MATLAB Tutorial for Calculus III - Part 1

Contents

1 Introduction to MATLAB

MATLAB is a powerful computer algebra system that can perform many mathematical calculations. It can also be used as a programming language. This lab is intended to introduce both of these capabilities. The MATLAB program is available to UCCS students in the computer lab in EAS 136.

1.1 Logging into the system

On the log-in screen, you will see a prompt for your username and password. Type in your username first. This will be the first letter of your first name followed by the first seven letters (or less) of your last name. For example, if your name is John Williams, then your username will be JWilliam.

If you have not previously logged in, you will need the initial password. The initial password is MmmssssL where:

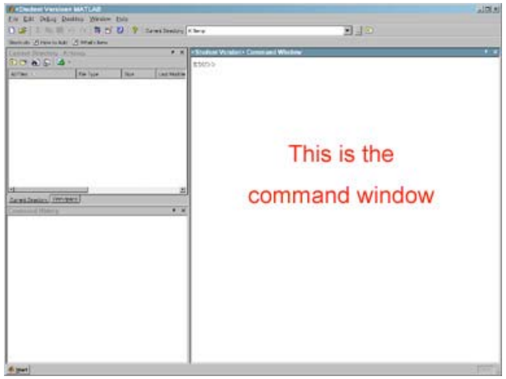
- Mmm = first 3 letters of your birthmonth
- ssss = last 4 digits of your student ID number
- L = first letter of last name
- (The first and last letters are capitalized.)

After you log in you will be asked to change this password, make sure you remember it!

****Note:** If for some reason either your username or your password does not work, first check that the DOMAIN is set to UFP. If it still doesn't work, ask someone for help.

1.2 Starting out with MATLAB

Double click the MATLAB icon on your screen. This will start the MATLAB program, and the MATLAB desktop appears.



The MATLAB desktop is divided into three smaller windows. On the upper left you will see the "Current Directory" subwindow. We will not be using this, so remove it by clicking the small "x" in the upper right corner of the subwindow. Now the "Workspace" subwindow appears; remove this also. Then remove the "Command History" subwindow. Now the only subwindow that is left is the "Command" subwindow. You should see a prompt in the Command window that looks like this: ">>".

2 MATLAB as scientific calculator

2.1 Introduction to calculations

The prompt is where you enter commands in MATLAB. Now we will look at a few of the functions MATLAB can perform. First, notice that MATLAB can function as an expensive calculator. Try typing the following:

```
>> 2+2

ans =

     4
```

If you did this correctly, you should get the answer displayed underneath.

****Note:** You may end commands in MATLAB with a semicolon. If you end a line with a semicolon, the command will be executed, but the solution will not be displayed. If you don't end a line with a semicolon (as above), the solution will be displayed below the line.

We can continue with multiplication. The asterisk symbol "*" is used to indicate multiplication. You cannot leave it out. Try typing the following line:

```
>> 2*2

ans =

     4
```

If you instead typed "(2)(2)" you would get an error message (try it!).

Subtraction should be written as follows:

```
>> 5-3

ans =

     2
```

Division is indicated by the "/" character:

```
>> 5/3

ans =

    1.6667
```

You can can use several operators in one line and group expressions using parenthesis. For example:

```
>> (1+7-3)*2

ans =

    10
```

Now try a few of your own calculations!

2.2 Assigning variables in MATLAB

It is often convenient to name a long expression for later use. That way, you don't have to repeatedly type in the long expression. Assignment in MATLAB is done with the operator "=". For example, the following line assigns the letter "a" to the approximate value of pi:

```
>> a = 3.14

a =

    3.1400
```

Now every time the letter "a" is used in an expression, it will actually refer to the value for pi. Try typing the following:

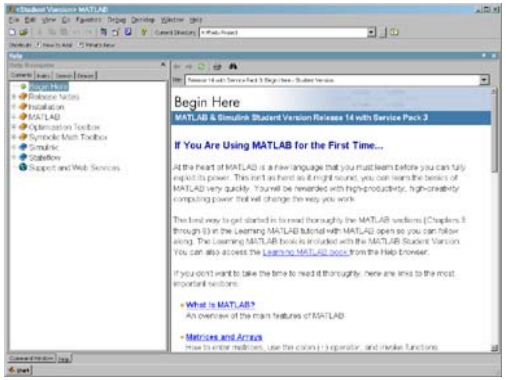
```
>> a

a =

    3.1400


>> 2*a

ans =

    6.2800
```

2.3 Help in MATLAB

In the example above, the value for pi was assigned to a variable to save it. Does the value for pi have to be entered everytime it is used? Maybe MATLAB has a better choice.

To find help in MATLAB, click the HELP icon:

The Help window now appears on the MATLAB desktop.



Begin Here

MATLAB & Simulink Student Version Release 14 with Service Pack 3

If You Are Using MATLAB for the First Time...

At the heart of MATLAB is a new language that you must learn before you can fully exploit its power. This isn't as hard as it might sound; you can learn the basics of MATLAB very quickly. You will be rewarded with high-productivity, high-creativity computing power that will change the way you work.

The best way to get started is to read thoroughly the MATLAB sections (Chapters 1 through 8) in the Learning MATLAB tutorial with MATLAB open so you can follow along. The Learning MATLAB book is included with the MATLAB Student Version. You can also access the Learning MATLAB book from the Help browser.

If you don't want to take the time to read it thoroughly, here are links to the most important sections:

- What Is MATLAB?
  An overview of the main features of MATLAB
- Matrices and Arrays
  How to enter matrices, use the colon (:) operator, and invoke functions

You can browse the table of contents by clicking on the ”Contents” tab in the ”Help Navigator” window. This is very similar to the help used in other programs. Simply use the mouse to click on the topic of your choice.

You can also select the ”Index” tab and then search the index for a given word or phrase by typing it in the ”search index for” text box. Try searching the index for ”pi”. You will find three entries for ”pi” in the index, labeled [1], [2], and [3]. Click on ”[3]” and you will see a list of special values you can use in expressions, one of which is ”pi”. So you have learned that there is no need to type the value of pi everytime it is used. Instead, substitute the constant, ”pi”. For example:

```
>> pi/2

ans =

    1.5708
```

2.4 Exponents in MATLAB

If you wish to find, for example, 13 to the fifth power, then you can easily do so in MATLAB. The caret ”^” (or ”hat”) is used in exponentiation. Try typing:

```
>> 13^5

ans =

      371293
```

Remember that you can also use the exponent operator to compute the roots of numbers. If you want to find the cube root of 876, simply type:

```
>> 876^(1/3)

ans =

    9.5683
```

2.5 Built-in MATLAB functions

MATLAB has the same built in functions that you see in scientific calculators. A few examples are, sin(x), cos(x), tan(x), log(x), sqrt(x), and exp(x). Try typing the following:

```
>> log(10)

ans =

    2.3026
```

This answer tells you that log(x) does not mean the base 10 logarithm, but the natural logarithm. If you need the base 10 logarithm, use the log10(x) function.

The trigonometric functions should be entered in the same way, for example, sin(pi/2) should be as below:

```
>> sin(pi/2)

ans =

     1
```

Now try typing sinpi/2, omitting the parentheses. What is the result?

MATLAB also recognizes the inverse trigonometric functions. For instance, the inverse tangent is ”atan”. Recalling that the inverse tangent of 1 is pi/4, we can enter it as below to see that the arctan function works as expected:

```
>> 4*atan(1)

ans =

    3.1416
```

Notice that by default MATLAB displays numeric results to 4 decimal places. For example:

```
>> 4/3

ans =

    1.3333
```

You can get 14 decimal places by typing ”format long”:

```
>> format long
>> 4/3

ans =

   1.33333333333333
```

To change back to 4 decimal places, type ”format short”:

```
>> format short
>> 4/3

ans =

    1.3333
```

Now, let’s look at e. Try computing e to the tenth power by typing ”e^10”

Well, that doesn’t work. In MATLAB, e^x is not a valid command. MATLAB is treating the letter ”e” as an undefined variable, just like it would ”x”. To find e^x, use the function, exp(x):

```
>> exp(10)

ans =

  2.2026e+004
```

2.6 Exercises

1. Compare and explain the following expressions:
   (a) 2+4/2
   (b) (2+4)/2
   (c) 2+(4/2)
   (d) 2(3+5)
2. Find the numerical values of the following:
   (a) The fifth root of 9.8
   (b) 2 to the eighth power
   (c) The angle whose tangent is 0.5 (in both degrees and radians)
   (d) The cosine of pi/4
   (e) e to the 13
   (f) The square root of 488

3 Symbolic computations

3.1 Introduction to symbolic variables

So far, we have been using MATLAB as a fancy numerical calculator. We were able to perform evaluations such as

```
>> 87*3^2 - (56+42)/11


ans =

  774.0909
```

But MATLAB also lets you do **symbolic** mathematics.

Suppose you have the equation $y = 2x + 5$. The letters x and y are symbols that represent numbers. You apply a set of rules to manipulate these symbols without knowing the particular numbers these symbols represent.

MATLAB also lets you define symbols that represent numbers without specifying the particular numbers the symbols represent. These are called ”symbolic variables”. Symbolic variables may be defined using the ”syms” command. Define ”x” as a symbolic variable:

```
>> syms x
```

And now define another symbolic variable in terms of x:

```
>> y = 2*x + 5
```

```
y =

2*x+5
```

(Note: you must type the ”*” to indicate multiplication.) Now if you type ”y” you don’t get ”y = some number”, you get:

```
>> y

y =

2*x+5
```

That is, the symbolic variable ”y” doesn’t represent any number in particular; it represents an expression involving the symbolic variable x.

****Note:** The variable ”y” is symbolic because we defined it in terms of another symbolic variable. That is, if the expression to the right of the equals sign contains a symbolic variable, then the new varaible will be defined as symbolic. If, however, everything to the right of the equals sign is numeric, then MATLAB will define a numeric variable.

Now you can define other symbolic variables in terms of x. For example, you can define the symbolic variable f to be a polynomial in x:

```
>> f = 3*x^2 + 4*x - 7

f =

3*x^2+4*x-7
```

Now you can perform operations on this symbolic expression. Try multiplying it by 2:

```
>> 2*f

ans =

6*x^2+8*x-14
```

You can also multiply f by another expression, say (x-3), in order to define a new polynomial, g. Type:

```
>> g = f*(x-3)
```

```
g =

(3*x^2+4*x-7)*(x-3)
```

MATLAB doesn’t expand the new expression but leaves it as we defined it in factored form. To expand this new expression, use the ”expand” function:

```
>> expand(g)

ans =

3*x^3-5*x^2-19*x+21
```

3.2 Finding solutions to equations

Above we defined the quadratic function $f = 3x^2 + 4x - 7$. We can use MATLAB to find a solution to this function. The MATLAB function ”solve(expression,variable)” will try to find the values of ”variable” which satisfy the equation ”expression=0”. For example, type ”solve(f,x)”:

```
>> solve(f,x)

ans =

    1
 -7/3
```

MATLAB finds the two values of x which satisfy $f = 0$, in this case 1 and -7/3.

Of course, sometimes no solutions to a given equation exist. To see what MATLAB does in this case, try solving the equation $1/x = 0$ for x. You should get the message ”Explicit solution could not be found”. (Try it!)

Note that the solve function will try to find a numeric solution if possible. If no numeric solution is possible, solve will try to find a symbolic solution.

You can also try factoring expressions using the ”factor” function. Try factoring the function f we defined above:

```
>> factor(f)

ans =

(3*x+7)*(x-1)
```

Also try factoring the expression g we defined above:

```
>> g
>> factor(g)


g =

(3*x^2+4*x-7)*(x-3)



ans =

(3*x+7)*(x-1)*(x-3)
```

3.3 Substitution

You can substitute a numerical value for a symbolic variable using the ”subs” function. For example, define the symbolic expression:

```
>> f = 2*x^2 - 3*x + 1


f =

2*x^2-3*x+1
```

To substitute the value $x = 2$ in the expression for f, enter the command:

```
>> subs(f,2)


ans =

     3
```

This returns $f(2)$, which has the value 3.

When your expression contains more than one variable, you can specify the variable for which you want to make the substitution. For example, define:

```
>> syms x y
>> f = x^2*y + 5*x*sqrt(y)
```

```
f =

x^2*y+5*x*y^(1/2)
```

To substitute the value $x = 3$ in the expression for f, enter:

```
>> subs(f,x,3)

ans =

9*y+15*y^(1/2)
```

One way to remember how the ”subs” function works is to remember subs(f,OLD,NEW).

3.4 Symbolic versus numeric computation

What happens if we calculate ”sin(2*pi)”?

```
>> format long
>> pi

ans =

 3.14159265358979


>> sin(2*pi)


ans =

 -2.449293598294706e-016
```

The result is a very small number but is not zero, as we expected. This is because ”pi” as defined by MATLAB is a finite-precision **numeric** variable, and is not exact. However, calculation with **symbolic** variables is exact. To convert numeric pi into symbolic pi, use the ”sym” function:

```
>> Pi = sym(pi)

Pi =

pi
```

This defines the symbolic variable ”Pi” (note the capital ”P”: MATLAB variables are case-sensitive, so ”Pi” is different than ”pi”). Now we can get the exact value of sin(2*Pi):

```
>> sin(2*Pi)

ans =

0
```

3.5 Symbolic calculus with MATLAB

You can take the derivative of an expression with the ”diff” function. For example, to find the first derivative of $y = x^4$ with respect to x

$$\frac{d}{dx}x^4$$

type these commands:

```
>> syms x
>> y = x^4
>> diff(y,x)

y =

x^4


ans =

4*x^3
```

A slightly more general example is:

$$\frac{d}{dx}x^n$$

which can be entered as:

```
>> syms x n
>> y = x^n
>> diff(y,x)
```

```
y =

x^n


ans =

x^n*n/x
```

Notice that MATLAB didn’t simplify the result, but you can do this using the ”simplify” function:

```
>> simplify(ans)

ans =

x^(n-1)*n
```

MATLAB knows all the rules of differentation, so you can really make it work for you. For example, use MATLAB to compute:

$$\frac{d}{dx}ln(x^3sin^2(x))$$

```
>> diff(x^3*log(sin(x)^2),x)

ans =

3*x^2*log(sin(x)^2)+2*x^3/sin(x)*cos(x)
```

You can do integrals

$$\int x^n dx$$

using the ”int” function:

```
>> syms x n
>> y = x^n
>> int(y,x)
```

```
y =

x^n


ans =

x^(n+1)/(n+1)
```

And you can specify the limits of integration as well:

$$\int_0^3 x^4 dx$$

```
>> int(x^4,x,0,3)
```

```
ans =

243/5
```

3.6 Exercises

1. Find solutions to:
   (a) $x^3 - 683x^2 + 11374x - 34632 = 0$
   (b) $3465x^3 - 14866x^2 + 13623x - 3542 = 0$
   (c) $x^2 - 2x + 2 = 0$
2. Define the symbolic variables a, b, c, and x; then find the roots of the quadratic equation $ax^2+bx+c = 0$. Do you recognize the quadratic formula? What happens if you try to find the roots of the cubic equation $ax^3 + bx^2 + cx + d = 0$?
3. Find the roots of the polynomial $x^7 + 2x^6 + 3x^5 + 4x^4 + 5x^3 + 6x^2 + 7x + 8 = 0$.
4. Find the critical points of $f(x) = x^3 - 27x$, that is, find the values of x where $df/dx = 0$. Choose one of these critical points and find the numeric value of d^2f/dx^2 at this point. Is $f(x)$ a local maximum or a local minimum at this point?

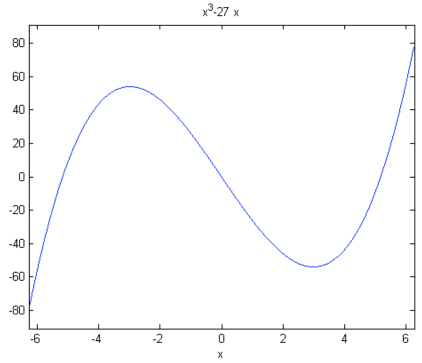
4 Plotting

4.1 Making 2D plots

You can use the "ezplot" function to make 2D graphs from your symbolic variables. For example:

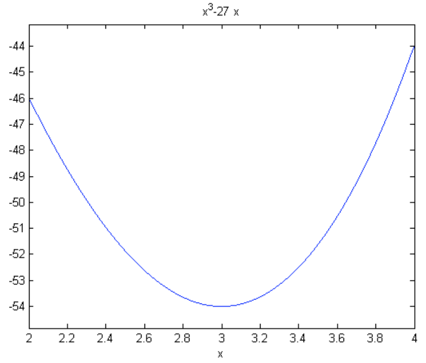
```
>> y = x^3 - 27*x
>> ezplot(y)
```

```
y =

x^3-27*x
```



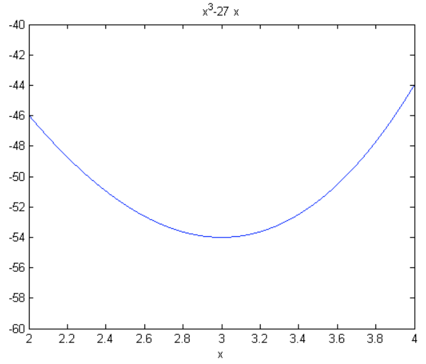
By default, ezplot graphs from x=-2*pi to x=+2*pi. You can change the domain by adding a second argument to the call to ezplot. For example, to plot the above function from x=2 to x=4 you type:

```
>> ezplot(y,[2,4])
```



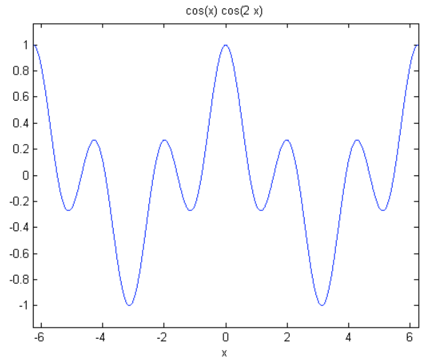
You may also change the limits of the y axis by using the ”ylim” function after you have made the graph. For example:

```
>> ezplot(y,[2,4])
>> ylim([-60,-40])
```

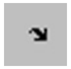


Note that you don’t have to define y first to use ezplot; you can just type in the function you want to graph:

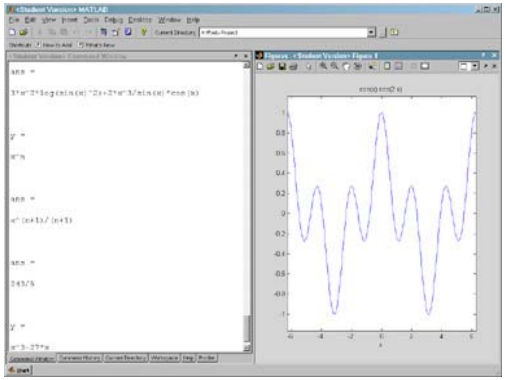
```
>> ezplot(cos(x)*cos(2*x))
```



After you have made a graph you can click on the ”dock” icon



to ”dock” the graph


<Student Version> MATLAB
File Edit View Insert Tools Debug Desktop Window Help
Current Directory
Shortcuts How to Add What's New
<Student Version> Command Window
Figures - <Student Version> Figure 1
Command Window Command History Current Directory Workspace Help Profiler
Start

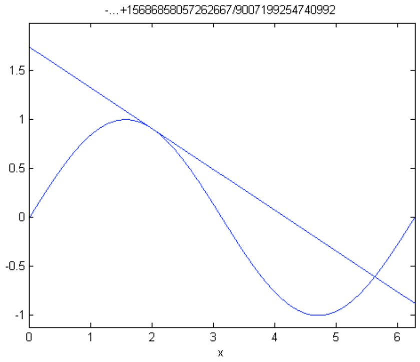
```
ans =

3*x^2*log(sin(x))^2+2*x^3/sin(x)*cos(x)


y =

x^n


ans =

x^(n+1)/(n+1)


ans =

243/5


y =

x^3-27*x
```

4.2 Example: a function and the tangent line on the same plot

In this example, we will plot the function $y = sin(x)$ and then show the tanget line to $y(x)$ at $x = 2$. Recall that the equation of the tangent line at $x = x_0$ is: $y - y_0 = m(x - x_0)$ where m is the slope of $y(x)$ at $x = x_0$.

We will make use of the "hold on" and "hold off" commands. The "hold on" command tells MATLAB to overlay all subsequent plots in the previous figure window. In this way we can put two curves on the same graph. The "hold off" command restores the default behavior.

```
>> syms x;    % x is the symbolic variable
>> y = sin(x);    % y is the function of x we want to plot
>> ezplot(y, [0,2*pi]);    % plot y(x) from x=0 to x=2*pi
>> dydx = diff(y);    % symbolic variable 'dydx' is the first derivative of y
>> m = subs(dydx, x, 2);    % m is slope of y(x) at x=2
>> y0 = subs(y, x, 2);    % y0 is value of y(x) at x=2
>> hold on;    % draw the next line on the same graph
>> ezplot(m*(x-2)+y0, [0,2*pi]);    % plot the tangent line at x=2
>> hold off;
```



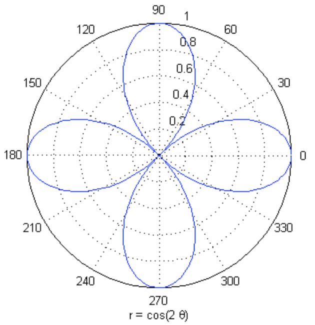
We could also make the tangent line a different color than $y(x)$. In order to do this we must obtain what is called a "handle" to the plot. A handle is just a pointer to a data structure that describes the graphics object. To obtain the handle we call ezplot as follows: handle = ezplot(...); Then we can use this handle to change the color using the "set" function.

```
>> handle = ezplot(m*(x-2)+y0, [0,2*pi]);    % plot the tangent line at x=2
>> set(handle, 'Color', 'Red');    % color the tangent line red
```

4.3 Polar plots

MATLAB will also plot functions in polar coordinates. To do this you use the "ezpolar" function:

```
>> syms theta
>> ezpolar(cos(2*theta))
```



4.4 Curves in space

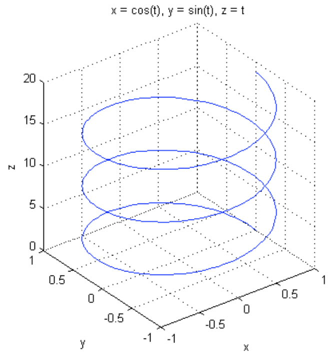
You can use the ”ezplot3” function to graph curves in space. For example, we can define a helix by the parametric equations:

$$x = cos(t),\ y = sin(t),\ z = t$$

(see Stewart section 14.1 example 4)

To plot this curve you would type:

```
>> syms t
>> ezplot3(cos(t),sin(t),t,[0,6*pi])
>> axis square;
```



4.5 3D vectors

You can define a vector in Matlab by entering the components of the vector. For example, if you want to define the vector **a** with components 2, -3, and 4 you would enter:

```
>> a = [ 2 -3 4]


a =

     2    -3     4
```

You enclose the components of the vector within square brackets and seperate the components with either blanks or commas.

4.6 Equation of a plane

Matlab will also create 3D plots. Here you will plot the plane through the point (1, -1, 1) with normal vector $\mathbf{i} + \mathbf{j} + \mathbf{k}$.

The vector equation of a plane is given by

$$\mathbf{n}\cdot(\mathbf{r} - \mathbf{r_0}) = 0$$

(see Stewart p. 861)

First, define the normal vector:

```
>> n = [1 1 1]


n =

     1     1     1
```

Next, define the point (1, -1, 1):

```
>> r0 = [1 -1 1]


r0 =

     1    -1     1
```

Finally, define the vector r:

```
>> syms x y z
>> r = [x y z]


r =

[ x, y, z]
```

Use the ”dot” function to take the dot product of two vectors. You can now enter an expression for the left-hand side of the vector equation:

```
>> dot(n,r-r0)
```

```
ans =

x+y+z-1
```

Note that Matlab has evaluated this expression in terms of the symbolic objects x, y, and z. So the equation for the plane becomes:

$$x + y + z - 1 = 0$$

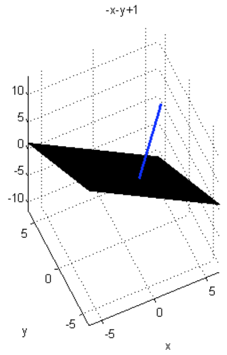
In order to have Matlab plot this plane, you need to tell it how to compute the value of z for each x and y, that is, you need to write the equation of the plane in the form that gives z as a function of x and y. You can use the "solve" function to have MATLAB do this for you. Here we want to solve "x+y-z+1=0" for z, so you type:

```
>> solve( x+y+z-1, z )
```

```
ans =

-x-y+1
```

Now you can plot the plane using the "ezsurf" function:

```
>> ezsurf( 1-x-y )
>> axis equal;
>> view(-25,62); % change viewpoint so we can see the normal vector
>> hold on;
>> quiver3(r0(1), r0(2), r0(3), n(1), n(2), n(3), 4, 'LineWidth', 2);
>> hold off;
```



In this plot, the surface is assigned colors as follows: The highest point on the surface (that is, the point with the largest value of z) is colored dark red and the lowest point is colored dark blue. Intermediate values of z are assigned the colors in the order red, orange, yellow, green, cyan, and blue.

You can rotate this surface in real-time. Click on the rotate icon:

and move the cursor over the surface. Now click and drag and the surface will rotate as you move the cursor.

We have used the "quiver" function to draw the vector normal to the plane. As used here, the function draws the vector **n** with the tail of **n** at the point **r0**. Then we magnified **n** by a factor of 4 to make it easier to see.

4.7 Two intersecting planes

Here you will plot two planes which intersect, x+y+z=1 and x-2y+3z=1 (Stewart example 7, p. 863).

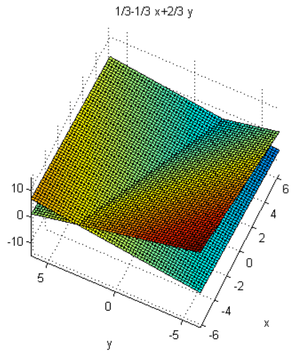
First, solve the equations of the two planes for z, either by hand or by using the "solve" function:

$$z = 1 - x - y$$

$$z = \frac{1 - x + 2y}{3}$$

Now you can use the "ezsurf" function to plot the planes. To place both planes on the same plot, type "hold on" between the calls to "ezsurf":

```
>> ezsurf(1-x-y)
>> view(-65,65)
>> hold on
>> ezsurf((1-x+2*y)/3)
>> axis square;
>> hold off
```



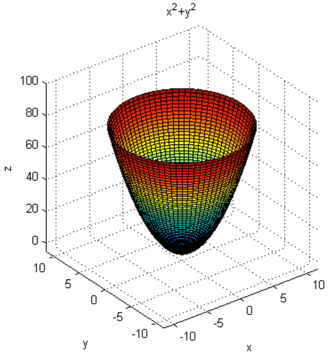
4.8 Surfaces in space

It is easy enough to plot other surfaces in space. A paraboloid is defined by:

$$z = x^2 + y^2$$

To plot this, type:

```
>> ezsurf(x^2+y^2,'circ')
>> axis square;
```



The "circ" argument tells MATLAB to use a circular disk as the domain.

4.9 Exercises

1. Graph the function:

$$y(x) = cos(x) + \frac{1}{1 + e^{-2x}}$$

   How many zeros of this function exist between $x = -4$ and $x = 4$?

2. Make a graph of the function:

$$y(x) = \frac{2 + 2x}{5sin(x) - 6}$$

   from $x = -16$ to $x = 10$. On the same plot, draw the tangent line to $y(x)$ at x=-6.

3. On the same plot, graph $y = sin(x)$ and the first three Taylor approximations to $sin(x)$ at $x_0 = 0$.