Math 10B
Mathematics and 3D Printing
Lecture Notes

David Bachman
1. Points and vectors

A point is a location in space. A vector is an arrow. Mathematically, there is absolutely no difference between the two: both are determined by a set of numbers. For example, \((3, 4)\) is a point, while \(\langle 3, 4 \rangle\) is a vector. The difference is only notational.

You locate points by starting at the origin, and moving parallel to the axes the prescribed amounts. Vectors are a little more ambiguous. For example, to draw the vector \(\langle 3, 4 \rangle\), you start anywhere, go right 3 units, and then go up 4 units. Then draw an arrow whose base is your starting point, and whose tip is where you ended up. Moving this arrow parallel to itself does not change the fact that it represents \(\langle 3, 4 \rangle\). Thus, the point \((3, 4)\) is at the tip of the vector \(\langle 3, 4 \rangle\), only when that vector is drawn with its base at the origin.

The biggest difference between points and vectors, besides how we draw them, is that points are considered geometric objects, while vectors are algebraic objects. In other words, we can add, subtract, and multiply vectors, while we only draw pictures of points.

In this class, there are three main algebraic operations we will do to vectors. Suppose \(V = \langle a, b \rangle\).

1. **Negation.** \(-V\) is the vector \((-a, -b)\). An arrow representing the vector \(-V\) has the exact same length as \(V\), but points in the opposite direction.

2. **Rescaling.** If \(t\) is some positive number, then \(tV\) is the vector \(\langle ta, tb \rangle\). An arrow representing \(tV\) points in the same direction as \(V\), but is \(t\) times as long. (When \(t\) is negative, then we think of \(tV\) as a combination of negation and rescaling.)

3. **Addition.** If \(W = \langle c, d \rangle\) then \(V + W\) is the vector \(\langle a + c, b + d \rangle\). To get an arrow that represents \(V + W\), place the base of \(W\) at the tip of \(V\). Then draw a new arrow whose base is that of \(V\), and whose tip is that of \(W\).

---

**Activities**

*Question 1.* \(V - W\) is defined to be \(V + (-W)\). What is \(V - W\) in terms of the numbers \(a, b, c,\) and \(d\)? How would you draw \(V - W\) in relation to \(V\) and \(W\)?
Question 2.

(1) Use Grapher to plot the corners of a square with side length two that has one corner at the origin.
(2) Add the vector \( \langle 2, 3 \rangle \) to each corner of the square and plot. What happened to the original square?
(3) Negate each corner of the original square and plot.
(4) Multiply each corner of the original square by 3 and plot.

Homework 1

(1) Use Grapher to recreate the above picture.
(2) Add the vector \( \langle 2, 3 \rangle \) to each point in the picture and plot.
(3) Negate each point in the picture and plot.
(4) Multiply each point of the picture by 3 and plot.

After each step, use the “export” command to get a jpeg image from Grapher. Then send all four jpegs as an attachment to bachman@pitzer.edu, with the subject line MATH 10B HOMEWORK 1.
2. Curves in two dimensions: lines

In the previous section we saw that when \( V \) is a vector, then \( tV \) points in the same direction, but is a different size. Hence, placing the base of \( tV \) at the origin and varying \( t \) results in points on a line through the origin, containing \( V \).

To move this line so that it starts at another point \( p \), we simply add a vector \( W \) to \( tV \) that points from the origin to \( p \).

Question 3.

(1) Use Grapher to plot \( tV \), where \( V = \langle 2, 3 \rangle \) and \( t \) ranges from 0 to 1.
(2) Now let \( W = \langle 1, -1 \rangle \) and plot \( W + tV \), for \( t \) between 0 and 1.
(3) What are the coordinates of the endpoints of the resulting line segment? What is the difference between them?
(4) Rewrite \( W + tV \) as a single vector whose coordinates are functions of \( t \). Confirm that plotting this with Grapher gives the same result.
(5) Use Grapher to construct a line segment from \((0, 1)\) to \((4, 0)\).

A vector whose coordinates are functions of a variable (usually \( t \)) is called a parameterization of the curve you get when you graph it. After the previous activity, you should be able to write a parameterization for any line segment. Another place where parameterizations arise is from graphs of functions. If \( f(x) \) is a function, then recall from high school that its graph are all of the point \((x, y)\) such that \( y = f(x) \). A parameterization for this graph is then given by the vector \( \langle t, f(t) \rangle \).

Question 4. Use Grapher to plot the graphs of the following functions, by writing them as parameterized curves.

(1) \( x^2 \)
(2) \( x^3 \)
(3) \( \ln x \)
(4) \( e^x \)
(5) \( e^{-x^2} \)

Homework 2

Recreate each of the following pictures.
3. Circles

Before proceeding, we must do a little trig review. Suppose $A$, $O$, and $H$ are the sides of a right triangle, with $H$ the hypotenuse. Let $a$, $o$, and $h$ be their lengths, and $\theta$ the (radian) measure of the angle between sides $A$ and $H$. Then $\cos \theta$ is defined to be the ratio $\frac{a}{h}$ and $\sin \theta$ is defined to be $\frac{o}{h}$.

This definition of $\cos \theta$ and $\sin \theta$ works great for right triangles, where $0 < \theta < \frac{\pi}{2}$, but not so well when $\theta$ is outside this range. For this we need a different interpretation.

Let $C$ be a circle of radius 1 in the plane, centered on the origin. Suppose $p = (x, y)$ is a point on this circle in the positive quadrant, and $\theta$ is the angle a ray from the origin to $p$ makes with the positive $x$-axis. Then dropping a vertical segment from $p$ to the $x$-axis creates a right triangle. The hypotenuse of this triangle is just the radius of the circle, which we have assumed is 1. The side adjacent to $\theta$ is $x$ and the third side is $y$. Thus, $\cos \theta = \frac{x}{1} = x$ and $\sin \theta = \frac{y}{1} = y$. We conclude that the coordinates of $p$ are $(x, y) = (\cos \theta, \sin \theta)$.

This interpretation allows us to extend the definition of $\cos \theta$ and $\sin \theta$ to values of $\theta$ greater than $\theta = \frac{\pi}{2}$ and less than 0. By definition, $\cos \theta$ and $\sin \theta$ are the $x$ and $y$ coordinates of the point $p$ on the unit circle that is on the ray from the origin to $p$ making an angle of $\theta$ with the positive $x$-axis.

---

Activities

*Question 5.* Where should the following points be? Plot them and check your guess.

1. $(\cos 0, \sin 0)$.
2. $(\cos \frac{2\pi}{3}, \sin \frac{2\pi}{3})$
3. $(\cos \frac{4\pi}{3}, \sin \frac{4\pi}{3})$

Use these points to construct a triangle.

*Question 6.* Where should $(2 \cos \frac{2\pi}{3}, 2 \sin \frac{2\pi}{3})$ be? Plot it and check.

*Question 7.* Guess what each of the following look like, and then check yourself.

1. $\begin{bmatrix} \cos t \\ \sin t \end{bmatrix}, t = 0..\pi$
(2) \[
\begin{bmatrix}
\cos t \\
\sin t
\end{bmatrix}, t = 0..2\pi
\]
(3) \[
2 \begin{bmatrix}
\cos t \\
\sin t
\end{bmatrix}, t = 0..2\pi
\]
(4) \[
\begin{bmatrix}
1 \\
1
\end{bmatrix} + \begin{bmatrix}
\cos t \\
\sin t
\end{bmatrix}, t = 0..2\pi
\]

**Question 8.** Plot a circle with center at \((-1, 1)\) and radius 3.

---

**Homework 3**

Recreate the following picture:
4. Distances and Midpoints

Here are two useful formulas:

1. The distance from a point \((x, y)\) to the origin is found from the Pythagorean Theorem:

\[
d = \sqrt{x^2 + y^2}
\]

In general, the distance between \((x_1, y_1)\) and \((x_2, y_2)\) is

\[
d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

2. The midpoint between \((x_1, y_1)\) and \((x_2, y_2)\) is

\[
\left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)
\]

**Activities**

1. Draw an equilateral triangle, \(T\).
2. Add the smallest circle that contains \(T\).
3. Add a new triangle whose vertices are at the midpoints of the edges of \(T\).
4. Add the largest circle that can be drawn inside \(T\).
Homework 4
Recreate the following picture:
5. Basis Vectors

One way to think about the vector $\langle a, b \rangle$ is as the sum $a\langle 1, 0 \rangle + b\langle 0, 1 \rangle$. The two vectors $e_1 = \langle 1, 0 \rangle$ and $e_2 = \langle 0, 1 \rangle$ are called the standard basis for the plane. Rotating these vectors by an angle of $\theta$ (counter-clockwise) results in a different basis, given by the pair of vectors $V_\theta = \langle \cos \theta, \sin \theta \rangle$ and $W_\theta = \langle -\sin \theta, \cos \theta \rangle$.

The upshot here is a way to rotate points, and thus entire objects. If $p = \langle a, b \rangle = ae_1 + be_2$, then the result of rotating $p$ by an angle $\theta$ around the origin is $aV_\theta + bW_\theta$.

### Activities

(1) Define vectors $V_{\frac{\pi}{6}}$ and $W_{\frac{\pi}{6}}$.

(2) Use $V_{\frac{\pi}{6}}$ and $W_{\frac{\pi}{6}}$ to rotate the points $(1,1)$, $(2,1)$, $(1,2)$, and $(2,2)$ by $\frac{\pi}{6}$ around the origin.

(3) Connect all four points to make a square.

(4) Translate this square by the vector $\langle 1, 2 \rangle$.

### Homework 5

Follow these steps:

(1) Define vectors $V(a) = \begin{bmatrix} \cos a \\ \sin a \end{bmatrix}$ and $W(a) = \begin{bmatrix} -\sin a \\ \cos a \end{bmatrix}$.

(2) Define points $p(a) = V(a)$, $q(a) = 2(\cos \frac{\pi}{8})V(a) + 2(\sin \frac{\pi}{8})W(a)$ and $s(a) = (\cos \frac{\pi}{4})V(a) + (\sin \frac{\pi}{4})W(a)$.

(3) Draw a line from $p\left( \frac{\pi n}{4} \right)$ to $q\left( \frac{\pi n}{4} \right)$, for $n = \{0, 1, 2, 3, 4, 5, 6, 7\}$.

(4) Draw a line from $q\left( \frac{\pi n}{4} \right)$ to $s\left( \frac{\pi n}{4} \right)$, for $n = \{0, 1, 2, 3, 4, 5, 6, 7\}$.

(5) Email me the resulting picture.

(6) Modify this to create a 7 pointed star, and email me its picture.
6. Final 2D curve assignment

Reconstruct the following picture of a gear. Note that it has 20 teeth, and each tooth is made up of three sides of a rectangle.
7. Three Dimensions

The concepts you have learned for points, lines, and circles change very little in three dimensions.

(1) Draw a circle of radius 1 in the \(xz\)-plane.
(2) Move your circle two units in the \(y\)-direction, and one unit in the \(z\)-direction.
(3) Construct the edges of a cube.

Homework 7

Reconstruct the following picture of a pentagonal cylinder of height 2, centered at the origin, around the \(x\)-axis:

As an added challenge, try to do this with just 3 lines in *Grapher*!
8. Practice

Construct the following objects:

(1) A hexagonal pyramid (construct a hexagon in the \(xy\)-plane, and then connect the vertices to the point \((0, 0, 1)\).)

(2) An octahedron (six vertices at \((\pm 1, 0, 0), (0, \pm 1, 0)\) and \((0, 0, \pm 1)\)).

(3) A horizontal circle of radius 1, at heights \(\{0, 1, 2, 3, 4\}\). Add six evenly spaced vertical lines connecting the lowest circle to the highest one, to form something that looks like a cylinder.

---

Homework 8

Reconstruct the following picture of a drum. The top and bottom are pentagons. The upper one is rotated with respect to the lower one. Each vertex in the top pentagon connects to two vertices in the bottom one. How many lines in *Grapher* does this take you?

Draw horizontal circles whose radii equal their height, at heights \(\{0, 1, 2, 3, 4\}\). On the same picture, add lines that connect \((0, 0, 0)\) to 8 evenly spaced points on the top circle. Picture is below.
9. Basis Vectors for $\mathbb{R}^3$.

Recall that in two dimensions we could rotate the point $(2, 5)$ around the origin by first writing it as $2e_1 + 5e_2$, and then converting to a rotated basis by writing $2V(a) + 5W(a)$, where $V(a) = \begin{bmatrix} \cos a \\ \sin a \end{bmatrix}$ and $W(a) = \begin{bmatrix} -\sin a \\ \cos a \end{bmatrix}$. Altering the angle $a$ then rotates the point by that much. (So, for example, if $a = \frac{n2\pi}{8}$ and $n = \{0, 1, 2, 3, 4, 5, 6\}$ then we will get 8 copies of $(2, 5)$, arranged around the origin.)

We now take this idea one step further. The basis $V(a) = \begin{bmatrix} \cos a \\ \sin a \\ 0 \end{bmatrix}$ and $W = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ describe a plane $P(a)$ through the $z$-axis in $\mathbb{R}^3$, at an angle $a$ from the $x$-axis.

---

Activities

(1) Draw the point $(2, 5)$ in the plane $P(\pi/4)$ by plotting $2V(\pi/4) + 5W$.

(2) Plot the points $2V(n2\pi/8) + 5W$, where $n = \{0, 1, 2, 3, 4, 5, 6\}$. Where are they?

(3) Create a square with vertices at $(1, 0)$, $(2, 0)$, $(2, 1)$, and $(1, 1)$, in the $\{V(a), W\}$ basis, where $a = \pi/4$.

(4) Replace $a$ with $n2\pi/8$, where $n = \{0, 1, 2, 3, 4, 5, 6\}$.
Homework 9
Reconstruct the following picture of seven hexagons.
10. A little slower this time...

The homework assignment from the previous section was a bit of a challenge. Let’s take it slower...

1. Start *Grapher* in 2D mode.
2. Define a basis at an angle \( a \) around the origin, \( V(a) = (\cos a, \sin a) \), \( W(a) = (-\sin a, \cos a) \).
3. Define the points of an equilateral triangle.
4. Translate your definition of these points into the rotating basis. Your points will look like \( p(a, n) \), where \( a \) is the angle you’ll use to define the basis, and \( n \) tells you which point of the triangle you are referring to.
5. For each angle \( a \) that is a multiple of \( \frac{2\pi}{8} \), connect the points to make 8 triangles.
6. Now open a separate window in *Grapher* in 3D mode.
7. Follow the same steps as above, but use the basis \( V(a) = (\cos a, \sin a, 0) \), \( W = (0, 0, 1) \).
8. Now modify your 2D and 3D code so that the original triangle is centered on \((2, 0)\).
Homework 10

Reconstruct the following picture of seven hexagons. (Hint: First think about how you would construct a hexagon in 2-dimensions, centered around the point (2,0).)
Squares!! The following python programs can be copied and pasted into TextWrangler, and then saved on your machine (with .py extension). Then open Rhino3D and type “runpythonscript.” When the dialog box pops up, navigate to where you saved the code.


```python
import rhinoscriptsyntax as rs

rs.AddLine([1,0,0],[0,1,0])
rs.AddLine([0,1,0],[-1,0,0])
rs.AddLine([-1,0,0],[0,-1,0])
rs.AddLine([0,-1,0],[1,0,0])
```


```python
import rhinoscriptsyntax as rs

pts=[]
pts.append([1,0,0])
pts.append([0,1,0])
pts.append([-1,0,0])
pts.append([0,-1,0])
pts.append([1,0,0])

rs.AddLine(pts[0],pts[1])
rs.AddLine(pts[1],pts[2])
rs.AddLine(pts[2],pts[3])
rs.AddLine(pts[3],pts[4])
```

```python
import rhinoscriptsyntax as rs
pts=[]
pts.append([1,0,0])
pts.append([0,1,0])
pts.append([-1,0,0])
pts.append([0,-1,0])
pts.append([1,0,0])
for n in range(4):
    rs.AddLine(pts[n],pts[n+1])
```


```python
import rhinoscriptsyntax as rs
import math
pi=math.pi
pts=[]
for n in range(5):
    angle=2*pi*n/4
    x=math.cos(angle)
    y=math.sin(angle)
    z=0
    pts.append([x,y,z])
for n in range(4):
    rs.AddLine(pts[n],pts[n+1])
```
import rhinoscriptsyntax as rs
import math

pi = math.pi
m = 4

pts = []
for n in range(m + 1):
    angle = 2 * pi * n / m
    x = math.cos(angle)
    y = math.sin(angle)
    z = 0
    pts.append([x, y, z])

for n in range(m):
    rs.AddLine(pts[n], pts[n + 1])

---

**Homework 11**

1. Modify the Version 5 code to draw a 10 sided polygon, centered at (2, 0).
2. Draw a triangle with vertices at (0, 0), (2, 0) and (1, 1).

In both case, take a screen shot (shift+command+4) and email me a picture.
12. More coding...

(1) Modify the Version 5 code from the previous section by adding a single line of code in the second loop that creates a line between the point “pts[n]” and the point [0, 0, 1]. The result should be a cone built on a polygon. Run the code with different values of $m$ (e.g. $m = 3, 8, 20$).

(2) Modify the Version 5 code from the previous section as follows:
   (a) Change the list “pts” to “bottom_pts”
   (b) Create a similar list of points called “top_pts”, and change $z = 0$ to $z = 1$.
   (c) In the loop that creates the edges, you should now have three lines of code. In the first line, a $m$-sided polygon will be drawn in the plane $z = 0$ by using the bottom_pts. In the second line, use top_pts to draw a polygon in the plane $z = 1$. In the third line, create vertical edges connecting the points of the bottom polygon to the points of the top one.
   (d) Run the code with different values of $m$ (e.g. $m = 3, 8, 20$).

(3) Make the “drum” shaped object from Homework 8 with python and rhino.
13. Surfaces!

In this section we introduce a new command to make surfaces:

\[ \text{rs.AddEdgeSrf(curve \_ ids)} \]

This creates a triangle or a quadrilateral. The identifier \text{“curve \_ ids”} is a list of 3 or four curves that together form a loop.

---

**Activities**

1. In the Version 5 code of the square-maker (with \( m = 4 \) as in the original code), replace the second “for” loop with the following code:

   \[
   \text{edges} = []
   \text{for n in range(m):}
   \text{edges.append(rs.AddLine(pts[n],pts[n+1]))}
   \text{rs.AddEdgeSrf(edges)}
   \]

2. Re-run the same script with \( m = 3 \) and then \( m = 5 \).

3. Make a pentagon out of 5 triangles, each having one corner at the origin.

4. Make a cone on a pentagon by connecting all the vertices to \((0,0,1)\) and “filling in” all the faces.

---

**Homework 13**

Make a cube with solid faces.

The way we built objects previously followed the logic of Gapher: Define points, connect pairs of points with lines, and then fill in the resulting polygons. In many cases other rhinoscript commands will accomplish the same results with less coding.

The command

\[
\text{rs.AddPolyline}(\text{list-of-points})
\]

creates a continuous polygonal curve that follows a set of points. To see how this differs from what we’ve already done, first run the (original) Version 5 square-maker code. Then point to a side of the resulting square with the mouse, and left-click. What do you see?

Now, replace the entire second “for” loop with the following command:

\[
\text{rs.AddPolyline}(pts)
\]

You should again see a square. Left-click on it. What happens this time?

The command

\[
\text{rs.AddPlanarSrf}(\text{polygonal-line})
\]

fills in a face with a prescribed boundary.

In your modified code, replace \text{rs.AddPolyline}(pts) with:

\[
\text{polygon} = \text{rs.AddPolyline}(pts) \\
\text{rs.AddPlanarSrf}(\text{polygon})
\]

Re-run your code. What happens? Now change \(m\) to 5 and run the code again. What happens?

Each command has its advantages and disadvantages. The \text{AddPlanarSrf} command is nice, because it will create a polygon with any number of sides. However, all of these sides must lie in the same plane. In contrast, the \text{AddEdgeSrf} command will only create a shape with three or four sides, but they don’t have to be planar. Which one is best will depend on your application.

---

Homework 14

Create a prism with a hexagonal top and bottom, connected by six vertical sides. Use the \text{AddPlanarSrf} command to create each side.
15. Basic vector operations in rhinoscript.

If you wanted to double the size of a vector $V$ in *Grapher*, you’d just type $2V$. Unfortunately, things are more difficult in rhinoscript. Scaling a vector is done by the command *VectorScale*. Here is the syntax:

```python
rs.VectorScale(vector, scale_factor)
```

Activity: Generate a list of 5 points called *pts*, equally spaced around the unit circle. Then add the following code:

```python
for n in range(5):
    rs.AddLine(rs.VectorScale(pts[n], 2), rs.VectorScale(pts[n+1], 2))
```

Adding and subtracting vectors is, unfortunately, equally awkward. Here is the syntax:

```python
rs.VectorAdd(vector1, vector2)
rs.VectorSubtract(vector1, vector2)
```

In the former case the result is the vector $\text{vector1} + \text{vector2}$. In the latter case the result is $\text{vector1} - \text{vector2}$.

Activity: Create a pentagon in the $xy$-plane. Then use the *VectorAdd* command to move it up, in the $z$-direction, 2 units.

---

**Homework 15**

Use rhinoscript, python, and rhino to make one of the hexagons from Homework 10, in the vertical plane that is at an angle of $\frac{\pi}{3}$ from the $x$-axis.
16. Continued....

Use rhinoscript, python, and rhino to make all of the hexagons from Homework 10.
17. Lists of lists

Activities

Use rhinoscript, python, and rhino to connect the vertices of consecutive hexagons from the previous assignment.

Python has the capability of making lists of lists. This is useful to make a grid of points on a surface. To construct such a thing, follow this template:

```python
points=[]
for m in range(m-count):
    points.append([])
    for n in range(n-count):
        ...
    points[m].append(n-th element of m-th list)
```

After this nested loop, you can retrieve the n-th element of the m-th list by writing `points[m][n]`.

Homework 17

Put the points of the seven hexagons into a list-of-lists. In a second nested loop, use these points to make rectangular surfaces that fill in all the cells, forming a solid object. (The `AddEdgeSrf` command seems to work best for this.)
18. Playing with the previous model

Modify the shape from the previous assignment so it is defined by 9 squares around the $z$-axis, with sides parallel to the $z$-axis and the $xy$-plane.

Homework 18

Put a $1/4$ twist in the shape you made in class, to obtain a picture like the following:
Define a torus shape by 50 different 30-sided regular polygons around the $z$-axis. Then modify your previous design so the polygons get bigger and bigger as you go around the $z$-axis. Your model should look like a snake eating its tail, as in the following picture:
Homework 19

Make the polygons get bigger and then smaller, so it looks like a torus with a bulge in one side:
20. Curve Thickening Part I: a little vector arithmetic

Very often we’ll want to 3D-print curves and surfaces. Unfortunately, only objects with some thickness can be printed, so in both cases you have to do something to make a printable object. A useful algebraic tool when doing this is the cross product. This is an operation that takes two vectors, and returns a third that is perpendicular to both.

Suppose \( V = \langle a, b, c \rangle \) and \( W = \langle d, e, f \rangle \). Then the cross product of \( V \) and \( W \) is given by

\[
V \times W = \langle bf - ce, cd - af, ae - bd \rangle
\]

Compute, by hand, \( \langle 1, 0, 1 \rangle \times \langle 0, 1, 2 \rangle \). What happens if you switch their order?

Fortunately, there is a rhinoscript command to compute cross products:

\[
\text{rs.VectorCrossProduct}(V, W)
\]

Unfortunately, the size of the resulting cross product may be very large, or very small. Often, we’d like a vector that points in this direction that has length one. A vector of length one is a unit vector, and the process of rescaling an arbitrary vector to a unit vector is called unitizing. Mathematically, the vector \( \langle a, b, c \rangle \) is unitized by computing the following:

\[
\left\langle \frac{a}{\sqrt{a^2 + b^2 + c^2}}, \frac{b}{\sqrt{a^2 + b^2 + c^2}}, \frac{c}{\sqrt{a^2 + b^2 + c^2}} \right\rangle
\]

Unitize, by hand, the vector \( \langle 1, 1, 2 \rangle \).

To unitize a vector \( V \) in rhinoscript, use the following command:

\[
\text{rs.VectorUnitize}(V)
\]

Re-scaling this vector by various amounts produces points on a line.

---

**Activities**

Write a python script that will have rhino draw a line segment of length 5 that is perpendicular to both \( \langle 1, 0, 1 \rangle \) and \( \langle 0, 1, 2 \rangle \).
Homework 20

Write a python script that will have rhino draw a hexagon in the plane that contains \( \langle 1, 0, 1 \rangle \) and is perpendicular to \( \langle 0, 1, 2 \rangle \).
21. Curve Thickening, Part II

Let \( p \), \( q \), and \( r \) be three points. If we connect these up sequentially, we get two line segments meeting at \( q \). Our goal now is to create a polygon centered at \( q \) that is roughly perpendicular to both line segments.

1. Let \( V = p - q \) and \( W = r - q \). Draw these vectors, based at \( q \).
2. Draw the vector \( A \) that is the average of \( V \) and \( W \) (i.e. \( A = \frac{1}{2}(V + W) \)).
3. Draw \( V \times A \).
4. Draw the plane containing \( A \) and \( V \times A \).
5. Draw a hexagon in this plane, centered at \( q \).

Homework 21

Write a python script to draw the hexagon described above, where \( p = (1, 0, 0) \), \( q = (2, 1, 0) \), and \( r = (5, 2, 0) \). (Don’t forget to unitize \( A \) and \( V \times A \).)
22. Curve Thickening, Part III

Write the following python program:

1) Create a list of points called “\texttt{pts}” of the form \((t, \sin t, 0)\), where \(t = 0, 1, 2, \ldots, 10\).

2) Write a for-loop that executes 9 times, that does the following on the \textit{n}th iteration:
   - Define \(p = \texttt{pts}[n]\), \(q = \texttt{pts}[n + 1]\), \(r = \texttt{pts}[n + 2]\).
   - Draw the hexagon described in the previous lesson, using \(p\), \(q\), and \(r\).


## 23. 3D printing project

### Instructions

1. Choose one item from what we’ve done in class so far. You may vary this item as much as you’d like to make something unique. (For example, instead of a torus with square cross section and half of a twist, you can have a pentagonal cross section and two twists.)

2. Export your model from rhino as an stl file, and import this file in the Makerbot slicing software.

3. Make sure your model is at most 3 inches (76 mm). Rescaling can be done in python, rhino, or the makerbot software.

4. Obtain an SD card.

5. Show your model, as displayed in Makerbot, to Dr. Bachman. (Together we’ll set it up to create a printable object.)

6. Export your printable file to your SD card, and give it to Dr. Bachman. Make sure your name is on it!

### Due dates

- Model choice: Monday, 4/6.
- Presentation of model in Makerbot: in class on Wednesday, 4/8.
24. Curve Thickening, Part IV

Continue with the hexagons you created around the curve \((t, \sin t, 0)\). Now connect these hexagons with rectangles, forming a hexagonal tube. Finally, cap off the ends to create a potentially printable object.

Homework 24
Create a tube with square cross section around the ellipse
\[
(2 \cos t, \sin t, 0), \quad 0 \leq t \leq 2\pi
\]
Homework 25

Thicken the following curves with square cross-sections, and then again with 30-sided cross-sections.

1. \([3 \cos t, 3 \sin t, 3t], 0 \leq t \leq 6\pi\) (Don’t forget to cap the ends.)
2. \([3 \cos 3t, 3 \cos(5t + \pi/2), 3 \cos(2t + \pi/6)], 0 \leq t \leq 2\pi\)
26. Torus Knots

A \((p,q)\)-torus knot is a thickened curve on a torus. The curve is characterized by \(p\) points, evenly spaced around a meridian, making a \(q/p\) rotation as they are swept around the longitude. Write code to create any \((p,q)\) torus knot. Include a parameter \(r\) in your code which determines the thickness of the knot.

Homework 26
Email a screen shot of the \((5,7)\)-torus knot.
27. Parameterized Surfaces

In mathematics, we often define surfaces in 3-dimensions by functions called *parameterizations*. A parameterization takes, as it's input, two variables, and returns three variables. For example, the following are parameterizations:

1. \( f(x, y) = (x, y, x^2 + y^2), -2 \leq x \leq 2, -2 \leq y \leq 2. \)
2. \( f(r, \theta) = (r \cos \theta, r \sin \theta, r^2), 0 \leq r \leq 2, 0 \leq \theta \leq 2\pi. \)

We display parameterized surfaces by defining a grid of points, and creating quadrilaterals with corners on the grid points. Such a grid is created with a list of lists, as before. Once you have the grid, you create the quads just as we did earlier.

Homework 27

Render the two parameterizations above. You may have to play with the number of grid points in each direction to get nice pictures. Start with 10.
28. More Surfaces

Try some of the following surfaces.

(1) \((x, y, x^2 - y^2)\)
(2) \((x, y, \sqrt{x^2 + y^2})\)
(3) \((r \cos \theta, r \sin \theta, r)\)
(4) \((r \cos \theta, r \sin \theta, r^2)\)
(5) \((r \cos \theta, r \sin \theta, \sqrt{r})\)
(6) \((r \cos \theta, r \sin \theta, \frac{1}{r})\)
(7) \((r \cos \theta, r \sin \theta, \theta)\)
(8) \((r \cos \theta, r \sin \theta, \cos r)\)
(9) \((\theta \cos \phi \cos \theta, \theta \sin \phi \sin \theta, \theta \cos \phi)\)
29. Thickening Surfaces

You have already seen how to create a list-of-list of points that defined a parameterized surface. Suppose $\text{pts}$ is such a list. To add some thickness to the surface, you have to define a new list-of-list of points that represents a parallel surface. To do this, follow these steps:

1. Define $V$ to be a vector from $\text{pts}[n][m]$ to $\text{pts}[n+1][m]$.
2. Define $W$ to be a vector from $\text{pts}[n][m]$ to $\text{pts}[n][m+1]$.
3. Compute the cross product of $V$ and $W$.
4. Unitize the result of the previous step.
5. Rescale the resulting vector to whatever distance you want the new surface to be from the old one.
6. Add the result of the previous step to the point $\text{pts}[n][m]$.
7. Define $\text{newpts}[n][m]$ to be the result of the last step.
8. After looping through all values of $n$ and $m$, create a surface using $\text{newpts}$.

Homework 29

Display the surface $(x, y, x^2 - y^2)$ and a parallel one that is 0.1 units away.
30. Final Project

(1) Choose any of the parameterizations from Homework 27 or 28.

(2) Create a 3D-printable surface. Don’t forget to include the vertical walls at the edges. The final object should be about 100mm in the longest dimension, and 2mm thick.

(3) Create a wire-frame version of the same surface (with 2mm diam. wires).

(4) Create an account on Shapeways.com and upload both surfaces to make sure they are both printable.