Homework 5: Spectral Methods

Due Thursday, December 12th at 1 am.

Problem 1:

Solve the boundary value problem

$$u_{xx} + 4u_x + e^x u = \sin(8x)$$

numerically on $[-1, 1]$ with boundary conditions $u(\pm 1) = 0$. Save $u(0)$ as `A1.dat`.

Problem 2:

Below is program 14 from the Spectral Methods codes we looked at in class, that solves $u_{xx} = e^u$:

```
N = 16;
 [D,x] = cheb(N); D2 = D^2; D2 = D2(2:N,2:N);
 u = zeros(N-1,1);
 change = 1; it = 0;
 while change > 1e-15                   % fixed-point iteration
   unew = D2\exp(u);
   change = norm(unew-u,inf);
   u = unew; it = it+1;
 end
 u = [0;u;0];
 clf, subplot('position',[.1 .4 .8 .5])
 plot(x,u,'.','markersize',16)
 xx = -1:.01:1;
 uu = polyval(polyfit(x,u,N),xx);
 line(xx,uu), grid on
 title(sprintf('no. steps = %d      u(0) =%18.14f',it,u(N/2+1)))
```

Modify this code to use Newton's method. Start from exactly the same initial conditions. Save the `change` variable at the first and second iteration into `A2.dat` and `A3.dat`. Look for super fast convergence in your `change` if you get Newton's method right.

Problem 3:

Solve the nonlinear initial boundary value problem

$$u_t = u_{xx} + e^u, \quad x \in [-1, 1], \quad t > 0, \quad u(\pm 1, t) = u(x, 0) = 0.$$

Save $u(0, 3.5)$ as `A4.dat`. Next, compute t_5 so that $u(0, t_5) = 5$ and save this time as `A5.dat`.

This problem is challenging as you get to $t = 3.5$ and beyond, so as a quick fix (unless you want to do more involved things yourself) we recommend using `ode23s` in Matlab with a very small dt when integrating in order to answer this question.

Bonus problem 1.

Write a routine for barycentric interpolation:

$$p(x) = \frac{\sum_{j=0}^{N} \frac{a_j^{-1} u_j}{x - x_j}}{\sum_{j=0}^{N} \frac{a_j^{-1}}{x - x_j}}$$

with

$$a_j := \prod_{k=0, k \neq j}^{N} (x_j - x_k)$$

Note that

$$p_j(x) = \frac{1}{a_j} \prod_{k=0, k \neq j}^{N} (x - x_k)$$

with a_j defined as above is the unique polynomial interpolant of degree N to the values 1 at x_j and 0 at x_k , for $k \neq j$. Therefore, given data on a grid, you can easily compute the coefficients (barycentric `polyfit`) and use them to interpolate to a finer grid (barycentric `polyval`).

Write these routines to replace calls to `polyfit` and `polyval` in Program 9:

```
N = 16;
 xx = -1.01:.005:1.01; clf
 for i = 1:2
   if i==1, s = 'equispaced points'; x = -1 + 2*(0:N)/N; end
   if i==2, s = 'Chebyshev points';  x = cos(pi*(0:N)/N); end
   subplot(2,2,i)
   u = 1./(1+16*x.^2);
   uu = 1./(1+16*xx.^2);
   p = polyfit(x,u,N);              % interpolation
   pp = polyval(p,xx);              % evaluation of interpolant
   plot(x,u,'.','markersize',13)
   line(xx,pp)
   axis([-1.1 1.1 -1 1.5]), title(s)
   error = norm(uu-pp,inf);
   text(-.5,-.5,['max error = ' num2str(error)])
 end
```

Save the *pp* results for equispaced points for $N = 40$ in A6.dat, and the *pp* results for Chebyshev points for $N = 60$ in A7.dat.

Bonus problem 2.

Suppose that

$$p(x) = \sum_{n=0}^{N} a_n T_n(x) = \sum_{n=0}^{N} c_n x^n,$$

let a and c be the vectors of coefficients above, and let A be the $(N+1) \times (N+1)$ matrix such that $c = Aa$.

Write a matlab function `cheb2mon` so that the command `A = cheb2mon(N)` constructs this A.

Use your matrix function to determine what polynomial (in the standard monomial expansion) corresponds to

$$T_0(x) - 2T_1(x) + 3T_2(x) + 2T_3(x) + T_4(x) - T_5(x).$$

Save the corresponding c vector as `A8.dat`.

Then figure out what Chebyshev expansion corresponds to $1 - 2x + 3x^2 + 2x^3 + x^4 - x^5$. Save the corresponding a vector as `A9.dat`.