AMATH 301
Midterm 1

NAME:

I Consider the following set of ordinary differential equations

$$2\frac{dy_1}{dt} = 4y_1 + 2y_2$$

$$3\frac{dy_2}{dt} = 3y_1 + 6y_2$$

Inserting the solution $\mathbf{y} = \mathbf{x}e^{(\lambda t)}$ results in an eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$.

(a) [1 point] Write the matrix $\mathbf{A}$ to construct the eigenvalue problem.

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

(b) [2 points] Obtain the eigenvalues of the eigenvalue problem.

$$\det(A-\lambda I) = \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = (\lambda-1)(\lambda-3)$$

$$\Rightarrow \lambda = 1 \text{ and } \lambda = 3$$

(c) [2 points] Obtain the normalized eigenvectors of the eigenvalue problem.

$$A - I = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \rightarrow x_1 = 1,\ x_2 = -1 \rightarrow \underline{x} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$A - 3I = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \rightarrow x_1 = 1,\ x_2 = 1 \rightarrow \underline{x} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

(d) [1 points] Write the general solution of the ordinary differential equation.

$$y(t) = C_1 \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{t} + C_2 \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} e^{3t}$$

(e) [1 point1] Write the MATLAB output for the following command:

```
>A.^2
>
```

$$\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$$

(f) [1 point] Write the MATLAB output for the following command:

```
>A^2
>
```

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

II Consider the one-dimensional differential equation

$$\frac{dy}{dt} = -54(y)$$

.

(a) [2 points] Write a MATLAB code that uses the FORWARD euler method to solve this equation with a time step of $\Delta t = 1$ over the interval [0,5], starting from the initial condition $y(0) = 1$. Will the absolute value $|y(t)|$ increase over time for this code? (You must include answer plus your work)

$$y(t+\Delta t) = y(t) + y'(t)\Delta t$$

$$y(t+\Delta t) = y(t) - 54y(t)\Delta t = y(t)(1-54\Delta t)$$

$$\{\text{or } y_{n+1} = y_n - 54y_n \cdot h\}$$

```
dt = 1
y(1) = 1
for i = 1:5
    y(i+1) = y(i)*(1-54*dt);
end
```

$|y(t+\Delta t)| = 53|y(t)| \Rightarrow$ increases over time

or $|y_{n+1}| = 53|y_n| \Rightarrow$ increases over time

(b) [2 points] Write a MATLAB code that uses the BACKWARD euler method to solve this equation with a time step of $\Delta t = 1$ over the same time interval, starting from the same initial condition. Will the absolute value $|y(t)|$ increase over time for this code? (You must include answer plus your work)

$$y_n = y_{n-1} - 54y_n\Delta t$$

$$y_n = \frac{y_{n-1}}{(1+54\cdot\Delta t)}$$

```
dt = 1
y(1) = 1
for i = 1:5,
    y(i+1) = y(i)/(1+54*dt)
end
```

$$|y_{n+1}| = \frac{|y_n|}{|1+54\Delta t|} = \frac{|y_n|}{55}$$

$\Rightarrow$ decreases over time.

III Suppose you need to fit a set of data with the following function:

$$f(x) = C10^{Ax}$$

by minimizing the following measure of error

$$E_2(A, C) = \sum_{k=1}^{n}(C10^{Ax_k} - y_k)^2$$

(a) [1 point] Write a transformation that can be used in order to linearize the given fit function.

$y(x) = \log_{10}(f(x)) = \log_{10} C + Ax$

let $B = \log_{10} C$

(b) [1 point] Write the linearized fit problem.

Fit $Y = Ax + b$ by minimizing

$E_2(A,B) = \sum_{k=1}^{n} (Ax_k + B - Y_k)^2$

(c) [3 points] Now suppose you are a researcher working on an epidemiology project that requires you to fit a data set which appears to follow an exponentiation trend of base 10, where the data given is the time t in days and the count of cells N over a period of time from 1 to 90 days.

Write a MATLAB code that imports the given data set CellGrowthRate.dat and fits the data with a base 10 exponentiation fitting function. (Tip, you may use polyfit and polyval defined at the end of this exam).

{Note that the data in the file CellGrowthRate.dat is stored as a matrix of two columns and 90 rows, where the first column corresponds to t and the second column corresponds to N}

```
load CellGrowth.dat -ASCII
x = CellGrowth(:,1);
y = CellGrowth(:,2);
Y = log10(y);
coeff = polyfit(x,Y,1)
A = coeff(1);
B = coeff(2);
C = 10^B;
```

(d) [1 point] Write the MATLAB code that would use the fitting function to predict the count of cells N for $t = 93$. That is, predict the count of cells on day 93 using the fitting function. Would this be called interpolation or extrapolation?

fit_93 = polyval(pcoeff, 93); ans = 10^{fit_93}

→ extrapolation

(e) [1 point] Write the MATLAB code which uses spline interpolation to fit the same raw data (instead of linearizing the problem) and evaluates it on the new set of data points xp = (1:.2:90)'; (see interp1 definition at the end of this exam).

xp = (1:.2:90)

spline_fit = interp1(x, y, xp, 'spline')

(f) [1 point] Write the MATLAB code which graphs on a single figure multiple graphs. Graph the data points (marked by a red star). Plot the spline fitting from part (e) with a blue line. Add a black square to the graph marking the predicted value N for day $t = 93$ from part (d). Add a legend to the figure with corresponding labels, as well as labels for the two axes.

figure(1), plot(x, y, 'r*'); hold on;

plot(x, spline_fit, 'b');

plot(93, $10^{fit\text{-}93}$, 'ks');

legend('Data set', 'spline fit', 'least square at t=93')

xlabel('t'); ylabel('f(x)');

MATLAB COMMANDS

```
polyfit: Fit polynomial to data.
    P = POLYFIT(X,Y,N) finds the coefficients of a polynomial P(X) of
    degree N that fits the data Y best in a least-squares sense. P is a
    row vector of length N+1 containing the polynomial coefficients in
    descending powers, P(1)*X^N + P(2)*X^(N-1) +...+ P(N)*X + P(N+1).

polyval: Evaluate polynomial.
    Y = POLYVAL(P,X) returns the value of a polynomial P evaluated at X. P
    is a vector of length N+1 whose elements are the coefficients of the
    polynomial in descending powers.
        Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1)

interp1:   INTERP1 1-D interpolation (table lookup).
    YI = INTERP1(X,Y,XI) interpolates to find YI, the values of
    the underlying function Y at the points in the vector XI.
    The vector X specifies the points at which the data Y is
    given.     YI = INTERP1(X,Y,XI,'method') specifies alternate methods.
    The default is linear interpolation.  Available methods are:

      'nearest'  - nearest neighbor interpolation
      'linear'   - linear interpolation
      'spline'   - piecewise cubic spline interpolation (SPLINE)
```