

## Polynomial Interpolation

Mth 351 August 5 2002 Maple 6

Bent E. Petersen

Filename: 351u2002\_interpolation\_polys.mws

In this worksheet we will construct interpolation polynomials for a function  $f$  in an interval for the cases of equispaced, Chebyshev and "random" nodes. I do a specific example for a simple function  $f$ . I have appended instructions on how to do the same examples in **Matlab** for those of you who prefer **Matlab**.

Everything you need for assignment 3 is here.

### Assignment 3. Due August 12 2002

Construct interpolation polynomials with 13 ( $n=12$ ) and with 21 ( $n=20$ ) nodes for the function  $f$  given by  $f(x) = 1/(1+50*x^2)$  on the interval  $[-1,1]$ . Try equispaced nodes and Chebyshev nodes

$$-\cos((2*k+1)*\text{Pi}/(2*n+2)) \quad , k = 0..n$$

Try other nodes if you like. Look at the error (perhaps plot it) and comment on what you see.

We first develop three simple routines to construct the interpolation nodes. We compute the nodes as floating point numbers to avoid large symbolic expressions.

### Enodes

Enodes( $n,a,b,d$ ) returns  $n+1$  equispaced nodes for the interval  $[a,b]$ . The nodes are evaluated to  $d$  decimal digits. If  $d$  is omitted then  $d$  is set equal to the current value of Digits.

```
> Enodes:=proc(n,a,b)
>   local L, k, d;
>   if nargs < 4 then
>     d:=Digits;
>   else
>     d:=args[4]; # optional parameter
>   fi;
>   L:=[seq(evalf(a+(b-a)*k/n,d), k=0..n)];
```

```
> end:
```

Here is a test of Enodes:

```
> Enodes(10,3,6,4);  
[3., 3.300, 3.600, 3.900, 4.200, 4.500, 4.800, 5.100, 5.400, 5.700, 6.]
```

## Cnodes

Cnodes(n,a,b,d) returns the n+1 Chebyshev nodes for the interval [a,b] evaluated to d digits precision. If d is omitted then d is set equal to the current value of Digits.

```
> Cnodes:=proc(n,a,b)  
>   local L, k, d;  
>   if nargs < 4 then  
>     d:=Digits;  
>   else  
>     d:=args[4]; # optional parameter  
>   fi;  
>   L:=[seq(evalf(a+(b-a)/2*(1-cos((2*k+1)*Pi/(2*n+2))),d),k=0..n)];  
> end:
```

Here is a test of Cnodes:

```
> Cnodes(10,3,6,4);  
[3.015, 3.136, 3.367, 3.689, 4.077, 4.500, 4.923, 5.311, 5.633, 5.864, 5.985]
```

## Rnodes

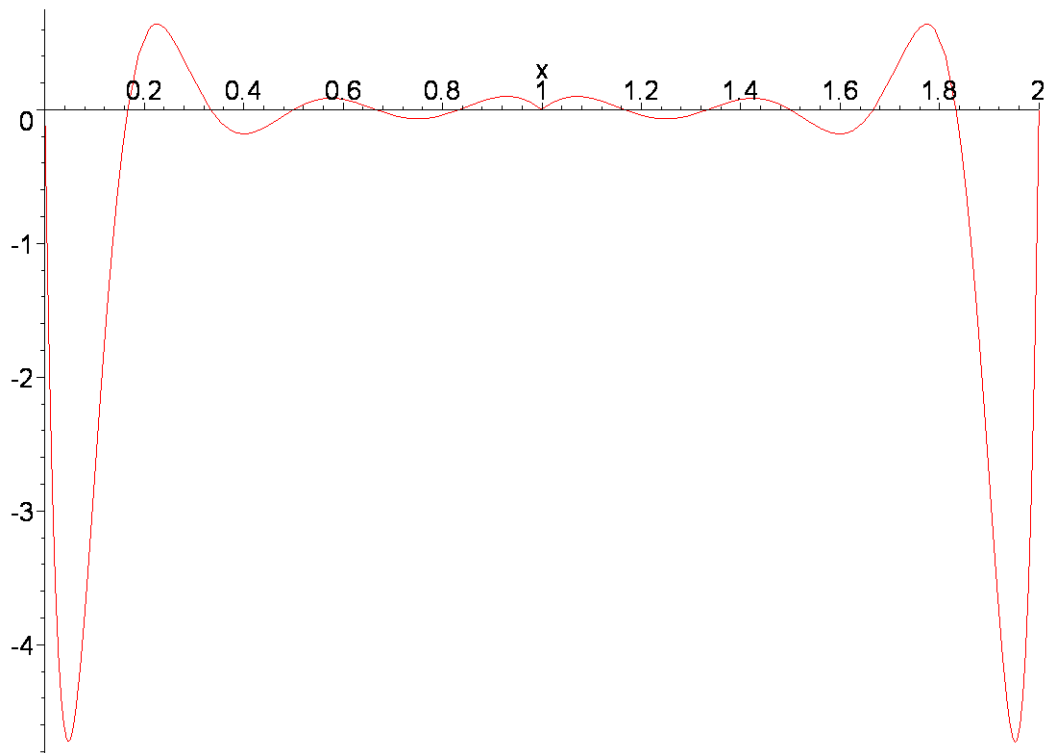
Rnodes(n,a,b,d) returns n+1 random nodes for the interval [a,b]. The nodes are evaluated to d decimal digits. If d is omitted then d is set equal to the current value of Digits.

```
> Rnodes:=proc(n,a,b)  
>   local L, k, d;  
>   if nargs < 4 then  
>     d:=Digits;  
>   else  
>     d:=args[4]; # optional parameter  
>   fi;  
>   L:={};  
>   # Avoid coincidences  
>   while nops(L) < n+1 do
```



$$+ 31722.40975 x^3 - 4372.023848 x^2 + 245.6828160 x + 3.$$

```
> plot(f(x)-pE,x=0..2,title="Error in equispaced interpolation");  
Error in equispaced interpolation
```



## Chebyshev interpolation

```
> YC:=map(f,XC);
```

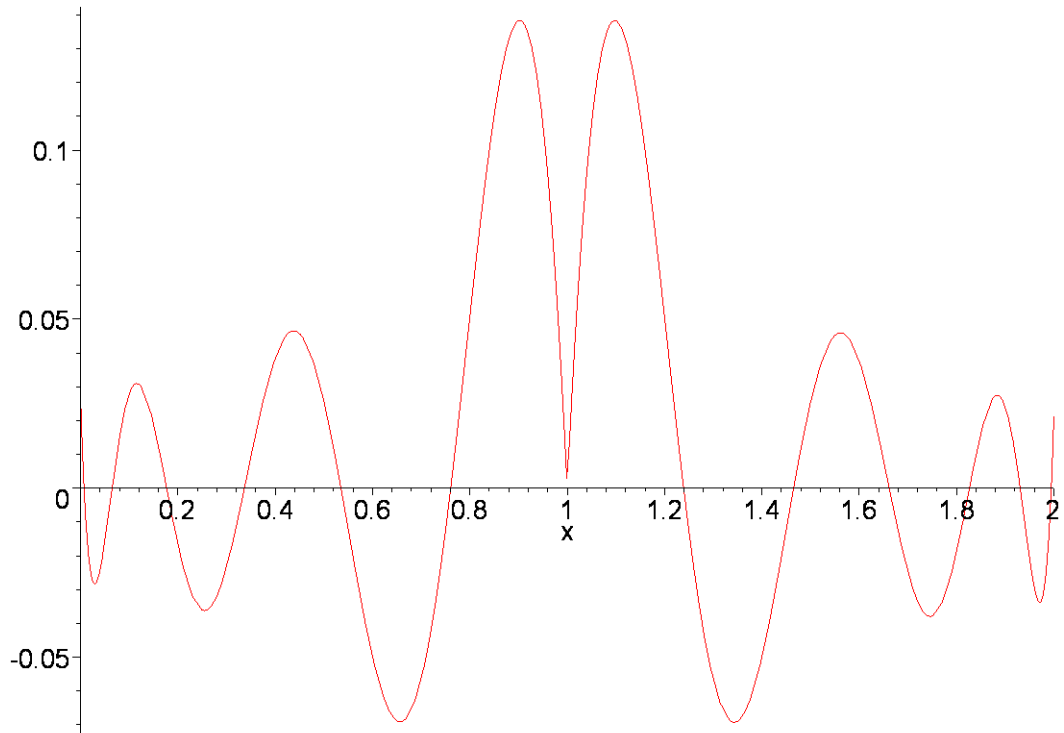
```
YC := [2.985418, 2.870032, 2.645968, 2.326246, 1.929454, 1.478642, 1., 1.95728, 2.85892,  
3.65248, 4.29192, 4.74008, 4.97084]
```

```
> pC:=sort(interp(XC,YC,x),x);
```

```
pC := -116.4835736 x12 + 1397.802894 x11 - 7305.576327 x10 + 21802.99042 x9  
- 40940.10755 x8 + 50262.40019 x7 - 40646.79187 x6 + 21314.86640 x5 - 6965.511895 x4  
+ 1318.110657 x3 - 126.3137714 x2 + 2.642053659 x + 2.972377837
```

```
> plot(f(x)-pC,x=0..2,title="Error in Chebyshev interpolation");
```

Error in Chebyshev interpolation



## Random interpolation

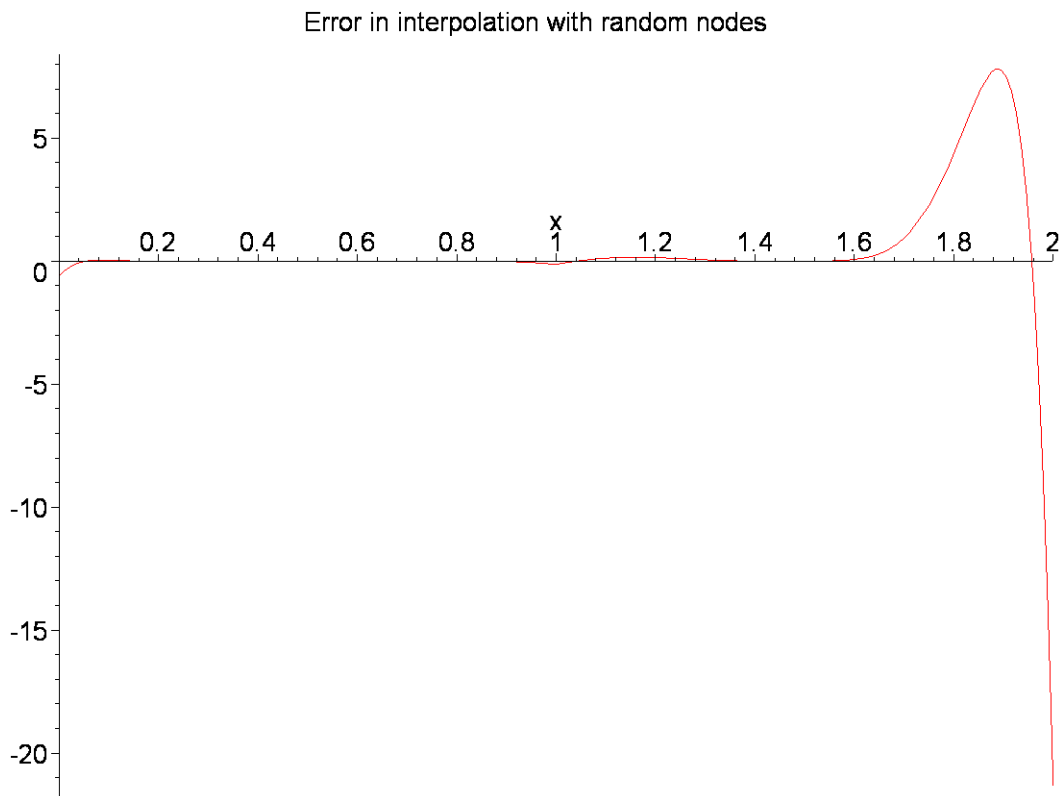
```
> YR:=map(f,XR);
```

```
YR := [2.8944370, 2.689062, 2.119686, 2.083080, 2.024854, 1.909286, 1.423032, 1.321590,  
2.62756, 2.92932, 2.99660, 3.06616, 4.82856]
```

```
> pR:=sort(interp(XR,YR,x),x);
```

```
pR := 248.4727240 x12 - 2627.876805 x11 + 12179.68150 x10 - 32576.91907 x9 + 55758.74928 x8  
- 64044.72934 x7 + 50363.06497 x6 - 27146.78529 x5 + 9863.880292 x4 - 2322.068974 x3  
+ 327.8611574 x2 - 25.81333769 x + 3.618510537
```

```
> plot(f(x)-pR,x=0..2,title="Error in interpolation with random  
nodes");
```



Note the errors are quite large in some cases above.

## Matlab

Here's one way to do the first two examples above in Matlab:

### Abcissae for plotting

```
>> Xeval = linspace(0,2,200);
```

### Common data

```
>> a = 0, b = 2, n = 12
```

### Equispaced interpolation

```
>> XE = linspace(a,b,n+1)
```

```
>> YE = XE + 3*abs(XE-1)
```

```
>> pE = polyfit(XE,YE,n)
```

```
>> ErrE = Xeval + 3*abs(Xeval-1) - polyval(pE,Xeval);
```

```
>> plot(Xeval,ErrE)
```

### **Chebyshev interpolation**

```
>> Xtmp = linspace(1/(2*n+2),(2*n+1)/(2*n+2),n+1)*pi
```

```
>> XC = 1 - cos(Xtmp)
```

```
>> YC = XC + 3*abs(XC-1)
```

```
>> pC = polyfit(XC,YC,n)
```

```
>> ErrC = Xeval + 3*abs(Xeval-1) - polyval(pC,Xeval);
```

```
>> plot(Xeval,ErrC)
```

I'll leave the random interpolation example to you.

[ >