AMATH 301 - Midterm Solutions
Summer 2008

Directions: No notes, books, or calculators are allowed. When asked to write Matlab code during this exam, write it as you would in an actual script you were going to run. You have 50 minutes to complete the exam, please give as much detail as possible in your solutions so we can attempt to give you partial credit. Please clearly indicate your answer and the process by which you game to it, i.e. *S*HOW YOUR WORK. Any questions, please ask. Good luck!

1. (*35 points*) An Oceanographer does an experiment where temperature of a tide pool is measured every 5 minutes for 2 hours ($t = [0, 5, 10, \ldots, 115, 120]$). The data from the experiment is stored in **time.dat** and **temp.dat**.

(a) Using **time.dat** and **temp.dat** write a matlab script which accomplishes the following two tasks: (This is *purely* a MATLAB question).

  (i) Calculate the average temperature of the given data and save that scalar value as A1.dat.

  (ii) Fit a cubic spline through the data which will give you the temperature at time $t = 3$ and $t = 17$. Save these temperature values as a row vector in A2.dat.

```
load time.dat
load temp.dat

%solve for the average temperature
Av = sum(temp)/length(temp);
save A1.dat Av -ASCII

%fit cubic spline
t = 0:1:120; %t values that will generate the values we're interested in
y = interp1(time,temp,t,'spline');
ans = [y(4) y(18)]
save A2.dat ans -ASCII
```

(b) Let us think of our data as a set of points, (t_k, Y_k), where t measures time and Y measures temperature. Suppose we wanted to find an interpolating polynomial for this data. What is the lowest order polynomial which is guaranteed to go through all the points given in **time.dat** and **temp.dat**?

Since we have 25 data points, we can create a polynomial of degree 24

(c) Recall that a popular method of piecewise polynomial interpolation is to create a cubic spline $S_k(t)$ which is of the form

$$S_k(t) = S_{k,0} + S_{k,1}(t - t_k) + S_{k,2}(t - t_k)^2 + S_{k,3}(t - t_k)^3$$

for $t \in [t_k, t_{k+1}]$. To solve for the four coefficients $S_{k,j}$ we impose four constraints.

(i) What are these constraints?

$$\begin{aligned} S_k(t_k) &= Y_k \\ S_k(t_{k+1}) &= Y_{k+1} \\ S'_k(t_{k+1}) &= S'_{k+1}(t_{k+1}) \\ S''_k(t_{k+1}) &= S''_{k+1}(t_{k+1}) \end{aligned}$$

(ii) Why do we impose these particular constraints?

These constraints guarantee continuity of the function, as well as the first and second derivatives and results in a smooth function.

(d) Using **time.dat** and **temp.dat** write a matlab script that will fit, in the least squares sense, the function $Y(t) = A\sin(Bt) + Ct$ where Y is temperature which is a function of t, time (ie. solve for the coefficients A, B, and C).

You may need to write two m-files, please show both of them. (Don't forget to load your data!)

To solve for the coefficients we will call

```
coeff = fminsearch('E2',[0 0 0])
```

where we define the function E2 in a separate m-file which will look like:

```
function err = E2(C)

load time.dat
load temp.dat

err = sum(abs(C(1)*sin(C(2)*time) + C(3)*time - temp).^2);
```

2. (*35 points*) Consider the following set of equations:

$$\begin{aligned} x_1 + 2x_3 &= 2 \\ 2x_1 + 3x_2 + x_3 &= 7 \\ 5x_1 - 3x_2 + 2x_3 &= 29 \end{aligned}$$

(a) Write down the linear system that represents the equations above.

$$\begin{bmatrix} 1 & 0 & 3 \\ 2 & 3 & 1 \\ 5 & -3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \\ 29 \end{bmatrix}$$

(b) Write a matlab script that would solve the system using a Gauss-Seidel iteration method using a tolerance of 10^{-6}, a maximum number of iterations of 100, and an initial guess of $[1, 0, 1]^T$.

```
max_iter = 100;
tol = 10^(-6);
XO = [1 0 1]'; %old values

%loop through Gauss Seidel where XN is the vector of new values
for j = 1:max_iter
      XN(1) = 2 - 2*XO(3);
      XN(2) = (7 - 2*XN(1) - XO(3))/3;
      XN(3) = (29 - 5*XN(1) - 3*XN(2))/2;

      if norm(XN-XO,2) < tol
            break
      end
      XO = XN;

end
```

(c) Can we guarantee that if we used a Jacobi iteration method that our method would work? Why or why not? No. The matrix is not strictly diagonally dominant. For example in the third row

$$|2| \ngtr |5| + |-3|$$

In fact, none of the rows meet the criteria!

3. (*30 points*) For the following questions, assume that we have a function $y(t)$ that we can differentiate to our heart's content.

(a) What is the Taylor series expansion of $y(t + \Delta t)$ about t?

$$y(t + \Delta t) = y(t) + \Delta t y'(t) + \frac{\Delta t^2}{2} y''(t) + \frac{\Delta t^3}{3!} y'''(t) + \dots$$

(b) What is the Taylor series expansion of $y(t - \Delta t)$ about t?

$$y(t + \Delta t) = y(t) - \Delta t y'(t) + \frac{\Delta t^2}{2} y''(t) - \frac{\Delta t^3}{3!} y'''(t) + \dots$$

(c) Consider the following difference formula:

$$\frac{y(t + \Delta t) - y(t - \Delta t)}{2\Delta t}$$

What derivative does this formula approximate? What is the order of the truncation error? (i.e. find the exponent in $\mathcal{O}(\Delta t^p)$)

Show how you came up with both the derivative and the truncation error.

$$\begin{aligned}
\frac{y(t + \Delta t) - y(t - \Delta t)}{2\Delta t} =& \frac{1}{2\Delta t}\left((y(t) + \Delta t y'(t) + \frac{\Delta t^2}{2} y''(t) + \frac{\Delta t^3}{3!} y'''(t) + \dots\right) \\
& - \frac{1}{2\Delta t}\left(y(t) - \Delta t y'(t) + \frac{\Delta t^2}{2} y''(t) - \frac{\Delta t^3}{3!} y'''(t) + \dots\right) \\
=& \frac{2\Delta t y'(t) + \frac{2\Delta t^3}{3!} y'''(t) + \mathcal{O}(\Delta t^5)}{2\Delta t} \\
=& y'(t) + \frac{\Delta t^2}{3!} y'''(t) + \mathcal{O}(\Delta t^4)
\end{aligned}$$

From this we can see that this approximates the first derivative $y'(t)$ and has second order truncation error, $\mathcal{O}(\Delta t^2)$.