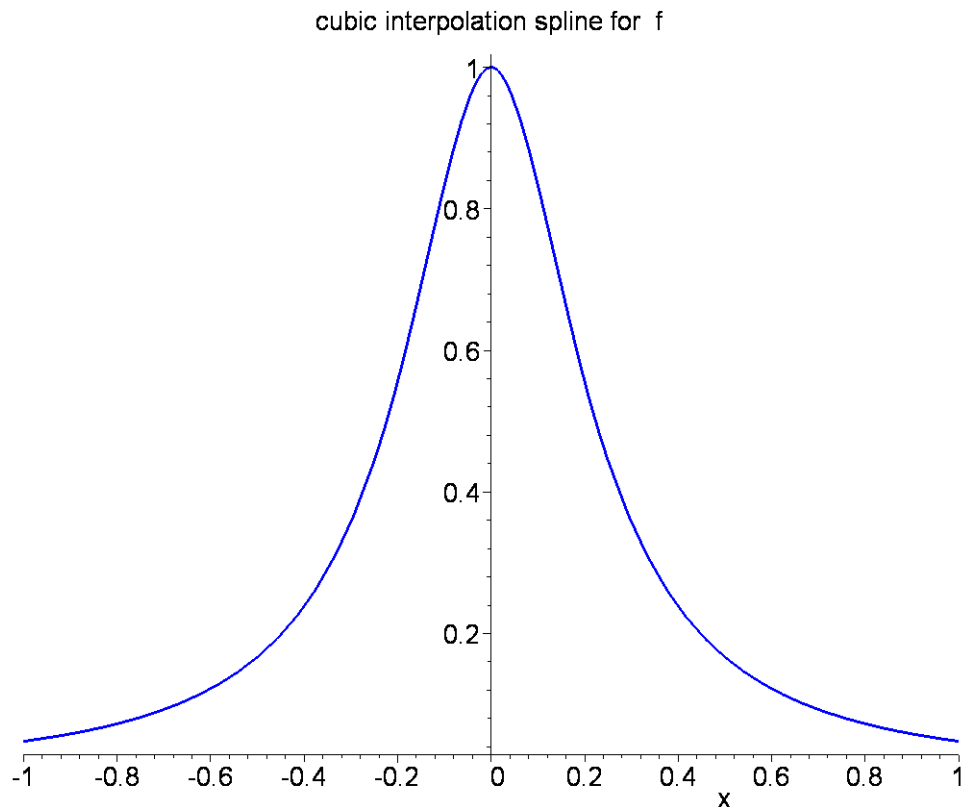
```
> YY3:=map(f,XX2);
```

$$YY3 := \left[\frac{1}{21}, \frac{5}{86}, \frac{5}{69}, \frac{5}{54}, \frac{5}{41}, \frac{1}{6}, \frac{5}{21}, \frac{5}{14}, \frac{5}{9}, \frac{5}{6}, 1, \frac{5}{6}, \frac{5}{9}, \frac{5}{14}, \frac{5}{21}, \frac{1}{6}, \frac{5}{41}, \frac{5}{54}, \frac{5}{69}, \frac{5}{86}, \frac{1}{21} \right]$$

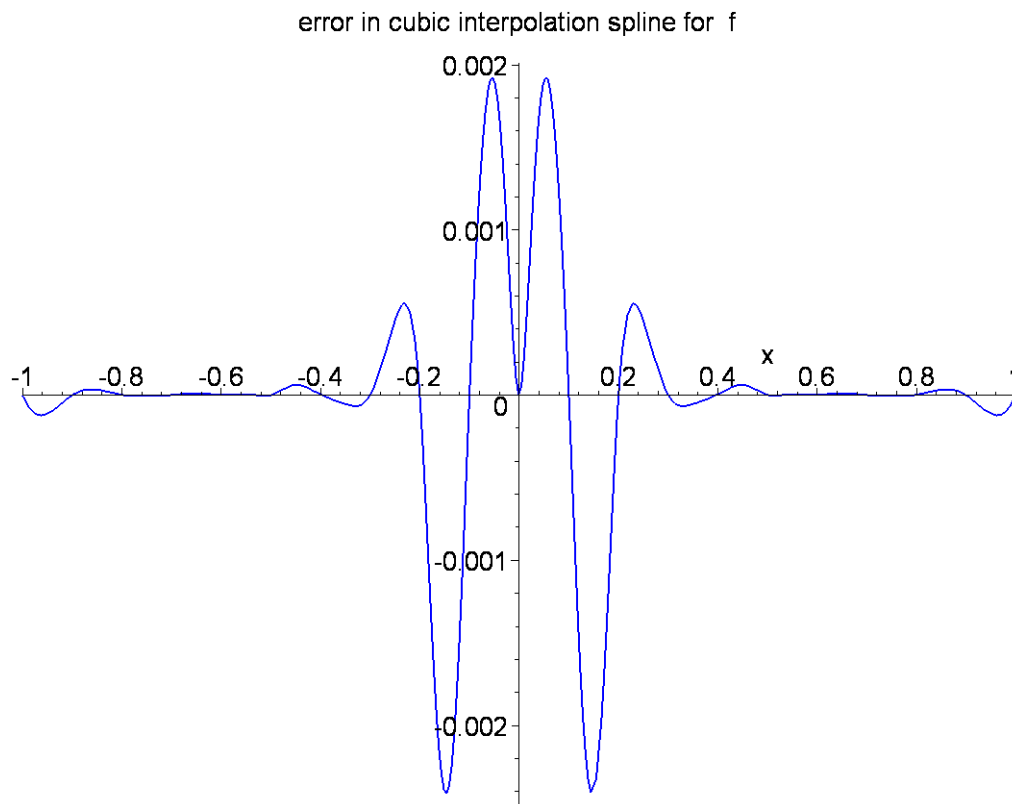
Let's look at the natural cubic spline interpolation first.

```
> ex3:=spline(XX2,YY3,x,cubic):
```

```
> plot(ex3,x=-1..1,thickness=3,color=blue,title="cubic interpolation spline for f");
```



```
> plot(f(x)-ex3,x=-1..1,thickness=2,color=blue,title="error in cubic interpolation spline for f");
```

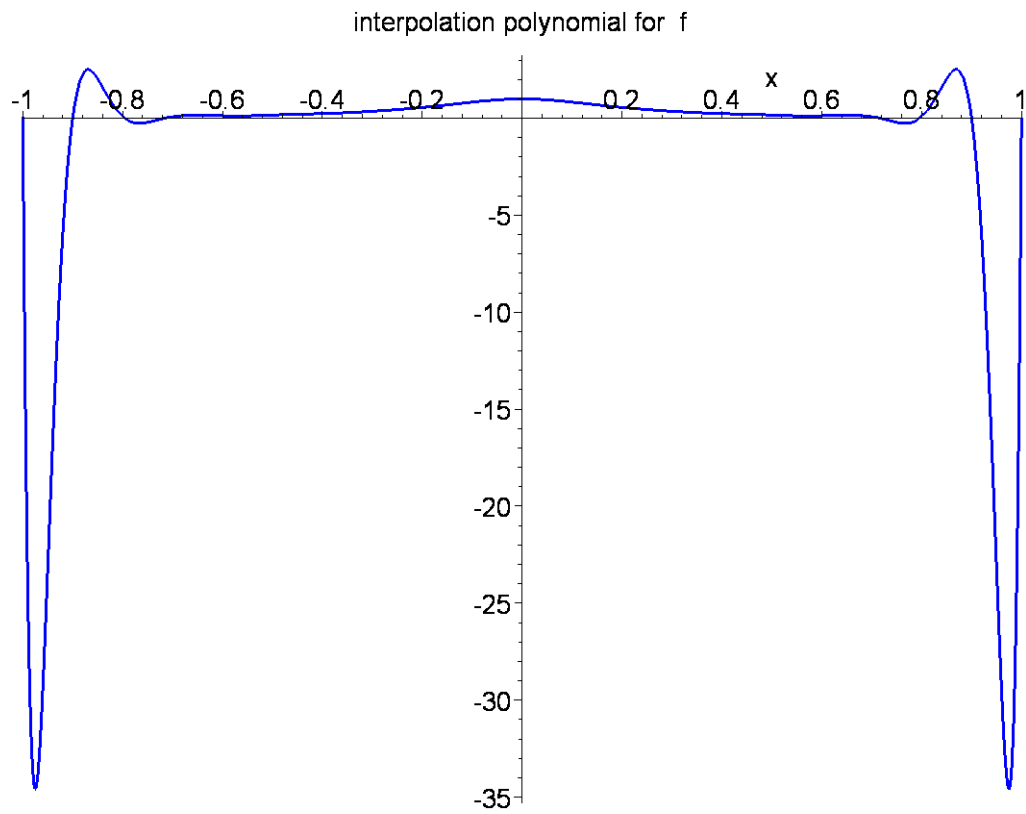


Now lets look at the interpolation polynomial

```
> poly:=interp(XX2,YY3,x);
```

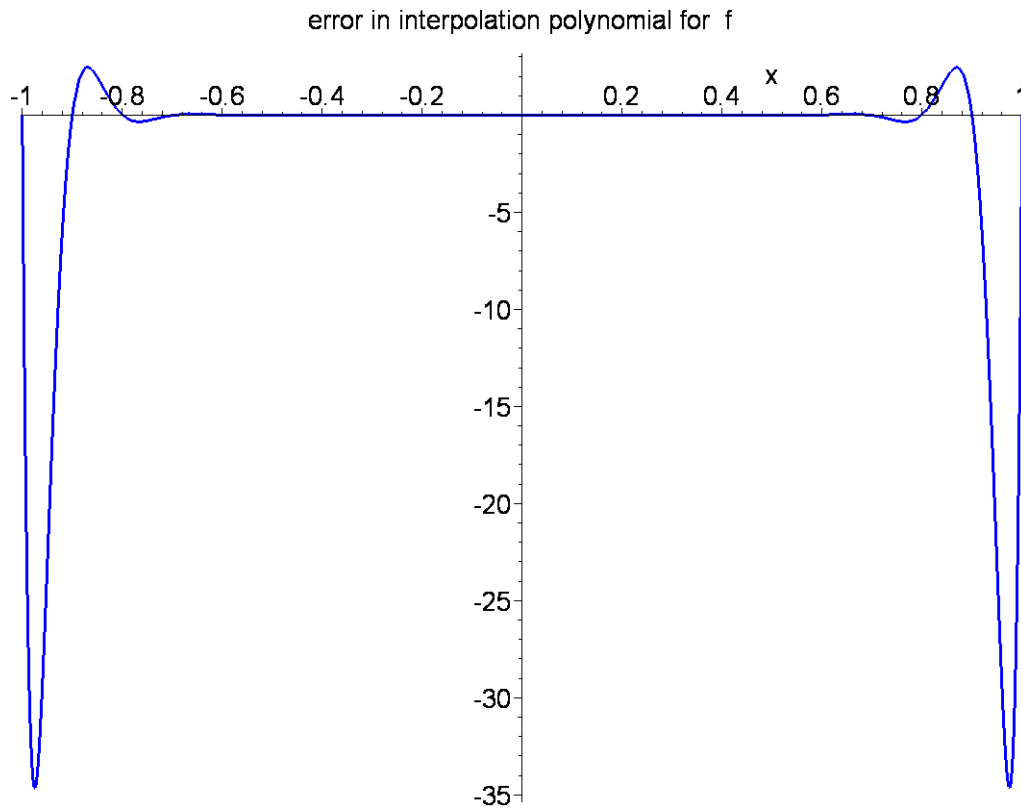
$$\begin{aligned}
 poly := & -\frac{50067740}{2554587}x^2 + \frac{583084034000000}{19099340001}x^8 + \frac{3344511921200}{10139155803}x^4 - \frac{360642262600000}{91252402227}x^6 \\
 & + \frac{29350000000000000}{30417467409}x^{16} - \frac{701368750000000000}{821271620043}x^{14} + \frac{123722971250000000}{273757206681}x^{12} \\
 & - \frac{13536954250000000}{91252402227}x^{10} + 1 + \frac{125000000000000000}{821271620043}x^{20} - \frac{162500000000000000}{273757206681}x^{18}
 \end{aligned}$$

```
> plot(poly,x=-1..1,thickness=3,color=blue,title="interpolation polynomial for f");
```



There really isn't much need to plot the error here - it is obviously bad. Nonetheless, here it is

```
> plot(poly-f(x),x=-1..1,thickness=3,color=blue,title="error in  
interpolation polynomial for f");
```



We expect the interpolation polynomial with Chebyshev nodes to be much better. Let's check it.

```
> n:=20: XX3:=[]: for k from 0 to n do
  XX3:=[op(XX3), -cos(Pi*(2*k+1)/(2*n+2))]: od:
> XX3;
```

$$\left[-\cos\left(\frac{1}{42}\pi\right), -\cos\left(\frac{1}{14}\pi\right), -\cos\left(\frac{5}{42}\pi\right), -\frac{1}{2}\sqrt{3}, -\cos\left(\frac{3}{14}\pi\right), -\cos\left(\frac{11}{42}\pi\right), -\cos\left(\frac{13}{42}\pi\right), \right.$$

$$\left. -\cos\left(\frac{5}{14}\pi\right), -\cos\left(\frac{17}{42}\pi\right), -\cos\left(\frac{19}{42}\pi\right), 0, \cos\left(\frac{19}{42}\pi\right), \cos\left(\frac{17}{42}\pi\right), \cos\left(\frac{5}{14}\pi\right), \cos\left(\frac{13}{42}\pi\right), \right.$$

$$\left. \cos\left(\frac{11}{42}\pi\right), \cos\left(\frac{3}{14}\pi\right), \frac{1}{2}\sqrt{3}, \cos\left(\frac{5}{42}\pi\right), \cos\left(\frac{1}{14}\pi\right), \cos\left(\frac{1}{42}\pi\right) \right]$$

```
> YY4:=map(f, XX3):
```

Now if we ask Maple to compute the interpolation polynomial for the nodes determined by XX3 and YY4 we will have a very long wait. Maple will try to do the calculation exactly. The resulting expression is huge. To keep the time required reasonable we will ask Maple to use floating point.

```
> XX5:=evalf(XX3, 16);
```

```
XX5 := [-.9972037971811801, -.9749279121818236, -.9308737486442043,
  -.8660254037844385, -.7818314824680298, -.6801727377709194, -.5633200580636222,
  -.4338837391175581, -.2947551744109038, -.1490422661761745, 0., .1490422661761745,
```

```
.2947551744109038, .4338837391175581, .5633200580636222, .6801727377709194,  
.7818314824680298, .8660254037844385, .9308737486442043, .9749279121818236,  
.9972037971811801]
```

```
> YY5:=evalf(YY4,16);
```

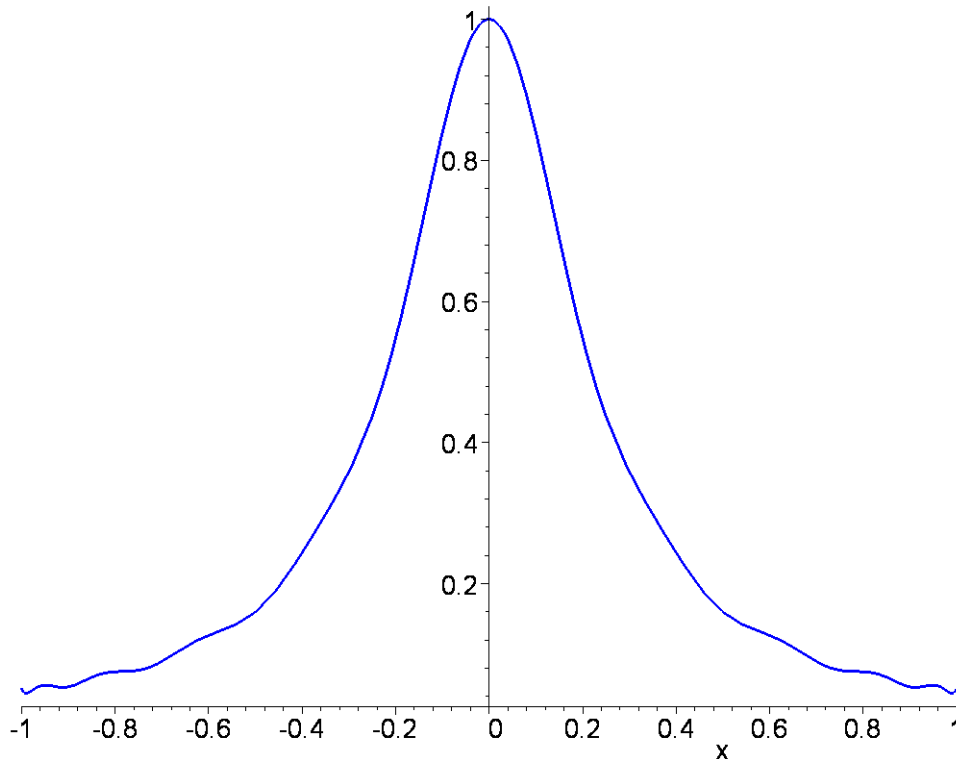
```
YY5 := [.04787367111999059, .04997579003057067, .05455382989253653,  
.06250000000000000, .07561316984287769, .09753529229176629, .1361175774293851,  
.2098590972240933, .3652818245174990, .6923903808027031, 1., .6923903808027031,  
.3652818245174990, .2098590972240933, .1361175774293851, .09753529229176629,  
.07561316984287769, .06250000000000000, .05455382989253653, .04997579003057067,  
.04787367111999059]
```

```
> Digits:=16: polyc:=sort(interp(XX5,YY5,x)); Digits:=10:
```

```
polyc := 4450.839713813448 x20 - .45251 10-7 x19 - 23589.45048326453 x18 + .18196 10-6 x17  
+ 53755.01664380374 x16 - .30026 10-6 x15 - 68893.99157553368 x14 + .27478 10-6 x13  
+ 54559.81191744943 x12 - .13950 10-6 x11 - 27646.60786099121 x10 + .44255 10-7 x9  
+ 8996.352335173073 x8 - .7855 10-8 x7 - 1848.311442863226 x6 + .7644 10-9 x5  
+ 233.6096603556530 x4 - .3344 10-10 x3 - 18.21724636095334 x2 + .4499 10-12 x  
+ .99999999999999990
```

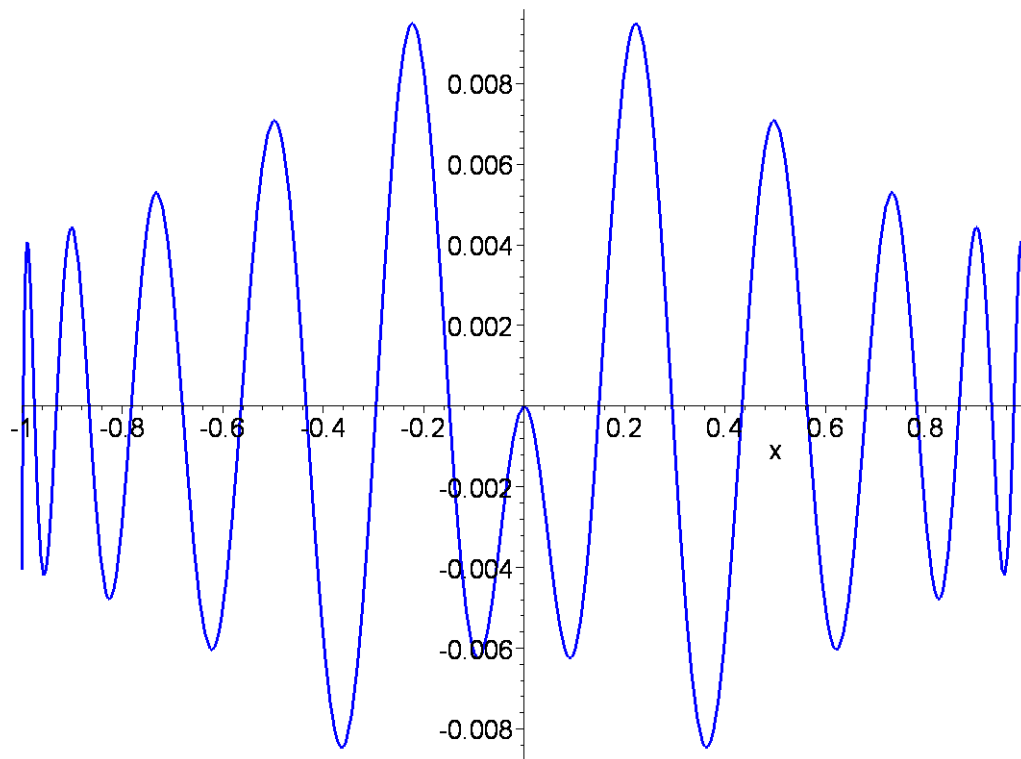
```
> plot(polyc,x=-1..1,thickness=3,color=blue,title="Chebyshev  
interpolation polynomial for f");
```

Chebyshev interpolation polynomial for f



```
> plot(f(x)-polyc,x=-1..1,thickness=3,color=blue,title="error in
```

Chebyshev interpolation polynomial for f");
error in Chebyshev interpolation polynomial for f



Even for this example the cubic spline does a little better than the Chebyshev interpolation polynomial. Of course, that will not always be the case.

Example. A sample natural cubic spline

In the examples above I suppressed the formulae for the cubic splines because they are very large. To see what a cubic spline "looks like" let's look at a simple example.

```
>  
> XX6 := [0, 1, 2, 3, 4, 5]: YY6 := [-1, -1, 1, 0, 2, 2]:  
> ex6 := spline(XX6, YY6, x, cubic);
```

$$ex6 := \begin{cases} -1 - \frac{9}{11}x + \frac{9}{11}x^3 & x < 1 \\ \frac{21}{11} - \frac{105}{11}x + \frac{96}{11}x^2 - \frac{23}{11}x^3 & x < 2 \\ -\frac{387}{11} + \frac{507}{11}x - \frac{210}{11}x^2 + \frac{28}{11}x^3 & x < 3 \\ 90 - \frac{870}{11}x + \frac{249}{11}x^2 - \frac{23}{11}x^3 & x < 4 \\ -\frac{1058}{11} + \frac{666}{11}x - \frac{135}{11}x^2 + \frac{9}{11}x^3 & \text{otherwise} \end{cases}$$

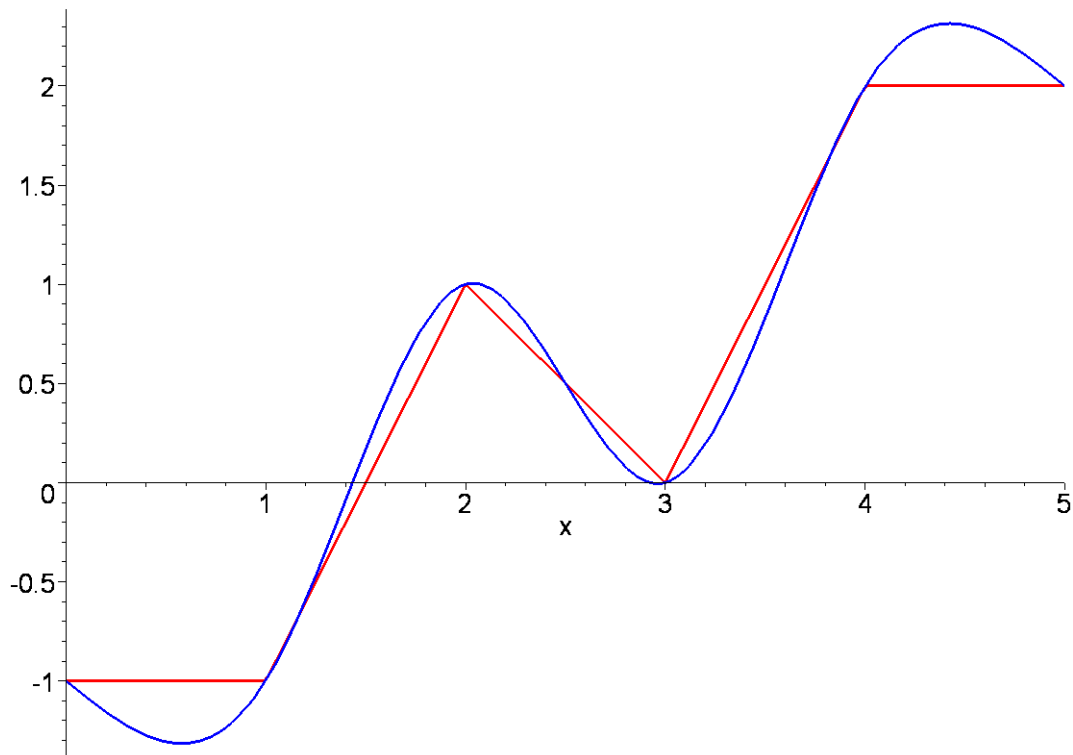
To see how it behaves we will plot it together with a piecewise linear function joining the data points.

```
> ex6lin:=spline(XX6,YY6,x,linear);
```

$$ex6lin := \begin{cases} -1 & x < 1 \\ -3 + 2x & x < 2 \\ 3 - x & x < 3 \\ -6 + 2x & x < 4 \\ 2 & \text{otherwise} \end{cases}$$

```
> plot([ex6,ex6lin],x=0..5,thickness=3,color=[blue,red],
title="Natural cubic spline compared with piecewise linear
interpolation");
```

Natural cubic spline compared with piecewise linear interpolation



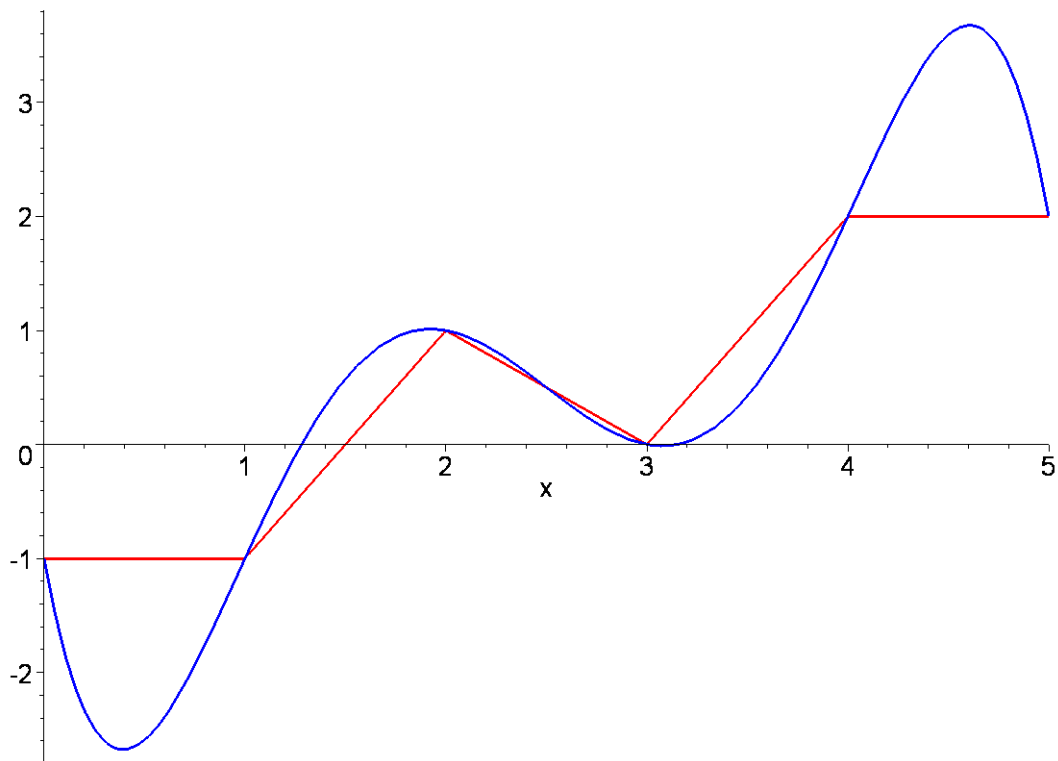
As you can see the spline does not oscillate much between the data points. Let's try the same comparison for the interpolation polynomial.

```
> ex7:=interp(XX6,YY6,x);
```

$$ex7 := -\frac{11}{60}x^5 + \frac{55}{24}x^4 - 10x^3 + \frac{425}{24}x^2 - \frac{589}{60}x - 1$$

```
> plot([ex7,ex6lin],x=0..5,thickness=3,color=[blue,red],title="Interpolation polynomial compared with piecewise linear interpolation");
```

Interpolation polynomial compared with piecewise linear interpolation



As you can see the oscillation is worse than for the cubic spline.

```
>
```