M427L MATLAB Guide/ Problems Part I
Instructor: Jacky Chong

This guide has been adapted and copied with permission from material created by Justin Wyss-Gallifent for his multivariable calculus course taught at the University of Maryland, College Park. The basic format of this guide is that you will sit down with this page open and a MATLAB window open and you will try the guide as you go. Once you are finished following through the guide the assignment should be very straightforward.

Contents

Downloading and Running MATLAB

To install MATLAB onto your personal devices (laptops and desktops), please follow the instructions on this page to submit a request to download the program. Once you submit a request, you should receive an email (within a few minutes) on your university email account. Follow the instructions on the email to complete the installation.

Okay, I'm running MATLAB, now what?

When you run MATLAB you will see a bunch of windows. The important one will have a prompt in it which looks like >> called the Command Window. This is where we will tell MATLAB what to do. Go into this window and type 2+3. You will see:

```
2+3
```

```
ans =

     5
```

Oh yeah, you know MATLAB! Let's do something more relevant to calculus like take a derivative. Before we do that it's important to note that MATLAB works a lot with what are called *symbolic expressions*. If we want to take a derivative we use the `diff` command. It's tempting to do `diff(x^2)` but produces an error. Try it and see!

Why does it error? The reason is that MATLAB doesn't know what `x` is and we must tell it that `x` is a symbol to work with. We do so with the `syms` command. So we can do the following two commands. The first line tells MATLAB that `x` is symoblic and will be symbolic until we tell it otherwise.

```
syms x
diff(x^2)
```

```
ans =

2*x
```

Eat your heart out Newton and Leibniz! Here's an even uglier derivative. Keep in mind that we don't need a `syms x` now because `x` is still symbolic and will remain so. Notice something important - that multiplication in MATLAB always needs a `*`. Leaving this out will give an error.

```
diff(2*x^2*sin(x)/cos(x))
```

```
ans =

2*x^2 + (2*x^2*sin(x)^2)/cos(x)^2 + (4*x*sin(x))/cos(x)
```

Oh boy, that's pretty messy. Can we simplify that at all? Indeed we can. We can wrap it in the MATLAB command simplify as follows:

```
simplify(2*diff(x^2*sin(x)/cos(x)))
```

```
ans =

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

Precalculus

Here are some precalculus things done in MATLAB.

Factor a polynomial:

```
factor(x^4-3*x^3-8*x^2+21*x+9)
```

```
ans =

[ x^2 + 3*x + 1, x - 3, x - 3]
```

It is clear from the output that the factor function returns an array of the irreducible factors of `x^4-3*x^3-8*x^2+21*x+9`.

Solve an equation. Consider the following command:

```
solve(x^2+x-1)
```

```
ans =

 - 5^(1/2)/2 - 1/2
   5^(1/2)/2 - 1/2
```

Notice that we gave `solve()` a symbolic expression and the solve command *assumes* that it equals `0` when it solves. We can tell it to equal something else.

```
solve(x^2+x-1==1)
```

```
ans =

 -2
  1
```

Note we have used `==` instead of the single equals sign. In some cases MATLAB will accept a single equals sign but MATLAB is currently undergoing changes to bring it in line with most programming languages. Typically single `=` are used for assigning variables whereas double `==` are used to check for equality. Solving an equation really means checking when two things are the same, hence `==`.

Since `solve(x^2+x-1)` solves for the roots of `x^2+x-1` symbolically, we have seen that the outputs are symbolic expressions. To get numerical outputs, we could recast the answer as a "Double", i.e. a decimal number.

```
double(solve(x^2+x-1))
```

```
ans =

   -1.6180
    0.6180
```

Task 1: Solve the equation $x^3 - 3x^2 - x = 0$. Give both the symbolic and numerical answers.

Substitute a value into a symbolic expression. Here we tell MATLAB to put `-1` in place of `x`. The reason it knows to put it in place of `x` is that `x` is the only variable. We'll do some stuff with multiple variables later.

```
subs(x^3+4/x+1,-1)
```

```
ans =

-4
```

Task 2: Substitute `x=3` into $\frac{1}{x} + e^x + x^2$. Note: How do you implement e^x in MATLAB?

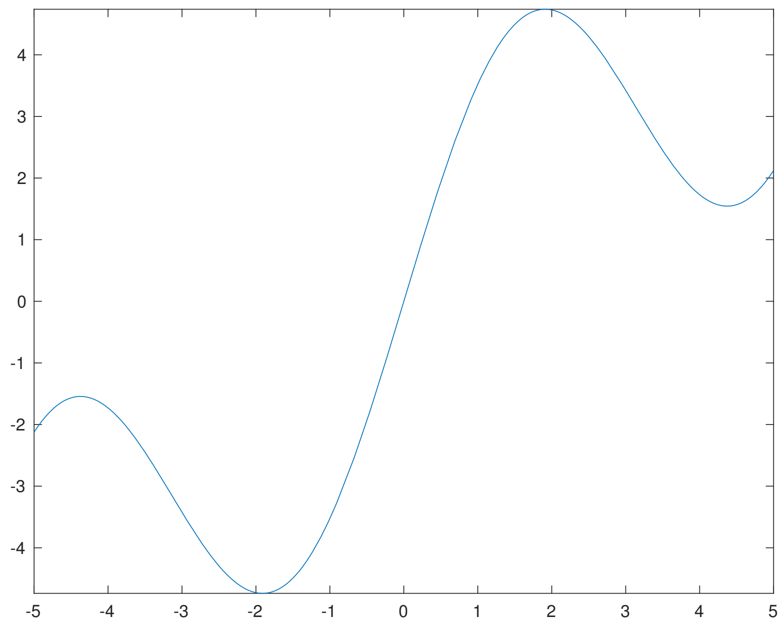
If you have multiple variables, then you should tell it which variable to substitute for. If you don't then it will make assumptions and those assumptions may confused you. So here I'll redeclare our variables just for clarity and then substitute for `y`:

```
syms x y
subs(x^2+y^2+x+y, y, 2)
```

```
ans =

x^2 + x + 6
```

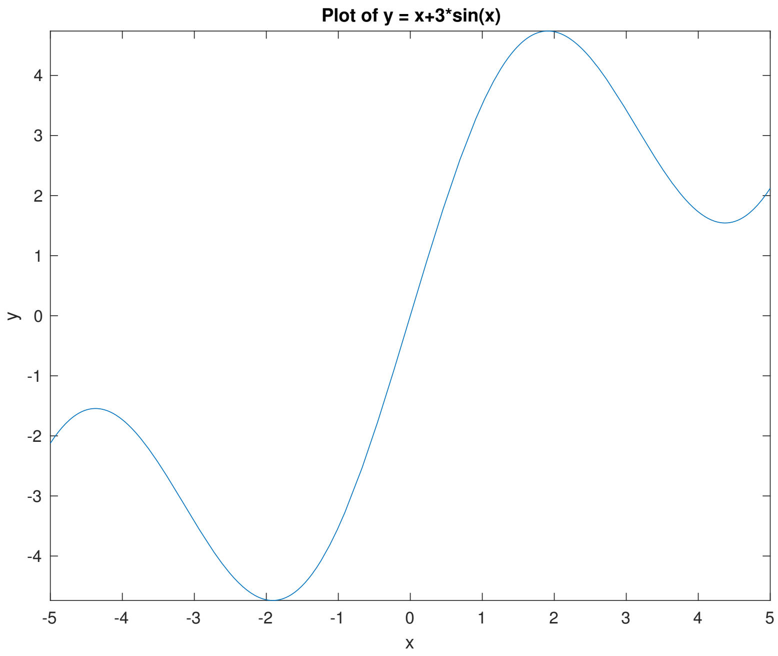
Plot a function. There are many ways to plot in MATLAB but the easiest way has the name `fplot`. When you run it you will get a new window which will pop up. In this tutorial the picture is embedded in the text. The `fplot` command tries to make reasonable choices as to the amount of the graph to show. Also it does not draw axes, instead it puts the graph in a box. I find that annoying but maybe you like boxes and other rectangular shapes. Honestly the reason MATLAB does this is so the axes stay out of the way and you can see what's going on.

```
fplot(x+3*sin(x))
```



To title the graph and label its axes, we could use `title()`, `xlabel()`, and `ylabel()` as follows:

```
fplot(x+3*sin(x))
title('Plot of y = x+3*sin(x)')
xlabel('x')
ylabel('y')
```

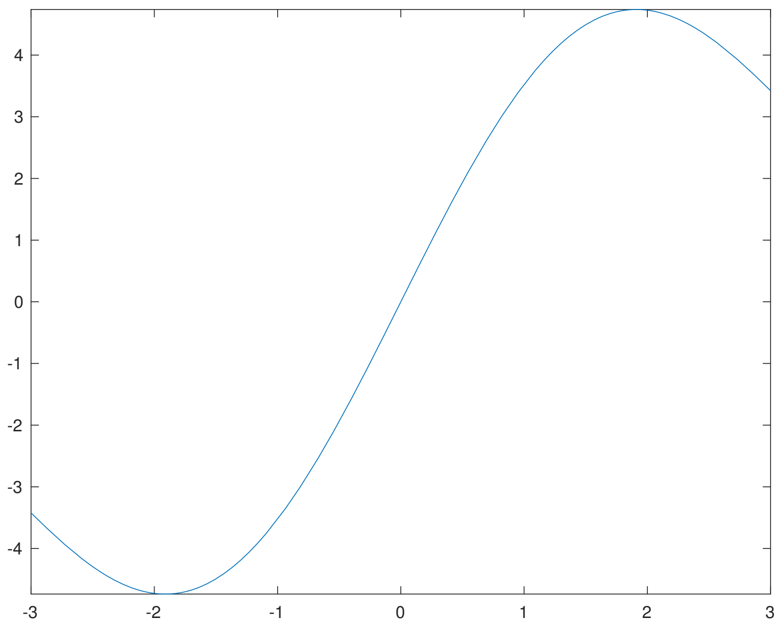


Notice the input expression in the `title()` function are in apostrophes. The purpose of the apostrophes is to cast the input into something called a ”String”, so when MATLAB runs the code it will not interpret the expression as functions or etc.

Task 3: Plot the function $v = \cos u + e^u$ and label the plot.

If you want to give it a specific domain of `x`-values you can do that too:

```
fplot(x+3*sin(x), [-3, 3])
```



Task 4: Plot the function $f(x) = x^2 + x \sin x$ on the interval $[-2\pi, 2\pi]$. Remember to label the plot.

Calculus I

We can also do some basic calculus.

Take a derivative and substitute. The following code takes the derivative of `x^3` and evaluates is at `x=3`:

```
subs(diff(x^3), x, 3)
```

```
ans =

27
```

Notice that subs is wrapped around `diff`. If we wrap `diff` around `subs` look at what happens and think about why:

```
diff(subs(x^3, x, 3))
```

```
ans =

0
```

Task 5: Find the derivative of $\frac{x^2}{x+1}$ at $x = -2$.

Find an indefinite integral. Let us find the indefinite integral of `x*sin(x)`.

```
int(x*sin(x))
```

```
ans =

sin(x) - x*cos(x)
```

Notice that MATLAB, like many careless students, never puts $a + C$ in its indefinite integrals. Grrr.

Task 6: Find the indefinite integral of $x^3 \arctan x$. Note: How do you implement $\arctan x$ in MATLAB?

Finding a definite integral. We can find the definite integral as follows:

```
int(x*sin(x), 0, pi/6)
```

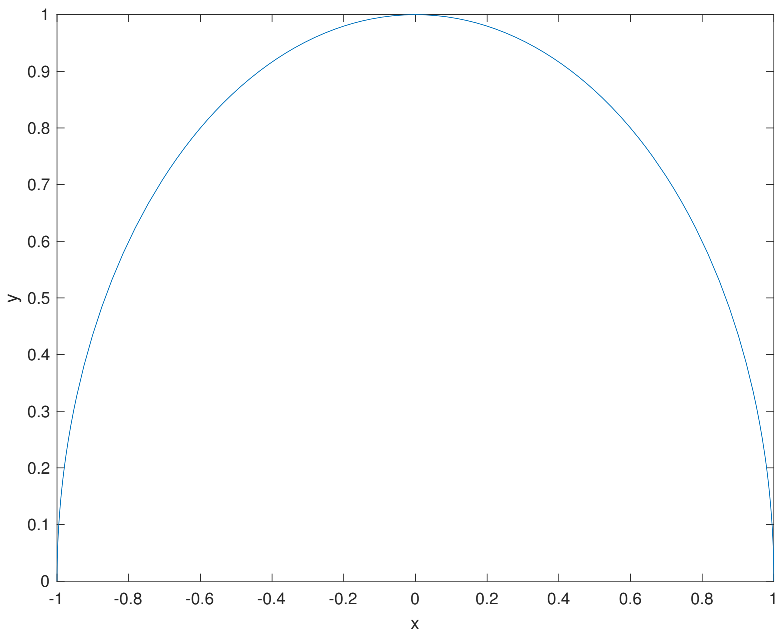
```
ans =

1/2 - (pi*3^(1/2))/12
```

Task 7: Find $\int_{-1}^{1} xe^{2x+1} \mathrm{d}x$. Give both the exact and decimal values.

Calculus II

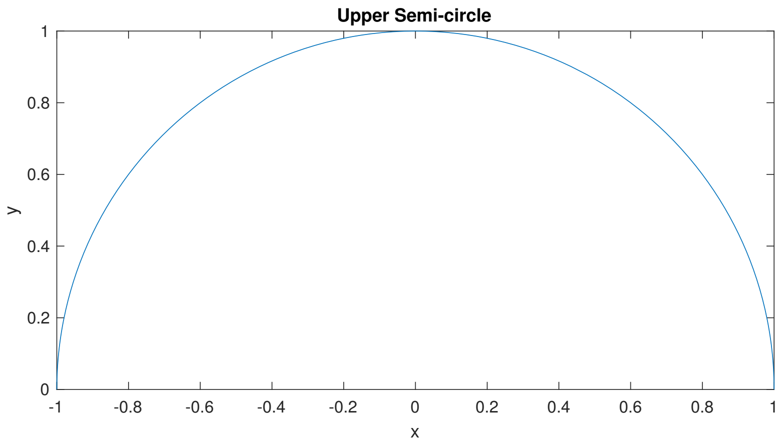
Plot the parametric curve `x=cos(t), y=sin(t)` for `0<=t<=pi`. Note that this is also a Calculus 3 problem if we think of this as plotting the vector-valued function given by $\mathbf{r}(t) = \cos(t)\mathbf{i} + \sin(t)\mathbf{j}$. Notice also the new symbolic variable `t` needs to be declared.

```
syms t
fplot(cos(t), sin(t), [0, pi])
xlabel('x')
ylabel('y')
```



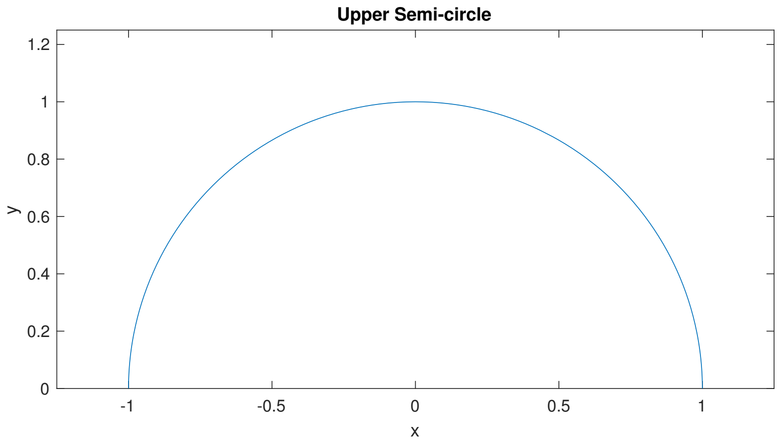
From experience, we know that the above parametric curve should plot a semi-circle. However, it is clear that the output looks more like a portion of an ellipse than a circle. We will fix this by adding the following command to the code:

```
fplot(cos(t), sin(t), [0, pi])
xlabel('x')
ylabel('y')
title('Upper Semi-circle')
axis equal % setting the scale of the axes to be equal
```



The above plot looks kind of tight. To relief the tightness, we might want increase the limits of the plot to get some white space. The simpliest way to do this is by the following:

```
fplot(cos(t), sin(t), [0, pi])
xlabel('x')
ylabel('y')
title('Upper Semi-circle')
axis equal
xlim([-1.25, 1.25]) % increase the x range of the plot
ylim([0, 1.25]) % increase the y range of the plot
```



Task 8: Plot the curve $x = \frac{t}{10}\cos t$ and $y = \frac{t}{5}\sin t$ for $0 \le t \le 4\pi$. Adjust the axis and set the output range to equal $-1.5 \le x \le 1.5$ and $-2.5 \le y \le 2.5$. Remember to label your axes!

Linear Algebra

Defining Vectors. Vectors in MATLAB are treated like in linear algebra so this may be new to you. The vector $2\mathbf{i} + 3\mathbf{j} + 7\mathbf{k}$ can be entered either as a horizontal vector or a vertical vector. I will use vertical vectors in this guide because that's how we think of them in M427L. As a vertical vector we use semicolons to separate rows:

```
u = [2; 3; 7]
```

```
u =

     2
     3
     7
```

As a horizontal vector, we use spaces or commas.

```
uComma = [2, 3, 7]
uSpace = [2 3 7]
```

```
uComma =

     2     3     7


uSpace =

     2     3     7
```

Remark: It is important to start using names and phrase to label quantities when coding instead of single letters.

Defining Matrices. Matrices are defined in a similar manner as in the case of vectors. We use semicolons to separate the different rows of the matrix. Let us consider the matrices in problem #15 in Review Exercise of Chapter 1 of Marsden and Tromba:

```
A = [1 5 2; 0 2 3; 1 0 2]

B = [2 0 1; 1 3 0; 2 4 1]
```

```
A =

     1     5     2
     0     2     3
     1     0     2


B =

     2     0     1
     1     3     0
     2     4     1
```

We can put variables inside the matrices and make substitution.

```
syms x y z
M = [x+2  y z; z y+1 10; 5 5 2]
M1 = subs(M, x, 1)
M2 = subs(M1, y, 2)
M3 = subs(M2, z, 3)
```

```
M =

[ x + 2,     y,  z]
[     z, y + 1, 10]
[     5,     5,  2]


M1 =

[ 3,     y,  z]
[ z, y + 1, 10]
[ 5,     5,  2]


M2 =

[ 3, 2,  z]
[ z, 3, 10]
[ 5, 5,  2]


M3 =

[ 3, 2,  3]
[ 3, 3, 10]
[ 5, 5,  2]
```

To compute the matrix addition A+B and matrix product AB and BA, we use `+, *` just like in the scalar case.

```
A+B

A*B

B*A
```

```
ans =

     3     5     3
     1     5     3
     3     4     3


ans =
    11    23     3
     8    18     3
     6     8     3


ans =

     3    10     6
     1    11    11
     3    18    18
```

Task 9: Solve Problem #34 in Review Exercises for Chapter 1 of Marsden and Tromba.

Transpose of Matrices. The transpose of A can be computed readily by using an apostrophes at the end:

```
C = [2 3 -2; 1 0 -1]
C_transpose = C'
```

```
C =

     2     3    -2
     1     0    -1


C_transpose =

     2     1
     3     0
    -2    -1
```

Determinant of Square Matrices. The determinant of a matrix can be computed by using the function `det()`. This can be done either:

```
A = [1 2 3; 4 5 6; 7 8 -9]
det(A) % on a given matrix A
det([1 2 3; 4 5 6; 7 8 -9]) % or inline
```

```
A =

     1     2     3
     4     5     6
     7     8    -9


ans =

   54.0000


ans =

   54.0000
```

Task 10: Solve Problem #2b in Chapter 1.3 of Marsden and Tromba.

Task 11: Solve Problem #42 in Review Exercises for Chapter 1 of Marsden and Tromba. Hint: simplify the expression.

Calculus III

Combinations of Vectors. Here are various combinations of vectors.

```
u=[6,9,12];
v=[-1,0,3];
norm(u)
w=u+v
dot(u,v)
cross(u,v)
dot(u,cross(u,v))
dot(u,v)/dot(v,v)*v
```

```
ans =

   16.1555


w =

     5     9    15


ans =

    30
```

```
ans =

    27   -30     9


ans =

     0


ans =

    -3     0     9
```

Notice also here the semicolons at the end of the first two lines. These semicolons suppress the output, meaning we're telling MATLAB "assign the vectors and keep quiet about it". Also note the use of `norm` for the length (magnitude) of a vector. Note: Don't use the MATLAB command `length` because that just tells you how many elements are in the vector.

A few things to observe: The `cross` command produces a new vector, obviously. The second-to-last value is 0 and you should have expected that. Why? What is the last calculation finding? It's something familiar.

Here's the distance from the point $(1, 2, 3)$ to the line with parametric equations $x = 2t, y = -3t + 1, z = 5$. Make sure you see what's going on here. The point Q is off the line, the point P is on the line and the vector $\mathbf{L}$ points along the line. The reason P and Q are given as vectors is that we can then easily do $Q - P$ to get the vector from $\overrightarrow{PQ}$.

```
Q=[1 2 -3];
P=[0 1 5];
L=[2 -3 0];
norm(cross(Q-P,L))/norm(L)


ans =

    8.1193
```

Warning: There are certain things which behave differently in MATLAB than you might expect because MATLAB knows a bit more than you might about some things. For example the dot product of two vectors has a more flexible definition if the entries are complex numbers. In the above example this was never an issue since none of the quantities are complex, but if we try to use a variable look at what happens. Notice I've put `clear all` first which completely clears out MATLAB so we know we're starting anew.

```
clear all;
syms t;
a=[t,2*t,5];
dot(a,a)

ans =

5*t*conj(t) + 25
```

What happened? MATLAB doesn't know that `t` is a real number and so it does the more generic dot product which we are not familiar with, giving rise to the `cong(t)` term. If we want MATLAB to know that `t` is a real number we can tell it:

```
clear all;
syms t;
assume(t,'real');
a=[t,2*t,5];
dot(a,a)

ans =

5*t^2 + 25
```

That additional line `assume(t,'real')` tells MATLAB to treat `t` as a real number so that `conj` stuff (complex conjugate) won't bother us.

We can also find the norm of a symbolic vector. Here we'll redefine our variables for clarity and I'll simplify:

```
assume(t,'real');
a=[t,2*t,5];
simplify(norm(a))

ans =

5^(1/2)*(t^2 + 5)^(1/2)
```

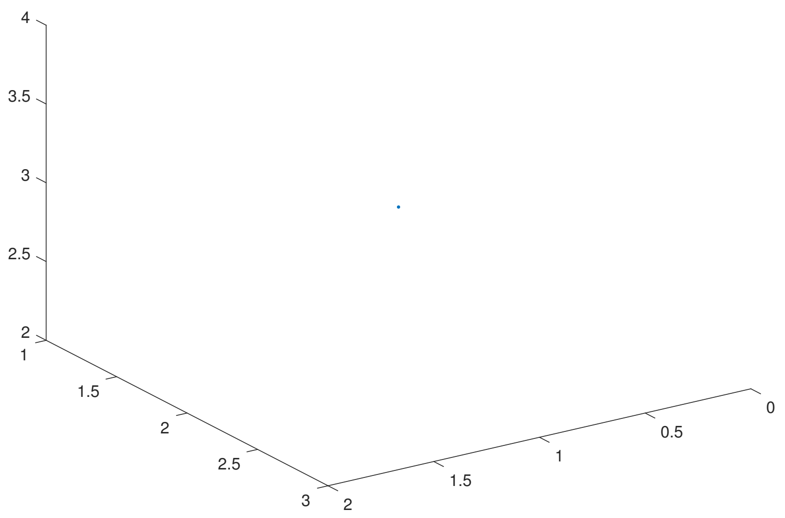
Task 12: Define four points $P = (2, -1, 3), Q = (0, 7, 9), R = (4, -9, -3)$ and $S = (7, -6, -6)$. With two subtractions and one dot product, all on one MATLAB line, show that the line through P and Q is perpendicular to the line through R and S.

Task 13: Define two points $P = (1, -2, 3)$ and $Q = (2, -1, 3)$ and one vector $\mathbf{n} = 2\mathbf{i}+2\mathbf{j}+3\mathbf{k}$. With one subtraction and one dot product, all on one MATLAB line, show that Q is not contained in the plane containing P and normal to $\mathbf{n}$.

Task 14: Define four points $P = (5, 0, 2), Q = (1, 1, 1), R = (0, 1, -2)$, and $S = (1, -2, -1)$. With five subtractions, two cross products and one dot product, all on one MATLAB line, find the distance from S to plane containing the other three points.

Plotting Points. Okay, let's plot some stuff in three dimensions! Here is how we can plot just one point, the point (1,2,3):

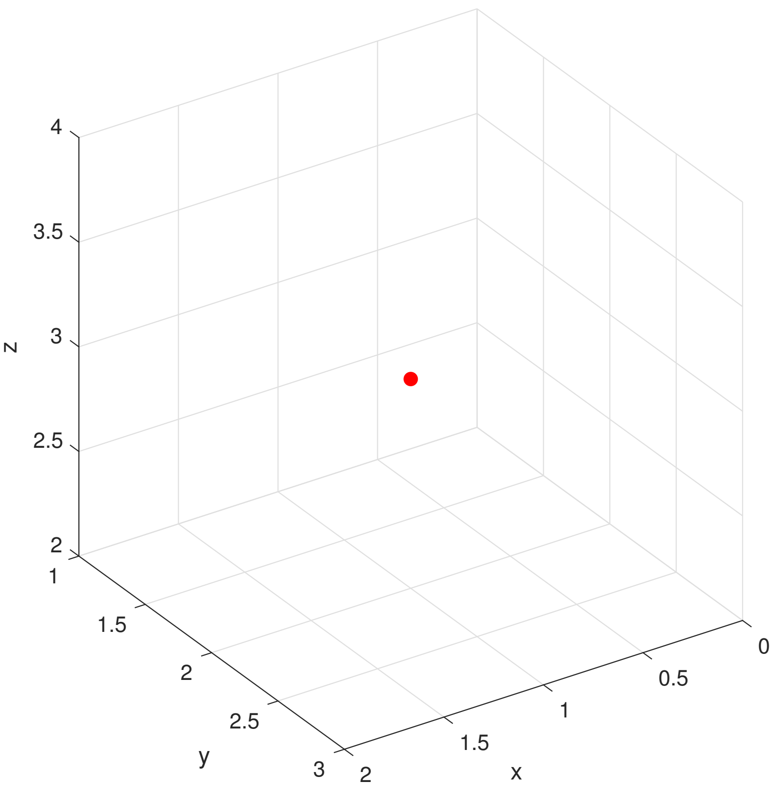
```
plot3(1,2,3,'.')
view([10 15 10])
```



To aid the visual, we could add grids, increase size of point (Marker), and label the axes:

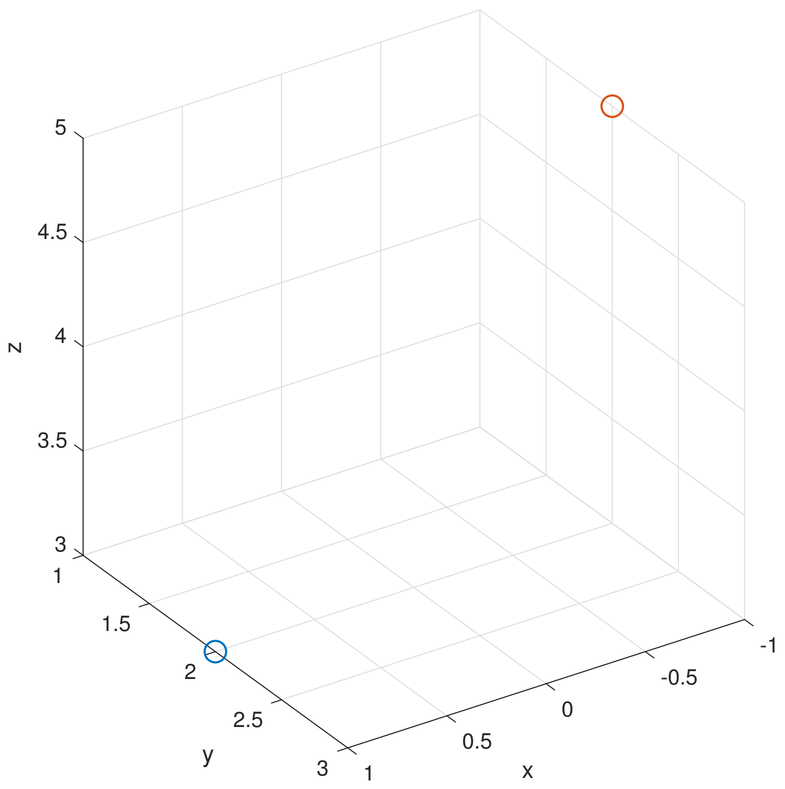
```
plot3(1,2,3,'.', 'MarkerSize',20, 'Color', 'r') % set point size and color
grid on
xlabel('x')
```

```
ylabel('y')
zlabel('z')
axis equal
view([10 15 10])
```



And here are two points

```
plot3(1,2,3,'o',-1,2,5,'o', 'MarkerSize',10)
grid on
xlabel('x')
ylabel('y')
zlabel('z')
axis equal
view([10 15 10])
```



Note the `'o'` tells MATLAB to draw circles for points. Also note the `view` command there. MATLAB has an annoying habit of viewing the axes from a nonstandard angle. The `view([10 15 10])` command puts our point-of-view at the point (10,15,10) so we look down at the origin as we're used to in class.

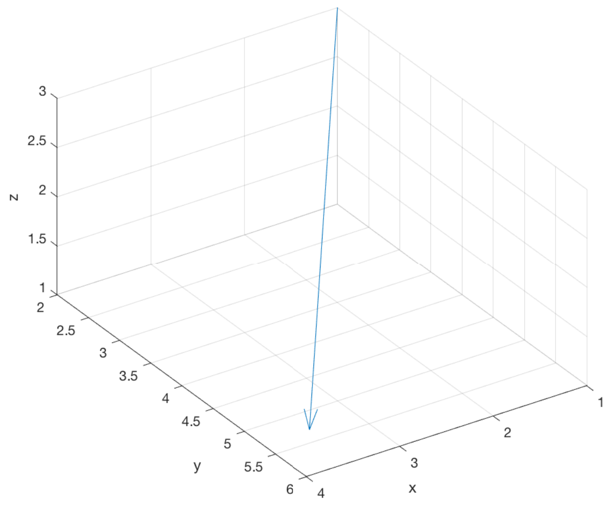
Important observation! On the figure window there is a little rotation button which allows you to drag the picture around and look at it from different angles. Very useful and fun too! I can't demonstrate in a fixed html document though.

Task 15: Plot the three points together $(1, 2, 3), (2, -3, 0)$, and $(-3, 5, 1)$.

Plotting Vectors. Here's a vector in 3D. The first three coordinates are the anchor point and the last three are the vector. So this is the vector $3\mathbf{i} + 4\mathbf{j} - 2\mathbf{k}$ anchored at $(1, 2, 3)$.

```
quiver3(1,2,3,3,4,-2)
xlabel('x')
ylabel('y')
zlabel('z')
```

```
view([10 15 10])
daspect([1 1 1])
```



This last command `daspect` sets the aspect ratio so that all the axes are scaled the same. I have no idea why MATLAB doesn't do this by default but grumble grumble grumble. This is more general than `axis equal` since we could input a different vector, say `[1 1 2]`, to change the ratios between the axes.

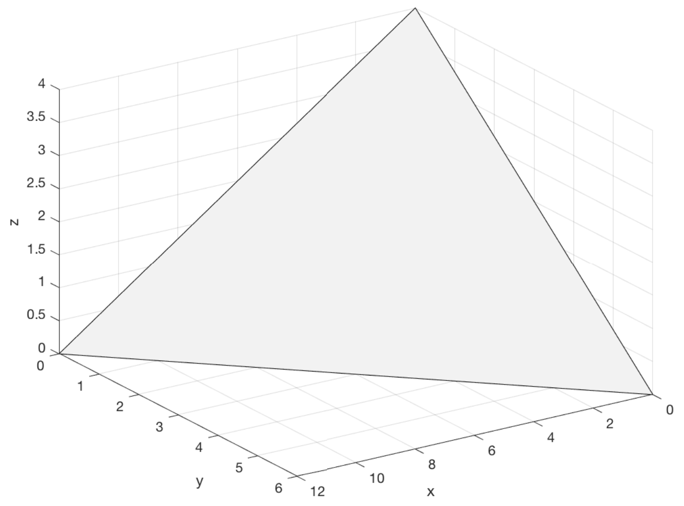
Task 16: Plot the vector $2\mathbf{i} - 3\mathbf{j} - \mathbf{k}$ anchored at (-1,4,3).

Plotting Planes. Just to close out let's draw some planes. MATLAB has a great command patch which draws a polygon. To use this to draw a plane we need to give it several points on the plane. It's a bit awkward - we need to give the points in order around the plane in either direction and we need to give it the `x`-coordinates together and the `y`-coordinates together. There's a slick way to do the second part but for the first part we just need to choose the points carefully, following around the part of the plane we want to plot. Here are some examples:

For the plane $x + 2y + 3z = 12$ we've seen that we can get a good picture using the intercepts $(12, 0, 0)$ and $(0, 6, 0)$ and $(0, 0, 4)$. The order we take them in doesn't matter since any order will either be counterclockwise or clockwise. To draw this with `patch` we do:

```
figure
points = [12 0 0;0 6 0;0 0 4];
patch(points(1,:),points(2,:),points(3,:),[0.95 0.95 0.95]);
grid on
```

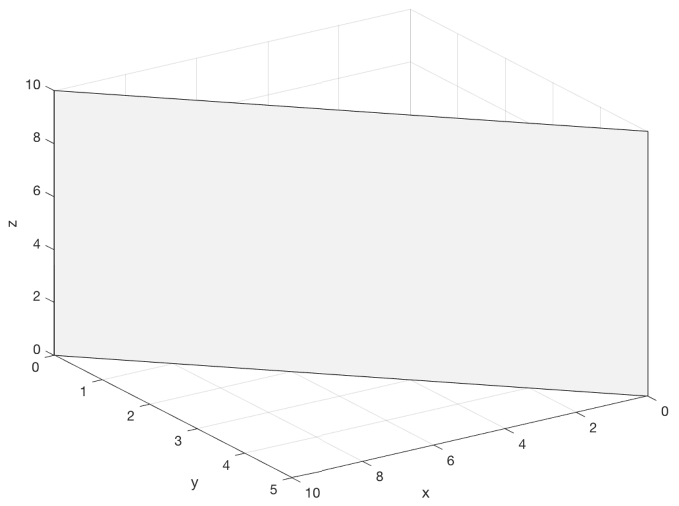
```
xlabel('x')
ylabel('y')
zlabel('z')
view([10 15 10])
```



Task 17: Plot the plane $2x + y + 4z = 16$.

For the plane $2x + 4y = 20$ we've seen that the way to approach this is to draw the line $2x + 4y = 20$ in the `xy`-plane and then extend upwards. The `x` and `y`-intercepts are $(10, 0, 0)$ and $(0, 5, 0)$ so we'll plot those along with $(10, 0, 10)$ and $(0, 5, 10)$. Note that the order we've chosen goes around the piece of the plane.

```
figure
points = [10 0 0;0 5 0;0 5 10;10 0 10];
patch(points(:,1),points(:,2),points(:,3),[0.95 0.95 0.95]);
grid on
xlabel('x')
ylabel('y')
zlabel('z')
view([10 15 10])
```

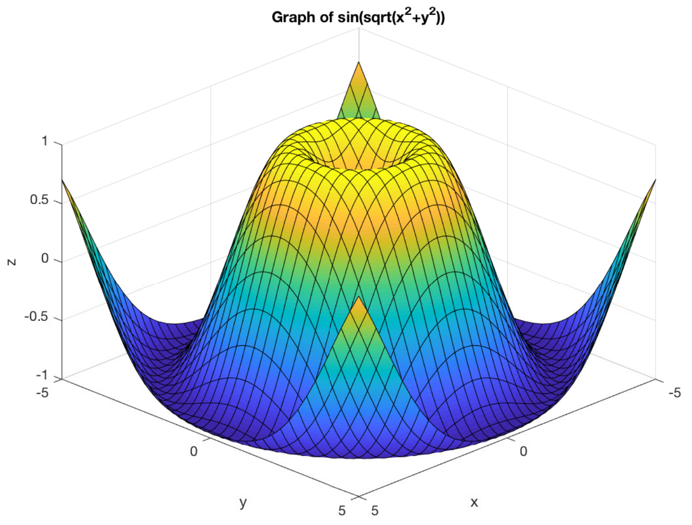


Task 18: Plot the plane $3y + 4z = 12$.

Task 19: Plot the plane $z = 3$.

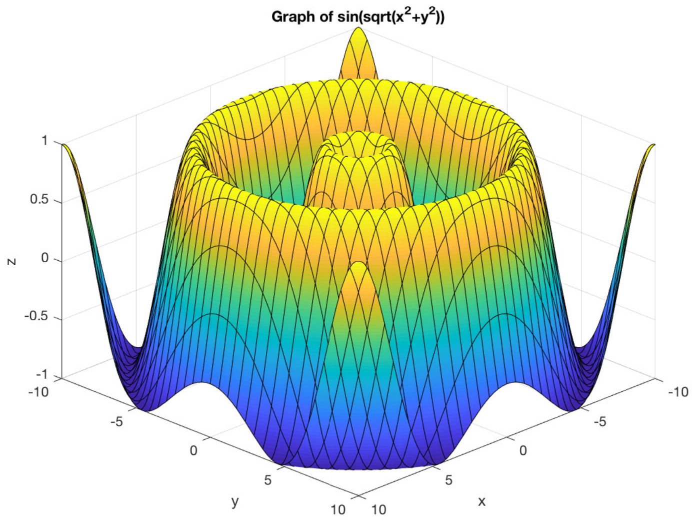
Plotting Graphs and Surfaces. In section 2.1 we learned how to graph functions of two variables. MATLAB can do this easily with the `fsurf` command. Here's an example of plotting the graph of $z = \sin(\sqrt{x^2 + y^2})$:

```
syms x y
fsurf(sin(sqrt(x^2+y^2)))
title('Graph of sin(sqrt(x^2+y^2))')
xlabel('x')
ylabel('y')
zlabel('z')
view([10 10 10])
```



If we wish to give it specific `x` and `y` like $-10 \le x \le 10$ and $-10 \le y \le 10$ we can do that too:

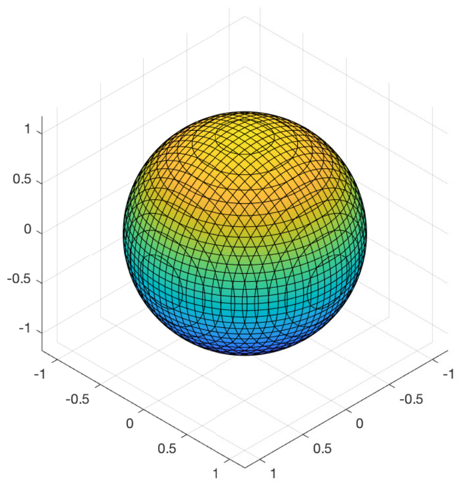
```
fsurf(sin(sqrt(x^2+y^2)), [-10 10 -10 10])
title('Graph of sin(sqrt(x^2+y^2))')
xlabel('x')
ylabel('y')
zlabel('z')
view([10 10 10])
```



Task 20: Plot the graph of $f(x, y) = y\sin(x) - x\cos(y)$ for x, y in $[-2\pi, 2\pi]$. Remember to label everything.

In the case when the surface at hand is not a graph, we need to use `fimplicit3` to plot the surface insteadd of `fsurf`. For instance, it is clear that $x^2+y^2+z^2 = 1$ is not a graph since it doesn't pass the vertical line test. So, to plot it, we do so as follow:

```
syms x y z
fimplicit3(x^2+y^2+z^2-1)
view([10 10 10])
daspect([1 1 1])
```



Task 21: Plot the surfaces of $z^2 - xy = 1$ and $z^2 - xy = -1$. How are they different?

Partial Derivatives. Since we can tell MATLAB which variable to take the derivative with respect to, finding partial derivatives is as easy as regular derivatives. Here's an example of a partial derivative with respect to `y`:

```
syms x y
diff(x^2*y+y^2/x,y)
```

```
ans =

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

Task 22: Find $\dfrac{\partial}{\partial y}\left[\dfrac{\sin(x^2y)}{y^2}\right]$.

MATLAB does have a gradient command but it gives numerical approximations, not what we want. Instead we want a specific manifestation of the MATLAB Jacobian command. Specifically for a function of two variables we do the following. What this really does is take a bunch of derivatives with respect to the letters given and product the output we desire. This is how we find ∇f for $f(x,y) = x^2y^3$:

```
jacobian(x^2*y^3, [x, y])
```

```
ans =

[ 2*x*y^3, 3*x^2*y^2]
```

Task 23: Find the gradient of $f(x,y) = \dfrac{xy}{x^2+y}$.

Here's the same gradient with values plugged in for `x` and `y`. Note the use of the familiar `subs` command but with {`x,y`} to substitute for both values:

```
subs(jacobian(x^2*y^3,[x y]),{x,y},{-1,3})
```

```
ans =

[ -54, 27]
```

Task 24: Find the gradient of $f(x,y,z) = \dfrac{xe^{xy}}{z}$ at $(x,y,z) = (1,2,3)$.

The `jacobian` function does more than just finding gradient vector of scalar-valued functions. In fact, `jacobian` can compute the jacobian matrix for a general transformation. Consider the following example:

```
syms x y z
F = [x^3*y*z^2; cos(x^2+y^2); sin(z^2-x*y)]
jacobian(F, [x, y, z])
```

```
F =

         x^3*y*z^2
    cos(x^2 + y^2)
```

```
 -sin(- z^2 + x*y)


ans =

[        3*x^2*y*z^2,             x^3*z^2,           2*x^3*y*z]
[ -2*x*sin(x^2 + y^2), -2*y*sin(x^2 + y^2),                   0]
[ -y*cos(- z^2 + x*y), -x*cos(- z^2 + x*y), 2*z*cos(- z^2 + x*y)]
```

Task 25: Solve Problem #8 in Review Exercises for Chapter 2 of Marsden and Tromba.