

Homework 3 - Solutions  
MA/CS 375, Fall 2005

1. Provide an upper bound for the Lagrange interpolation error for

$$f(x) = \sin(2x), \quad \frac{\pi}{4} \leq x \leq \frac{3\pi}{4}$$

with 7 equally spaced nodes

$$x_k = \frac{\pi}{4} + \frac{k\pi}{12}, \quad k = 0, \dots, 6$$

Plot the interpolant (use `polyfit`) and compute the maximum error by evaluating  $f$  and the interpolant on 100 equally spaced points. How do the results compare with your upper bound?

*Solution:* The text provides an upper bound on the error for Lagrange interpolation which is given as equation (3.7).

$$\max_{x \in I} |E_n f(x)| \leq \frac{\max_{x \in I} |f^{(n+1)}(x)|}{4(n+1)} h^{n+1}$$

We have that  $n = 6$ ,  $I = [-1, 1]$ ,  $h = x_{k+1} - x_k = \frac{\pi}{12}$  and

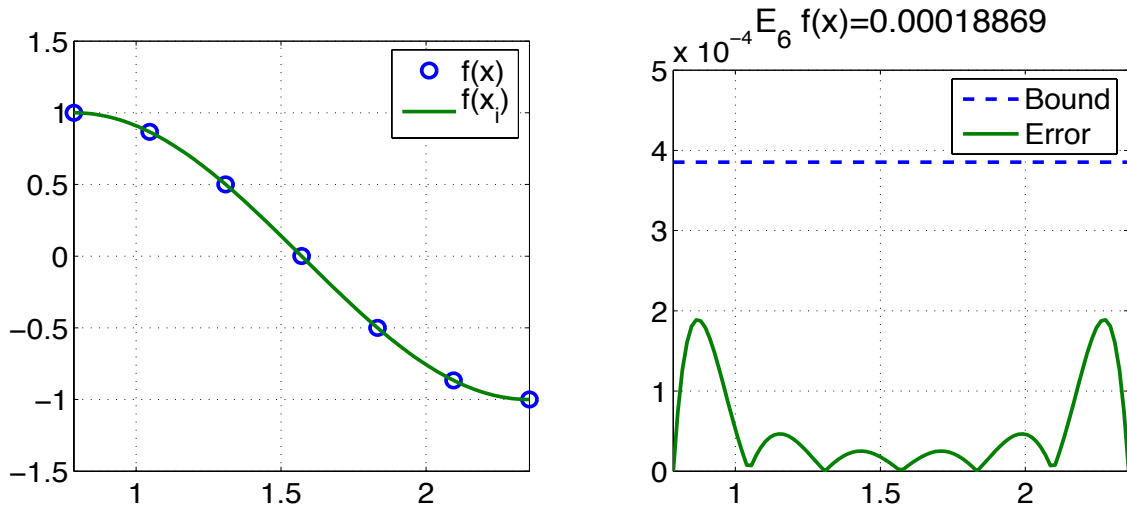
$$f^{(n+1)}(x) = \frac{d^{n+1}}{dx^{n+1}} f(x) = -128 \cos(2\pi x), \quad \max_{x \in I} |f^{(n+1)}(x)| = 128$$

$$\max_{x \in I} |E_6 f(x)| \leq \frac{128}{4(7)} \left(\frac{\pi}{12}\right)^7 = \frac{\pi^7}{7837208} \approx 3.8533 \times 10^{-4}$$

The code follows:

```
x=linspace(pi/4,3*pi/4,7)';
xi=linspace(pi/4,3*pi/4,100)';
f=sin(2*x);
exact=sin(2*xi);
bound=ones(100,1)*(128/(4*7))*(pi/12)^7;
p=polyfit(x,f,6);
fi=polyval(p,xi);
err=abs(fi-exact);
subplot(1,2,1),plot(x,f,'o',xi,exact,'-', 'LineWidth',2, 'MarkerSize',8);
grid on;
axis([pi/4 3*pi/4 -1.5 1.5]);
legend('f(x)', 'f(x_i)');
subplot(1,2,2),plot(xi,bound,'--',xi,err, 'LineWidth',2);
grid on;
```

```
axis([pi/4 3*pi/4 0 5e-4]);
legend('Bound','Error');
title(['E_6 f(x)=',num2str(max(err))], 'FontSize',18)
```



The theoretical bound is roughly twice as large as the actual error.

- The following is a table of the measured flame speeds at standard conditions for ethylene-air mixtures as a function of volume percentage of the ethylene (from Linnett and Hoare, *Third Symposium on Combustion, Flame and Explosion Phenomena*): Use both piecewise linear interpolation and spline interpolation to predict

Volume % ethylene	4.60	5.19	5.97	6.80	7.52	8.50	9.32	10.12
$V$ in cm/s	35.1	42.5	53.7	67.5	71.7	68.0	53.3	28.6

the flame speed for two ethylene-air mixtures which are 7.91% and 9.0% ethylene by volume, respectively. The measured speeds at these volumes are 70.9 cm/s and 61.1 cm/s respectively. Which predictions better matched the measurements?

*Solution:* Comparing at the 7.91% concentration, linear interpolation yields a value of 70.2276 and an error of 0.6724, whereas the spline method yields a value of 71.6329 and an error of 0.7329. For the 9.0% concentration point, linear interpolation yields a value of 59.0366 and error of 2.0634, whereas the spline method yields a value of 60.3893 and an error of 0.7107. For both points the spline method produces a better approximation.

3. Use `fft` to decode a time series of 7 pulses (the signal of a ohone number dialing). Each persion will be assigned a different number, but they can all be downloaded from the web page. In particular, use the signal in the file chosen from `tele_1.wav ... tele_9.wav`, where the number corresponds to the last digit of your ID number. (If it is zero, use the last nonzero digit.) You will need this information which lists the two dominant frequencies in Hz corresponding to each button on the touch-tone keyboard. Additional information (which you may find useful in decoding some of these signals), can be found in the “numbers” available on the webpage.

(in Hz)	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

*Solution:* A somewhat automated script (it requires knowledge of the pulse intervals) determined the tele-

file↓ digit→	1	2	3	4	5	6	7
1	2	6	8	9	4	8	1
2	8	6	3	5	7	0	9
3	1	8	1	4	4	6	6
4	1	2	1	3	8	7	6
5	2	4	7	3	*	0	1
6	1	2	1	2	3	6	3
7	1	4	1	5	2	3	9
8	2	6	5	3	0	8	1
9	2	6	5	3	0	8	1

phone numbers from each wav file.

```
function key=m375hw3p3(n,digit);

tol=25;
telrow=[697;770;852;941]; telcol=[1209;1336;1477];

symbol=['1','2','3';...
        '4','5','6';...
        '7','8','9';...
        '*','0','#'];
```

```

filename=['tele_',num2str(n),'.wav'];

% Lookup tables for the sampling intervals
lower=[3.718e4, 7.1465e4, 1.073e5, 1.590e5, 1.995e5, 2.642e5, 3.123e5;...
4.604e4, 8e4 , 1.228e5, 1.984e5, 2.4e5 , 2.836e5, 3.452e5;...
4.485e4, 1.234e5 , 1.814e5, 2.085e5, 2.607e5, 2.797e5, 2.967e5;...
1 , 8.35e4 , 1.065e5, 1.55e5 , 2.195e5, 2.46e5 , 3.19e5 ;...
1.35e4 , 5.25e4 , 9.8e4 , 2.02e5 , 2.45e5 , 2.69e5 , 3.085e5;...
1.4e4 , 5.2e4 , 8.227e4, 1.15e5 , 2.065e5, 2.445e5, 2.88e5 ;...
1.4e4 , 4.1e4 , 7.6e4 , 1.07e5 , 1.41e5 , 1.65e5 , 2.04e5 ;...
1.74e5 , 1.99e5 , 2.36e5 , 2.7e5 , 3.01e5 , 3.385e5, 3.7e5 ;...
1.74e5 , 1.99e5 , 2.36e5 , 2.7e5 , 3.01e5 , 3.385e5, 3.7e5];

upper=[4.398e4, 8.14e4 , 1.175e5, 1.655e5, 2.097e5, 2.743e5, 3.225e5;...
5.62e4 , 9e4 , 1.364e5, 2.086e5, 2.5e5 , 2.938e5, 3.587e5;...
4.82e4 , 1.267e5 , 1.846e5, 2.15e5 , 2.64e5 , 2.831e5, 3.004e5;...
3600 , 8.655e4 , 1.1e5 , 1.62e5 , 2.23e5 , 2.495e5, 3.255e5;...
2.65e4 , 6.3e4 , 1.5e5 , 2.145e5, 2.57e5 , 2.79e5 , 3.12e5 ;...
1.75e4 , 5.85e4 , 8.905e4, 1.25e5 , 2.165e5, 2.51e5 , 2.98e5 ;...
1.75e4 , 4.8e4 , 7.91e4 , 1.17e5 , 1.445e5, 1.715e5, 2.17e5 ;...
1.776e5, 2.09e5 , 2.455e5, 2.765e5, 3.11e5 , 3.42e5 , 3.734e5;...
1.776e5, 2.09e5 , 2.455e5, 2.765e5, 3.11e5 , 3.42e5 , 3.734e5];

[signal,Fs]=wavread(filename,[lower(n,digit) upper(n,digit)]);
N=length(signal); time=(1:N)/Fs;
spectrum=fft(signal);
N2=ceil(N/2);
spectrum=spectrum(1:N2);
freq=(1:N2)*Fs/N;
E=abs(spectrum);
tone=freq(find(E>50))';
[val,rowdex]=scomp(telrow,tone,tol);
[val,coldex]=scomp(telcol,tone,tol);
key=symbol(rowdex(1),coldex(1));

function [c,row]=scomp(a,b,tol)
a=a(:); b=b(:);
A=repmat(a,1,length(b)); B=repmat(b.',length(a),1);
[row,col]=find(abs(A-B)<tol); c=a(row);

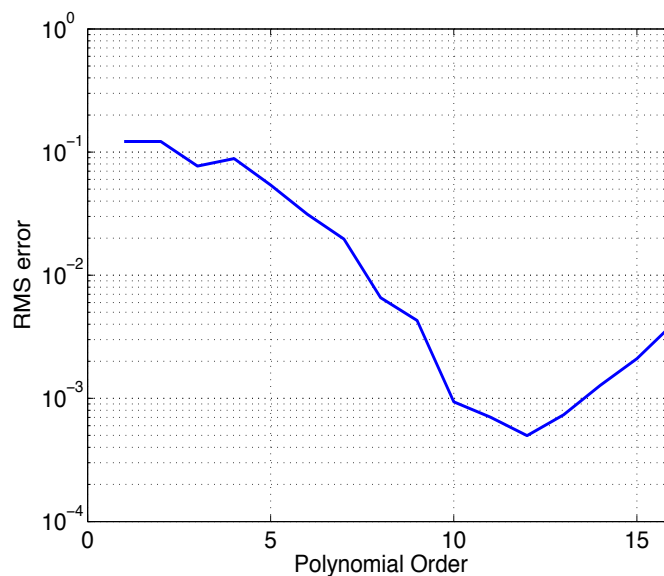
```

4. Evaluate the function  $f(x) = \cos(2\pi x)$  at 17 equispaced nodes in  $[-1, 1]$ . To the data at each node add random perturbations uniformly distributed in  $[-10^{-3}, 10^{-3}]$ . Use **polyfit** with the perturbed data to compute the least squares polynomial approximations of degrees 1-16. For each of the 16 approximations, compute the normalized RMS difference between the approximating polynomial and  $f$  at 100 equally spaced points in  $[-1, 1]$ . (The normalized RMS difference between the length 100 vectors  $y_1$  and  $y_2$  is computed by  $\text{norm}(y_1 - y_2) / 100$ ). Plot the differences versus the polynomial degree. Which degree yields the best results? Suggest an explanation.

*Solution:*

```
function rmserr=m375hw3p4(N)
x=linspace(-1,1,N+1)';
f=cos(2*pi*x);
pert=1e-3*(2*rand(N+1,1)-1);
fp=f+pert;
coeff=polyfit(x,fp,N);
xi=linspace(-1,1,100)';
fi=polyval(coeff,xi);
exact=cos(2*pi*xi);
rmserr=norm(fi-exact)/100;
```

Running this script 100 times and averaging the RMS error over all trials results in an average behavior.



The RMS error decreases for increasing polynomial order until up until a point of around 11th order, after which the error begins to increase again. The initial improvement arises from having an increasing set of polynomials to approximate the exact function  $f(x)$ , however, the algorithm `polyfit` becomes ill-conditioned resulting in increasing errors for additional terms. In practice, this problem can be improved by choosing a different basis set than the monomials such a basis of orthogonal polynomials and choosing a non-uniform grid. The interpolation error is always largest near the boundaries for a uniform grid. The maximum error can be reduced by redistributing the grid points so that they are closer together near the boundaries and further apart in the middle of the interval.

5. Show that the regression line passes through the point whose abscissa is the average of  $\{x_i\}$  and whose ordinate is the average of  $\{f(x_i)\}$ . (Give a more detailed answer than in the text's solutions.)

*Solution:* equation (3.20) in the text, the coefficients of the regression line satisfy the equations

$$a_0(n+1) + a_1 \sum_{i=0}^n x_i = \sum_{i=0}^n f(x_i) \quad (1)$$

$$a_0 \sum_{i=0}^n x_i + a_1 \sum_{i=0}^n \sum_{i=0}^n x_i^2 = \sum_{i=0}^n x_i f(x_i) \quad (2)$$

The average abscissa and ordinate are

$$\bar{x} = \frac{1}{n+1} \sum_{i=0}^n x_i, \quad \bar{f} = \frac{1}{n+1} \sum_{i=0}^n f(x_i)$$

Dividing equation (1) by  $n+1$  gives

$$a_0 + a_1 \frac{1}{1+n} \sum_{i=0}^n x_i = \frac{1}{n+1} \sum_{i=0}^n f(x_i), \quad a_0 + a_1 \bar{x} = \bar{f}$$

Therefore the average ordinate and abscissa lie on the regression line.