Today’s class

- Parametric form
  - curves
  - planes
- Polygonal meshes
- Newell’s method
- Surface models
Circles

- Cartesian equation for a circle centred at the origin is $x^2 + y^2 = r^2$
- This is difficult to draw, since for each value of $x$ that we choose there are two values of $y$ that satisfy the equation
- Thus, the points we generate will not be in succession around the circumference, nor will they be equally spaced
Polar coordinates

\[
\cos \theta = \frac{x}{r}
\]
\[
x = r \cos \theta
\]

\[
\sin \theta = \frac{y}{r}
\]
\[
y = r \sin \theta
\]
Parametric form of a circle

- Let $\theta$ vary between 0 and $2\pi$ in the polar coordinate form of the equation for a circle
  \[ x = r \cos \theta \]
  \[ y = r \sin \theta \]
- Alternatively, let $t$ vary between 0 and 1 in
  \[ x(t) = r \cos 2\pi t \]
  \[ y(t) = r \sin 2\pi t \]
Circle with center not at origin

- Start with a circle centered at the origin
- Translate in the $x$ direction by the $x$ coordinate of the circle’s center
- Translate in the $y$ direction by the $y$ coordinate of the circle’s center

\[
x(t) = c_x + r \cos 2\pi t
\]
\[
y(t) = c_y + r \sin 2\pi t
\]
Ellipses

- Cartesian equation of an ellipse centered at the origin:
  \[ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \]

- Parametric form:
  \[ x(t) = a \cos 2\pi t \]
  \[ y(t) = b \sin 2\pi t \]
Ellipses not centered at the origin

- Cartesian equation of an ellipse centered at \((c_x, c_y)\):

\[
\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} = 1
\]

- Parametric form:

\[
x(t) = c_x + a \cos 2\pi t \\
y(t) = c_y + b \sin 2\pi t
\]
Drawing parametric curves

- Vary \( t \) from 0 to 1 and pick (usually) equally spaced values of \( t \)
- Compute \( x(t) \) and \( y(t) \) at each of these values
- Connect these points with line segments
Planes

- 3 non-collinear points define a plane; let these points be labelled A, B, and C
- The normal to the plane can be determined by taking the cross product of two vectors between the three points (such as A-C and B-C)
Parametric representation of a plane

- Let vector $\mathbf{a} = \mathbf{A} - \mathbf{C}$ and vector $\mathbf{b} = \mathbf{B} - \mathbf{C}$
- Then $\mathbf{P}(s, t) = \mathbf{C} + s\mathbf{a} + t\mathbf{b}$
- Note that this is the same as $\mathbf{P}(s, t) = s\mathbf{A} + t\mathbf{B} + (1-s-t)\mathbf{C}$, which is an affine combination (coefficients sum to 1)
Planar patches

- Given the parametric form, \( P(s, t) = C + sa + tb \)
- If we allow \( s \) and \( t \) to vary between \(-\infty\) and \(+\infty\) then we get an infinite plane
- If \( s \) and \( t \) are restricted to finite ranges we get a part of the plane (it is a parallelogram)
- Frequently \( s \) and \( t \) lie in \([0, 1]\)
Point normal form for a plane

- Given any single point on the plane, say A, and the normal, \( \mathbf{n} \), to the plane
- Then for any arbitrary point, \( R \), on the plane, \( \mathbf{n} \cdot (R - A) = 0 \)
- \( \mathbf{n} \cdot R = \mathbf{n} \cdot A = D \), some constant
Polygonal meshes

- In 3D we deal with faces rather than edges
- Describe an object as a collection of polygons
- Such a collection is a **polygonal mesh**
Modeling a polygon mesh

- An object is a list of faces (polygons)
- Each face contains:
  - a list of vertices
  - a list of normals associated with each vertex
  - any attributes
    - color
    - transparency
    - reflectivity
    - etc.
Normals in OpenGL

- Each vertex in an OpenGL geometry has a normal associated with it.
- Use `glNormal3{b,s,i,f,d}[v]` to specify a normal.
- You must call `glNormal3` before you call `glVertex` if you want the vertex you are specifying to have the normal you want it to.
Newell’s method

- Need to find the normal for a possibly non-planar polygon
- Let’s look at a quadrilateral as an example
- Divide the quadrilateral into two triangles
Newell’s method (cont.)

- Find the normal for each triangle and add them together (this gives a scaled version of the average normal).
- The normal for the triangle made up of P₀, P₁, and P₂ is

\[
\begin{vmatrix}
  i & j & k \\
  x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\
  x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\
\end{vmatrix}
\]

- For the x component of the normal, the value is

\[(y_1 - y_0)(z_2 - z_0) - (y_2 - y_0)(z_1 - z_0)\]
Newell’s method (cont.)

- Similarly, for the triangle made up of $P_0$, $P_2$, and $P_3$, the $x$ component of the normal is $(y_2-y_0)(z_3-z_0) - (y_3-y_0)(z_2-z_0)$.
- The total normal is the sum of the previous two expressions:
  
  $$(y_1-y_0)(z_2-z_0) - (y_2-y_0)(z_1-z_0) +$$
  $$(y_2-y_0)(z_3-z_0) - (y_3-y_0)(z_2-z_0) =$$
  $$y_1z_2-y_1z_0-y_2z_1+y_0z_1+y_2z_3-y_0z_3+y_3z_2+y_3z_0$$
Newell’s method (cont.)

- Add and subtract (giving a zero-sum change) terms of $y_0z_0$, $y_1z_1$, $y_2z_2$, $y_3z_3$
- $y_1z_2 - y_1z_0 - y_2z_1 + y_0z_1 + y_2z_3 - y_0z_3 - y_3z_2 + y_3z_0 + y_0z_0 - y_0z_0 + y_1z_1 - y_1z_1 + y_2z_2 - y_2z_2 + y_3z_3 - y_3z_3$
- Rearrange terms for grouping:
  
  $y_0z_0 + y_0z_1 - y_1z_0 - y_1z_1 + y_1z_1 + y_1z_2 - y_2z_1 - y_2z_2 + y_2z_2 + y_2z_3 - y_3z_2 - y_3z_3 + y_3z_3 + y_3z_0 - y_0z_3 - y_0z_0$
Newell’s method (cont.)

- Grouping the previous expression in the “right” way gives the normal as $(y_0-y_1)(z_0+z_1) + (y_1-y_2)(z_1+z_2) + (y_2-y_3)(z_2+z_3) + (y_3-y_0)(z_3+z_0)$
- This expression can be generalized to $\Sigma(y_i-y_j)(z_i+z_j)$ where $j=(i+1)\%n$
- There are similar expressions for the $y$ component ($\Sigma(z_i-z_j)(x_i+x_j)$) and the $z$ component ($\Sigma(x_i-x_j)(y_i+y_j)$) of the normal
Newell’s method (cont.)

- This is proportional to an “average” normal
- Collinear triples add 0 to the sum
- For a planar polygon the true normal is found
- For a nearly planar polygon an approximate normal is found
Surface model forms

- Implicit form: \( f(x, y, z) = 0 \)
  - Example: sphere, \( x^2 + y^2 + z^2 - r^2 = 0 \)
  - Easy to do an inside-outside test
    - \( f(x,y,z) < 0 \) \( \Rightarrow \) (x,y,z) is inside surface
    - \( f(x,y,z) = 0 \) \( \Rightarrow \) (x,y,z) is on surface
    - \( f(x,y,z) > 0 \) \( \Rightarrow \) (x,y,z) is outside surface

- Parametric form: \( P(u,v)=(X(u,v), Y(u,v), Z(u,v)) \)
  - For a plane, pick 3 non-collinear points
  - Fix 1 point, and find vectors to other 2
  - These two vectors form a basis
  - \( P(u,v) = P_1 + (P_2-P_1)u + (P_3-P_1)v \)
Patches

- When a surface is composed of many pieces, each with its own equation, each piece is called a patch
Planar patches

- Parametric form for the plane equation is $\mathbf{P}(u, v) = \mathbf{c} + a\mathbf{u} + b\mathbf{v}$
- A patch is obtained if we restrict $u$ and $v$ to finite ranges
- The $u$ and $v$ contours of this patch are straight lines
- To draw the patch, choose a projection (orthographic or perspective), compute the projected points for each contour, and connect them with straight lines
Curved surfaces

- Parametric form for a curved surface is $\mathbf{P}(u, v) = X(u, v)\mathbf{i} + Y(u, v)\mathbf{j} + Z(u, v)\mathbf{k}$
- The shape of the surface depends on the choices of functions $X$, $Y$, and $Z$, as well as the ranges on $u$ and $v$
- The $u$ and $v$ contours are curves
- Algorithm to draw a contour:
  - At a set of points (usually equispaced) spread over the range of the non-contour variable compute the values of $\mathbf{P}(u, v)$ for a fixed value of the contour variable
  - Project each point to its 2D location
  - Draw a polyline between the projected points
Ruled surfaces

- If, through every point on a surface, you can draw at least one line that lies entirely on the surface, then the surface is said to be ruled
Parametric form of ruled surfaces

- Take four points that you want to define a surface over.
- Pick two of the points for one edge of the surface and define a parametric function $p_0(u)$ over it.
- Take the other two points and do the same thing, getting $p_1(u)$, defined over the same range of $u$ values.
- Pick some fixed value of $u$, say $u'$, and draw the straight line between $p_0(u')$ and $p_1(u')$.
- This line can be described parametrically by $(1-v)p_0(u') + vp_1(u')$.
- Now let $u$ vary and the parametric form for the ruled surface is $p(u, v) = (1-v)p_0(u) + vp_1(u)$.
Bilinear patches

- Let $p_0(u)$ be a straight line between its two points, $p_{00}$ and $p_{10}$.
- Thus, $p_0(u) = (1-u)p_{00} + up_{10}$.
- Similarly, let $p_1(u)$ be a straight line between its two points, $p_{01}$ and $p_{11}$.
- Thus, $p_1(u) = (1-u)p_{01} + up_{11}$.
- Plugging into the equation for a ruled surface gives:
  \[ p(u,v) = (1-v)[(1-u)p_{00} + up_{10}] + v[(1-u)p_{01} + up_{11}] \]
- Since this is linear in both $u$ and $v$, such a surface is called a **bilinear patch**.
Example bilinear patch

This is a sample bilinear patch using 
(-4, -4, 0), (6, -2, 0), 
(-5, 7, 0), and (5, 6, 0) as the vertices
Bilinear patch code

- The C++ code to do a bilinear patch is available online in file `bilinear.cpp`
A different looking patch

This is the bilinear patch for vertices (3, -2, 0), (3, 2, 0), (4, 5, 0), and (-4, 5, 0)
Normal to a patch

- For a curved surface the normal varies at each point on the surface.
- Assume the surface is locally smooth so that derivatives exist.
- Then, at the point in question, find the tangent along the \( u \) contour and the tangent along the \( v \) contour and take their cross product.
- The two tangents in question are the partial derivatives of \( p \) with respect to \( u \) and \( v \): 
  \[
  \mathbf{n} = \left( \frac{\partial p}{\partial u} \right) \times \left( \frac{\partial p}{\partial v} \right)
  \]
Revolving a curve around an axis

- The curve to the left is defined parametrically as \( c(v) = (R \cos(v), R \sin(v)) \) for \(-\pi/2 \leq v \leq \pi/2\)
- If you were to rotate this curve around the y-axis what would you get?
- Answer: a sphere
Parametric sphere equation

- Let $u$ measure the angle as we go from the positive $x$-axis to the positive $z$-axis.
- The sphere’s parametric equation is defined over $-\pi \leq u \leq \pi$ as
  \[ p(u, v) = (R \cos(v) \cos(u), R \sin(v), R \cos(v) \sin(u)) \]
The sphere

- Here is a picture of the sphere generated using its parametric form.
- Note that drawing the sphere this way is similar to the bilinear patches we did earlier.
General surface of revolution

- Given a profile curve $\mathbf{c}(v) = (x(v), y(v))$
- The surface of revolution has the equation $\mathbf{p}(u, v) = (x(v) \cos(u), y(v), x(v) \sin(u))$
A different profile curve

- What do you think this curve will generate when rotated about the y-axis?
- Can you come up with the parametric representation of this curve?
The goblet

- It generates a goblet
- The parametric representation of the profile curve can be found in the example program `revolution.cpp`
Perspective projections

- For surfaces of revolution we need to transform each point along a contour trace, perspectively project it, and join with a polyline

- Code to do this is also in revolution.cpp
Better view of the sphere

- View is at (5, 5, 5)
- Near clipping plane is 1
- Far clipping plane is 15
Better view of the goblet

- View is at (4, 3, 4)
- Near clipping plane is 1
- Far clipping plane is 20
Traces

- What do we see if we “slice” a surface of revolution by a plane parallel to the $xz$-plane?
- We see a circle.
- What do we see if the plane is parallel to the $xy$-plane?
- We see the profile curve that was used to sweep out the surface of revolution.
- The curves obtained by “slicing” a surface by a plane are called traces.