Introduction to Mathematical Modeling - Math 3550
Calculus, Linear Algebra, and Differential and Difference Equations for Mathematical Modeling© - Version 13

1. You may access the required software, called Mathematica, in the following three ways. 1. Get your own free copy at https://www.weber.edu/software/mathematica_request.html. Activate your copy of Mathematica through "Activate through your organization (SSO)". See Wolfram support page https://support.wolfram.com/54713 for help. If you need a laptop/tablet, contact WSU Computing Services https://www.weber.edu/ComputerLabs/laptopcheckout. 2. Any campus computer lab, including Tracy Hall Computer Lab, TY 126, and Elizabeth Hall Computer Lab, EH 213. 3. Virtual Lab: For the Virtual Lab instructions, see http://weber.edu/virtuallab.

2. Here is how to get on Mathematica:

   Log in on a computer. Move the mouse pointer to the **Mathematica icon**, which looks like the above star like symbol, and rapidly **click twice** using the **left mouse button**. You may find the Mathematica icon on the desktop, under the button Start - Programs - Wolfram or Mathematica, or by doing a search for Mathematica. Then click on NewDocument , or on New Notebook, if a notebook doesn't automatically open. If you need help, ask the lab aide.

3. Entering information is like using a word processor. You can also use the arrow and backspace keys for editing. Try typing some text and practice editing it. Mathematica displays all works in "cells". The bracket on the far right indicate the cells. You can create a new cell for your next input by clicking the mouse below the cell you have been working in. First a horizontal line will appear and then when you type a new cell will be created. This lab assumes you have already passed Math 1200, so we skip arithmetic and algebra commands.

   Throughout this lab and future labs input the statements in `True Type` font exactly as is!

4. You can draw a function, relation, or equation using the `Plot` command. Each expression must be followed by a list which gives the variable(s) and the domain (minimum value to maximum value). Enter the following.

   (a) `Plot[x^2-3x+4,{x,-10,10}]`
   Change the domain from -5 to 5 and then again from -100 to 100. You do not need a symmetric domain, try from 0 to 20.

   (b) Mathematica can plot parametric equations, in two or three dimensions.
   `ParametricPlot[{2 Cos[t],3 Sin[t]},{t,0,2Pi}]` (This is the graph of the ellipse $\frac{x^2}{4} + \frac{y^2}{9} = 1$.)
   `ParametricPlot3D[{Cos[t],Sin[t],t},{t,0,10}]` This is the graph of an ascending helix.

   (c) Mathematica can plot 3-dimensional graphs of the form $z = f(x, y)$.
   `Plot3D[Sin[x^2 y],{x,-Pi,Pi},{y,-Pi,Pi}]` This is the graph of $z = sin(x^2y)$.

   (d) You can graph a polar equation using the `PolarPlot` command.
   `PolarPlot[Sin[2t],{t,0,2Pi}]` This is the graph of $r = \sin(2\theta)$. (I used t was in place of θ since it was easier to type in!)

4. Continued.

   (e) To graph two or more functions on the same coordinate system, enter them as a list.
   `Plot[{Cos[x],x^3},{x,-2Pi,2Pi}]` (This is the graph of the two functions $y = \cos x$ and $y = x^3$ on the same coordinate system.)

   (f) You can read the coordinates of a point on a graph by first right clicking on the graph and choosing Get Coordinates (or, first, Drawing Tools) in the window that opens up. Move the cursor over the graph. The coordinates will appear on the screen. Now, try to read the coordinates of the point of intersection of the two curves in the last part. (Compare the approximate x value of the point of intersection with what you get from the `FindRoot` command, shown next.)

   Now, Try to read the coordinates of the point of intersection of the two curves in the last part. This x value (say 0.5) can be used to get a more precise answer using the `FindRoot` command. Try the following.
   `FindRoot[Cos[x]==x^3,{x, 0.5}]`

5. Mathematics can evaluate limits. However, it is best to first investigate existence of the limit of the function by graphing it or considering a table of its values, before applying the `Limit` command. Let's investigate $\lim_{x\to 0} \dfrac{\sin x}{x}$. Enter the following.

   (a) `Plot[Sin[x]/x, {x, -Pi, Pi}]` This graph shows the limit exists.

   (a) `Limit[Sin[x]/x,x->0, Direction->`"FromBelow"`]` (This is $\lim_{x\to 0^-} \frac{\sin x}{x}$.)

   (b) `Limit[Sin[x]/x,x->0, Direction->`"FromAbove"`]` (This is $\lim_{x\to 0^+} \frac{\sin x}{x}$.)

   (c) `Limit[Sin[x]/x,x->0]` (Since the left and right limits are equal, this is the correct value of $\lim_{x\to 0} \frac{\sin x}{x}$.)

6. Mathematica can do symbolic differentiation.

   (a) `D[x^2 Sin[x]-3x+1,x]` (This gives the derivative of $4 = x^2 \sin x - 3x + 1$ with respect to x.)

   (b) You can also use the prime notation.
   `f[x_]:=x^3-2x^2+5;f'[x]` Is $f'(x) = 3x^2 - 4x$?
   `f''[x]` (This gives the second derivative $f''(x)$.)
   `D[f[x],{x,3}]` (This gives the third derivative $f'''(x)$).

   (c) You can also perform implicit differentiation.
   `D[x^2 y[x]^2 + x Sin[y[x]]==1,x]` This differentiates the equation $x^2y^2 + x\sin y = 1$ with respect to x. (`y[x]` indicates y is a function of x.)
   `Solve[%,y'[x]]` Now, you have solved for y'.

   (d) You can also find partial derivatives.

   ```
   f[x_,y_,z_]:= x^4 z^3 y +x Sin[z+y];
   D[f[x,y,z],x]
   ```

   This is the first partial derivative of f with respect to x, this can also be done by `D[f[x,y,z],{x,1}]`.
   `D[f[x,y,z],{x,2},{y,1},{z,1}]` This is $\dfrac{\partial^3 f}{\partial z\, \partial y\, \partial^2 x}$.

7. Mathematica does symbolic integration. Mathematica evaluates definite integrals both numerically, and, also, by doing symbolic integration first. Input the following.

   (a) `Integrate [x^2+1,x]` (Mathematica gives the antiderivative with $c = 0$.)

   (b) `Integrate [x^2+1,{x,-1,2}]` (This evaluates $\int_{-1}^{2} (x^2+1)\,dx$.)

   (c) `f[x_,y_]:=1-(x^2/4)-(y^2 / 9);`
   `Integrate[f[x,y],{x,-2,2},{y,-3, 3}]`
   This evaluates the iterated integral $\int_{-2}^{2}\int_{-3}^{3} f\,dx\,dy$ which gives the volume under the surface $z = 1 - \frac{x^2}{4} - \frac{y^2}{9}$ and above the xy-plane.

   (d) Numerical Integration. Often we do not know the antiderivative of a function but we can approximate its definite integral numerically.
   `NIntegrate[Exp[-x^2],{x,0,Infinity}]` (This evaluates $\int_{0}^{\infty} e^{-x^2}\,dx$.)

   (e) However, you can hand evaluate some integrals more efficiently than Mathematica. The easy way to evaluate $\int 3x^2(2010-x^3)^{1999}\,dx$ is by u-substitution. Mathematica expands the integrand before finding its antiderivative!
   `Integrate[3x^2 (2010-x^3)^1999, x]`

8. The `Series` command can be used to find the power series expansion of a function about a given point up to a given desired power. The `Sum` command can find the value of some sseries.

   (a) `Series[E^x, {x,0,6}]` (This is the Maclaurin series of $f(x) = e^x$ with all terms up to the sixth power stated explicitly and `O[x]`^{7} stands for all higher order terms.)

   (b) By adding the command `Normal` a series is truncated to a polynomial.
   `Normal[Series[Cos[x], {x,0,8}]]` (This is the 8th-degree Maclaurin polynomial of $f(x) = \cos x$.)

   (c) `Series[Log[x], {x,1,5}]` (This is the Taylor series expansion of $f(x) = \ln x$ about $x = 1$ with all term up to the fifth power stated explicitly.)

   (d) Mathematica can find the sum of certain series.
   `Sum[1/n^2, {n,1,Infinity}]` (This gives the sum of the series $\sum_{n=1}^{\infty} \frac{1}{n^2}$.)

9. Braces, { }, signal a "list" to Mathematica. A list can be sued to represent a set, an interval, a table of data, the parameters in a command, a vector, a matrix, etc. For a matrix, each row is inputted as a list in the order it appears and rows are separated by commas. If a row has only one entry, it is not necessary to use braces. Now try the following.

   (a) `u={1,-3}` This is the vector $\vec{u} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$.

   (b) `MatrixForm[u]` This command shows $\vec{u}$ in matrix form.

   (c) `v={{-2,5,-1}}` This is the row vector $\vec{v} = \begin{bmatrix} -2 & 5 & -1 \end{bmatrix}$.
   `MatrixForm[v]`

   (d) `m={{1,2},{3,4},{5,6}}` This is the matrix $m = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 4 & 6 \end{bmatrix}$.

10. You can add or subtract matrices using + and − signs. The symbol "." is the multiplication sign for matrices. Input the following.

    (a) `n={{3,2,1},{6,5,4}}` This is the matrix $n = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \end{bmatrix}$.

    `l={{3,-1},{-2,6},{-4,5}}` This is the matrix $l = \begin{bmatrix} 3 & -1 \\ -2 & 6 \\ -4 & 5 \end{bmatrix}$.

    (b) `m-2*l` This results in the matrix $m - 2l$.

    (c) `n.m` This gives the product nm.
    `MatrixForm[n.m]`

11. Try the following.

    (a) `Dimensions[m]` This gives the size of the matrix m. The first number is the number of rows and the second is the number of columns.

    (b) `i=IdentityMatrix[3]` This is the 3 × 3 identity matrix i.

    (c) `d=DiagonalMatrix[{2,2,-3}]` This is the 3 × 3 diagonal matrix d with the diagonal entries 2, 2 and −3.

    (d) `Transpose[m]` This gives the transpose of the matrix m.
    `MatrixForm[Transpose[m]]`

12. Here we will perform more matrix operations. Input the following.

    (a) `a={{2.,3,-5},{-1,4,2},{5,7,2}}` The decimal point after 2 is used to get Mathematica to work in decimal (not exact) calculations.

    (b) `Det[a]` This gives the determinant of the matrix a.

    (c) `Inverse[a]` This gives the inverse of the matrix a.
    `MatrixForm[Inverse[a]]`

    (d) `MatrixPower[a,3]` This gives the matrix a^3.

    (e) `Eigenvalues[a]` This gives the eigenvalues of the matrix a. This matrix has one real-valued and a pair of complex conjugate eigenvalues.

    (f) `Eigenvectors[a]` This gives the eigenvectors of the matrix a.

    (g) `Eigensystem[a]` This gives a list of eigenvalues followed by their corresponding eigenvactors.

13. We can solve a linear system using matrices. The system $\begin{cases} 2x + 3y - 5z = 14 \\ -x + 4y + 2z = 0 \\ 5x + 7y + 2z = -9 \end{cases}$ in matrix form is $\begin{bmatrix} 2 & 3 & 5 \\ -1 & 4 & 2 \\ 5 & 7 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 14 \\ 0 \\ -9 \end{bmatrix}$. Notice that the coefficient matrix is the matrix a used above.

    (a) `Solve[{2x+3y-5z==14, -x+4y+2z==0, 5x+7y+2z==-9},{x,y,z}]` This gives the solution using the `Solve` command.

    (b) `b={14,0,-9}` This is the constant matrix on the right side.

    (c) Now, our system is $a \begin{bmatrix} x \\ y \\ z \end{bmatrix} = b$ and the solution is $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = a^{-1}\, b$. Input the following.
    `Inverse[a].b` Did you get the same answer as before?

14. The direction field of a differential equation of the form $\frac{dy}{dx} = f(x, y)$ is the plot of the vectors $\vec{v} = \langle 1, f(x, y)\rangle$ at each grid point (x, y) in the plane. The Mathematics command for such a plot is `VectorPlot`. Consider the differential equation $\frac{dy}{dx} = y - x^2$.

    (a) `VectorPlot[{1,y-x^2}, {x,-3,3}, {y,-4,4}]` (Not quite the direction field since all vectors have the same length.)

    (b) You can change the number of vectors by specifying the number of vectors.
    `VectorPlot[{1,y-x^2}, {x,-3,3}, {y,-4,4}, VectorPoints->20]`

    (c) Notice that Mathematica draws all vectors with the same length. It uses the vector colors to indicate the length of vectors. To preserve the length of vectors, use the following command.
    `VectorPlot[{1,y-x^2},{x,-3,3},{y,-4,4},VectorScaling->Automatic,VectorPoints->20]` (This is the direction field of the given ODE for $-3 \le x \le 3$ and $-4 \le y \le 4$.)

    (d) Mathematica can display the vectors as part of the approximate trajectories of the solutions using the `StreamPlot` command.
    `StreamPlot[{1,y-x^2}, {x,-3,3}, {y,-4,4}]`

    (e) The same commands work systems of ODE's. The vector field for the system $\frac{dy}{dx} = f(x, y)$, $\frac{dy}{dx} = g(x, y)$ is the graph of vectors of the form $\vec{v} = \langle f(x, y), g(x, y)\rangle$. Consider the system$\frac{dy}{dx} = -x + x^3$, $\frac{dy}{dx} = -2y$. Try each of the following.
    ```
    VectorPlot[{-x+x^3, -2y}, {x,-3,3}, {y,-4,4}]
    StreamPlot[{-x+x^3, -2y}, {x,-3,3}, {y,-4,4}]
    ```

15. Using the `DSolve` and `NDSolve` commands, we can find solutions to many differential equations.

    (a) The syntax for the general solution of $y' = f(x, y)$ is `DSolve[y'[x]==`$f(x, y)$`, y[x], x]`. Consider the equation $\frac{dy}{dx} = y - x^2$.
    `DSolve[y'[x]==y[x]-x^2, y[x], x]` (`C[1]` in the answer is the constant c_1.)

    (b) The syntax for the solution of the IVP $y' = f(x, y)$, $y(x_0) = y_0$ is `DSolve[{y'[x]==`$f(x, y)$`, y[`x_0`]==`y_0`}, y[x], x]`. Consider the IVP $\frac{dy}{dx} = y - x^2$, $y(0) = 2$.
    `DSolve[{y'[x]==y[x]-x^2,y[0]==2}, y[x], x]`

    (c) In a similar fashion, you can solve a system of equations.
    `DSolve[{x'[t]==2x[t]+y[t],y'[t]==x[t]-2y[t]}, {x[t],y[t]}, t]` (This gives the general solution of the system $\frac{dx}{dt} = 2x + y$, $\frac{dy}{dt} = x - 2y$.)
    `DSolve[{x'[t]==2x[t]+y[t],y'[t]==x[t]-2y[t], x[0]==2, y[0]==1}, {x[t],y[t]}, t]` (This gives the solution of the IVP system $\frac{dx}{dt} = 2x + y$, $\frac{dy}{dt} = x - 2y$, $x(0) = 2$, $y(0) = 1$.)

    (d) Sometimes the ODE can only be solved numerically or you may just want to graph the solution. In that case the command `NDSolve` is used and its output is graphed. The syntax for this command is `NDSolve[{y'[x]==`$f(x, y)$`, y[`x_0`]==`y_0`}, y[x], {x, `x` min, `x` max}]`.
    `NDSolve[{y'[x]==y[x]-x^2,y[0]==2}, y[x], {x, -3,3}]` (Now, Mathematics has the numerical solution, a collection of points on the solution, for $-3 \le x \le 3$.) To graph it, do the following.
    `Plot[Evaluate[y[x] /. %], {x, -3,3}]` (`/.` means replace all and `y[x] /. %` means replace `y[x]` with the last output.)

    (e) To graph the solution of the IVP system $\frac{dx}{dt} = -x + x^3$, $\frac{dy}{dt} = -2y$, $x(0) = 0.5$, $y(0) = 1$, do the following.
    ```
    NDSolve[{x'[t]==2x[t]+y[t],y'[t]==x[t]-2y[t],x[0]==2,y[0]==1},{x[t],y[t]},{t,-2,2}]
    ParametricPlot[Evaluate[{x[t], y[t]} /. %], {t, -2, 2}]
    ```

15. Continued.

(f) We can graph the direction field and a solution together. The following is the direction field for $\frac{dy}{dx} = y - x^2$ and graph of the solution of the IVP for the same ODE with the initial condition $y(0) = 2$.

```
graph1=VectorPlot[{1, y - x^2}, {x, -3, 3}, {y, -4, 4}]
NDSolve[{y'[x]==y[x]-x^2, y[0]==2}, y[x], {x, -3, 3}]
graph2=Plot[Evaluate[y[x] /. %], {x, -3, 3}]
Show[graph1, graph2]
```

(g) To combine a direction field with graphs of solutions of several IVP's with the same ODE, use the `Table` command. The following is as in the last part except that initial conditions are $y(0) = n$ for $n = -3, -2, -1, 0, 1, 2, 3$.

```
graph1=VectorPlot[{1, y - x^2}, {x, -3, 3}, {y, -4, 4}]
Table[NDSolve[{y'[x]==y[x]-x^2, y[0]==n}, y[x], {x,-3,3}],{n,-3,3,1}]
graph3=Plot[Evaluate[y[x] /. %], {x,-3,3}]
Show[graph1, graph3]
```

16. Given a recurrence or difference equation $a_{n+1} = f(a_n)$ with initial condition a_0, Mathematica can be used to find its terms, graph the terms and even solve it. Consider the recurrence relation $a_{n+1} = 0.5a_n + 1$, $a_0 = 3$.

(a) The syntax to find the numerical values of a recurrence relation for $n = 0$ to $n = m$ is `RecurrenceTable [{a[n+1]==`$f(a_n)$`, a[0]==`a_0`}, a, {n, 0, m}]`. Try the following.
`Clear[a,n]` This clears the a and n values used above.
`ReccurenceTable[{a[n+1]==0.5 a[n]+1, a[0]==3, a, {n,0,5}]` This gives the values of a_0 through a_5.

(b) We can graph this simple recurrence sequence as follows.
`a[0]=3;`
`a[n_]:=0.5 a[n-1]+1` We subtracted one from the index so we could define a as a function of n.
`ListPlot[Table[{n, a[n]},{n,0,5}]]`

(c) We can also graph several simple recurrence sequences with different initial values a_{01}, a_{02}, ..., a_{0k} for $n = 0$ to $n = m$ using `ListPlot[Table[Table[{n, a[n]},{n, 0, m}],{a[0],{`a_{01}`, `a_{02}`, ... , `a_{0k}`}}]]`.
`a[n_]:=0.5 a[n-1]+1` This function definition was not necessary since we did it in the last step.
`ListPlot[Table[Table[{n, a[n]},{n,0,5}],{a[0],{-1,3,6}}]]` This graph three sequences with initial conditions -1, 3, and 6.

(d) We can solve a recurrence or difference equation by `RSolve[{a[n+1]==`$f(a_n)$`, a[0]==`a_0`}, a[n], n]`.
`Clear[a]`
`RSolve[{a[n+1]==0.5 a[n]+1,a[0]==3},a[n],n]` The solution simplifies to $a_n = 2 + 2^{-n}$.

17. Similarly, we can work with a system of recurrence or difference equations.

(a) Consider the recurrence system $a_{n+1} = a_n(1.1 - 0.01b_n) + 1$, $b_{n+1} = b_n(-0.1 + 0.05a_n)$, $a_0 = 30$, $b_0 = 5$. The following gives the values of (a_0, b_0) through (a_5, b_5).
`Clear[b]` This clears the b value used above.

```
RecurrenceTable[{a[n+1]==a[n](1.1-0.01b[n]), b[n+1]==b[n](-0.1+0.05a[n]),
                                              a[0]==30,b[0]==5}, {a,b}, {n,0,5}]
```

17. Continued.

    (b) We can graph the above simple system of recurrence equations as follows.

    ```
    a[0]=30;b[0]=5;
    ```

    `a[n_]:= a[n-1](1.1-0.01b[n-1]);b[n_]:=b[n-1](-0.1+0.05 a[n-1])` We subtracted one from the index so we could define a and b as a function of n.

    ```
    ListPlot[Table[{a[n], b[n]},{n,0,5}]]
    ```

    (c) We can only solve very simple systems of recurrence equations. Consider $a_{n+1} = b_n + n + 1$, $b_{n+1} = a_n - n - 1$, $a_0 = b_0 = 1$. Input the following.

    `Clear[a,b]` This clears a and b values found above.

    ```
    RSolve[{a[n+1]==b[n]+n+1, b[n+1]==a[n]-n-1,  a[0]==b[0] == 1}, {a[n], b[n]}, n]
    ```

18. To work with complicated recurrence relations or systems of recurrence relations or when many terms need to be calculated, we need to avoid recalculating all previous terms each time a new term is calculated.

    (a) The syntax to find and plot the numerical values of a recurrence relation for $n = 0$ to $n = m$ is

    `xvalues = Table[n, {n, 0, m}];`
    `yvalues = RecurrenceTable[{a[n] ==` $f(a_{n-1})$`, a[0] ==` a_0`}, a, {n, 0, m}]`
    `points = Transpose[{xvalues, yvalues}]`
    `ListPlot[points]`

    Now consider $a_{n+1} = a_n^{1.01} + 0.25(-1)^n a_n$, $a_0 = 5$ and input the following to graph a_0 through a_{50}.

    ```
    xvalues = Table[n, {n, 0, 50}];
    yvalues = RecurrenceTable[{a[n]==a[n-1]^1.01+0.25(-1)^n a[n-1], a[0]==5}, a, {n,0,50}];
    points = Transpose[{xvalues, yvalues}];
    ListPlot[points]
    ```

    (b) Consider the system of recurrence relations $a_{n+1} = a_n(1.1 - 0.01b_n)$, $b_{n+1} = b_n(-0.1 + 0.05a_n)$, $a_0 = 30$, $b_0 = 5$. The following will plot the ordered pairs (a_0, b_0) through (a_{50}, b_{50}).

    ```
    points=RecurrenceTable[{a[n+1]==a[n](1.1-0.01b[n]), b[n+1]==b[n](-0.1+0.05a[n]),
                                  a[0]==30,b[0]==5}, {a,b}, {n,0,50}];
    ```

    We needed to name the list of values, in order to recall it.

    ```
    ListPlot[points, PlotRange->All]
    ```

    (c) We can also plot values of a and b separately as a function of n. Try the following.

    ```
    points=RecurrenceTable[{a[n+1]==a[n](1.1-0.01b[n]), b[n+1]==b[n](-0.1+0.05a[n]),
                                  a[0]==30,b[0]==5}, {a,b}, {n,0,50}];
    ```

    This was not necessary, since we have already done it.

    `apoints=Table[{i-1,points[[i,1]]},{i,1,51}];` The command `points[[i,1]]` picks the first element in the ith ordered pair in points.

    `bpoints=Table[{i-1,points[[i,2]]},{i,1,51}];` The command `{i-1,points[[i,2]]}` forms ordered pair (n, b_n) for each of the n values.

    ```
    ListPlot[{apoints, bpoints},PlotStyle->{{Red,PointSize[0.02]},
                                    {Blue, PointSize[0.01]}}, PlotRange->All]
    ```

    The a values are the larger red points.

19. It is essential, as a courtesy to future users, and continued problem-free usage, to leave the computers as you find them. When you are done, close the application and/or log off properly.