# 02: Meshes & The Euler Formula

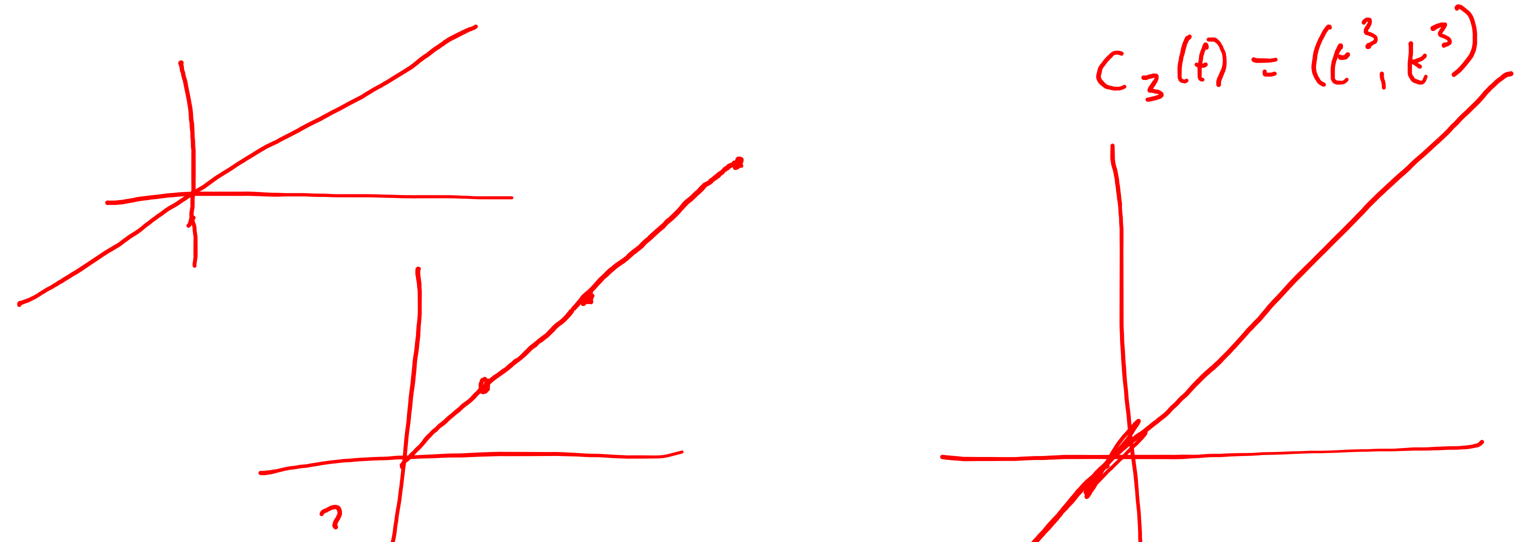
Dr. Hamish Carr



UNIVERSITY OF LEEDS

# Definition of a Curve

- A curve is a set of points given by a function:
  - $\mathcal{C}: \mathbb{R} \rightarrow \mathbb{R}^n$, where n is usually 2 or 3
- Multiple functions describe the same curve:
  - $\mathcal{C}_1(t) = (t, t)$
  - $\mathcal{C}_2(t) = (t^2, t^2)$




- This leads us to differential geometry

# Definition of a Surface

- $S: \mathcal{M} \to \mathbb{R}^3$ where:
  - $S$ is the surface
  - $\mathcal{M}$ is the (2-)manifold it is mapped from
  - $\mathbb{R}^3$ is the embedding space of the surface
- This is a slightly circular definition, but:
  - The position (x,y,z) is just an attribute of $S$
  - And this is often useful later on

# Mathematical Ideal

- Each surface is known analytically
  - Usually parametrised by texture cords $u,v$
  - $(x,y,z)$ are function of $(u,v)$
  - Normal vectors are perfect
  - Any other property is parametrised by $u,v$
    - And can be determined when needed
- But it's not that neat in practice

# The Reality

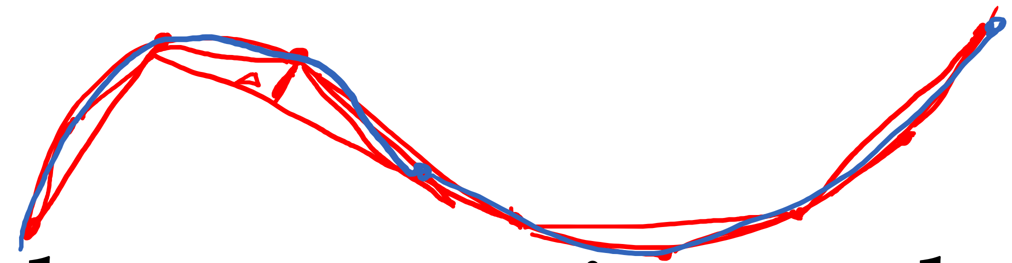
- Can only parametrise simple surfaces
- Parametrisation implies distortion
- Computing intersections, &c. is expensive
- And it's hard to construct a surface
  - i.e. the infamous creative control
  - Artists want to decide what the surface is
  - So they need workable tools

# The Compromise

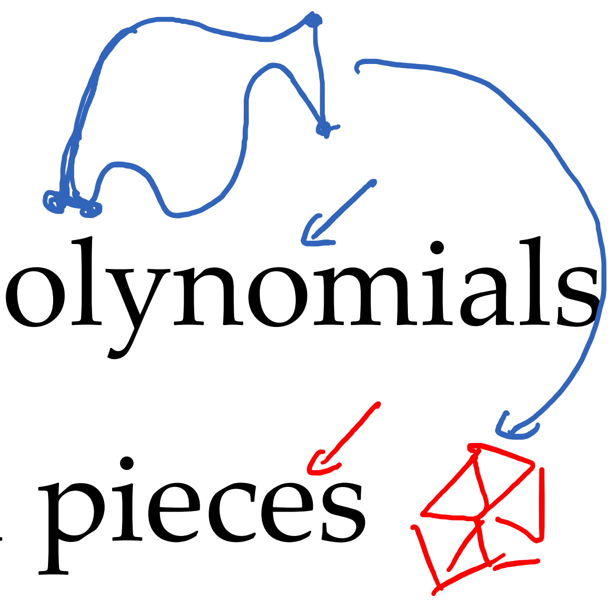
- We use meshes to represent geometry
  - Position, normals, tangents, &c.
- We use textures & maps for properties
  - And store them in arrays
  - Using (u,v) as lookups into the arrays
  - And we can store any lookup we want
  - Heightmaps, lightmaps, shadowmaps, &c.

# Differential Geometry

- Weierstrass Theorem:



  - Any smooth function can be approximated with polynomials to any desired accuracy
  - Two basic choices:
    - p-refinement: more complex polynomials
    - h-refinement: more individual pieces



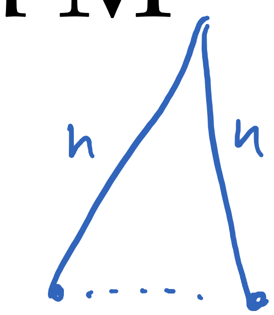
  - p-refinement leads to higher-order surfaces
  - h-refinement means more triangles

# H-Refinement

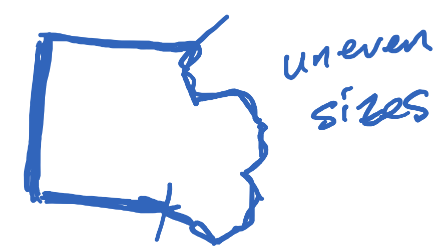
- Just add more triangles / polygons
- But how?
  - We will need *data structures*
  - Which means discussing some basics first
  - And identifying the operations required

# Approximation Error

- Let $h$ be the maximum edge length in M
- The approximation error is $O(h^2)$




- Corollary: high curvature => more edges
  - Perceptual issues also kick in:

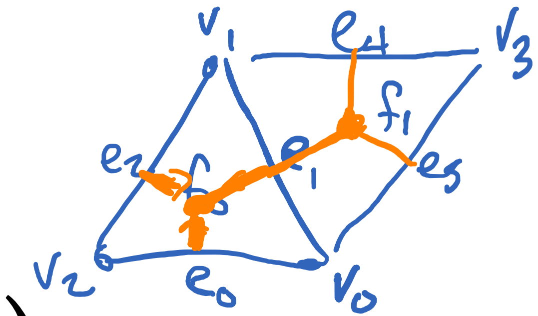



  - Humans pay more attention to:
    - Edges, movement, textural changes
    - Eyes, hands, ...



UNIVERSITY OF LEEDS

# Formal Definition of Mesh




- A mesh consists of:
  - V the vertex set (a set of indices)
  - E the edge set (pairs of indices)
  - F the face set (3+-tuples of indices)
  - P points – a geometric embedding of V
  - A attributes – extra properties
- F, E, V are essentially a graph

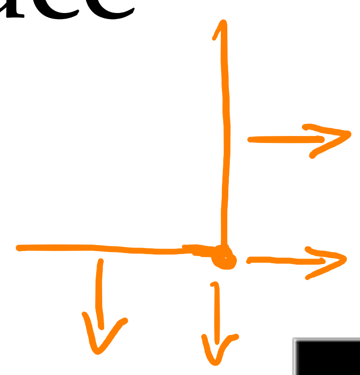
$e_0 = (v_0, v_2)$
$e_1 = (v_1, v_0)$
$e_2 = (v_1, v_2)$
$f_0 = (v_0, v_1, v_2)$

# Mesh Attributes

- Attributes can be:
  - Position (x,y,z,[w])
  - Normal vector (nx, ny, nz)
  - Colour (r,g,b)
  - Texture coordinates (u,v,w)
  - Material (r,g,b + lots of extra stuff)
  - &c., &c., &c.

# Attribute Location

- Attributes can belong to:
  - Vertices
  - Edges
  - Faces
- Or even to a given vertex on a given face
  - -> eg normal along sharp edges





UNIVERSITY OF LEEDS

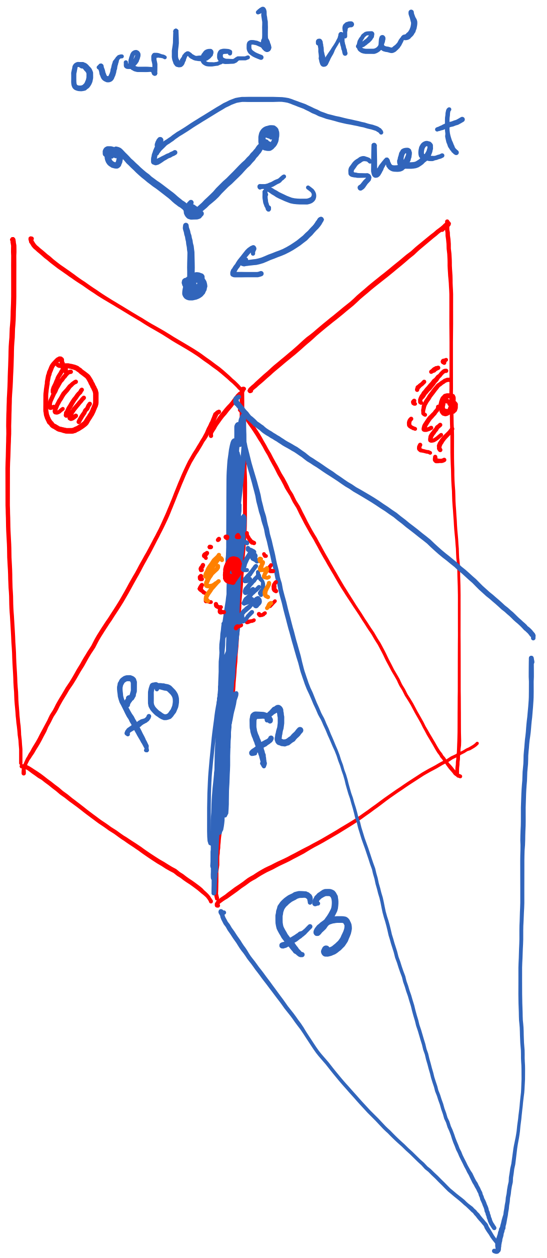
# Fundamental Choice

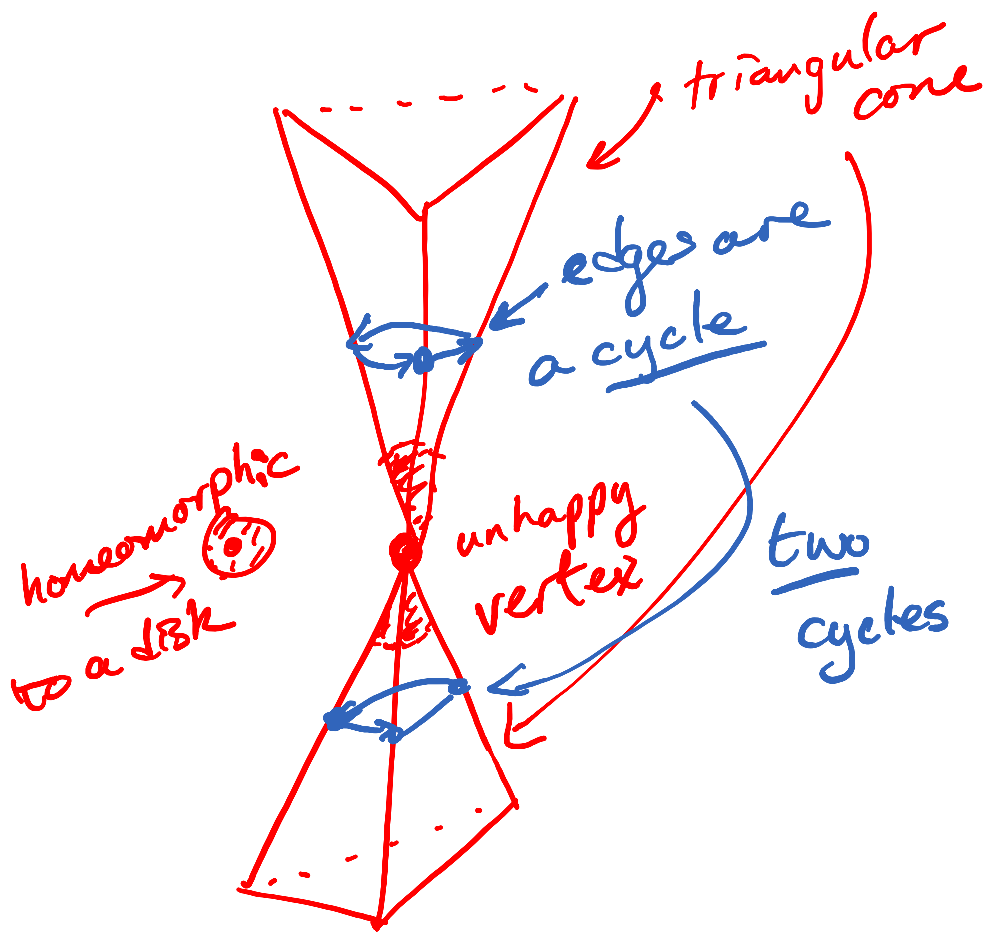
- We can store attributes on the mesh itself
- Or we can store attributes in textures
  - And store texture coordinates in the mesh
- We can even store positions in textures
  - But mostly we do positions directly

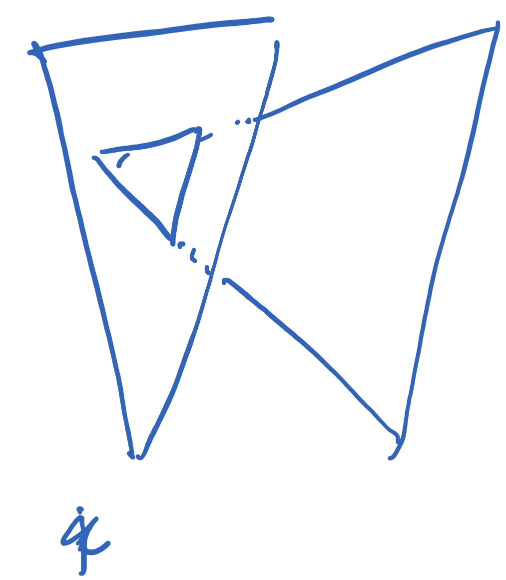
# Manifold Meshes

- Triangle mesh is 2-manifold iff:
  - all edges share two faces
  - no pinch points at vertices
    - i.e. single cycle around each vertex
  - no self-intersections
- A non-self-intersecting, closed polyhedron

# Manifold Sketches



non-manifold

Condition 1: every edge has exactly 2 faces

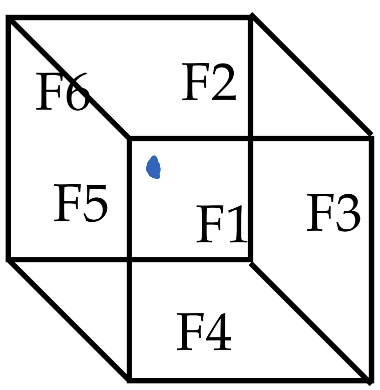
non-manifold pinch point

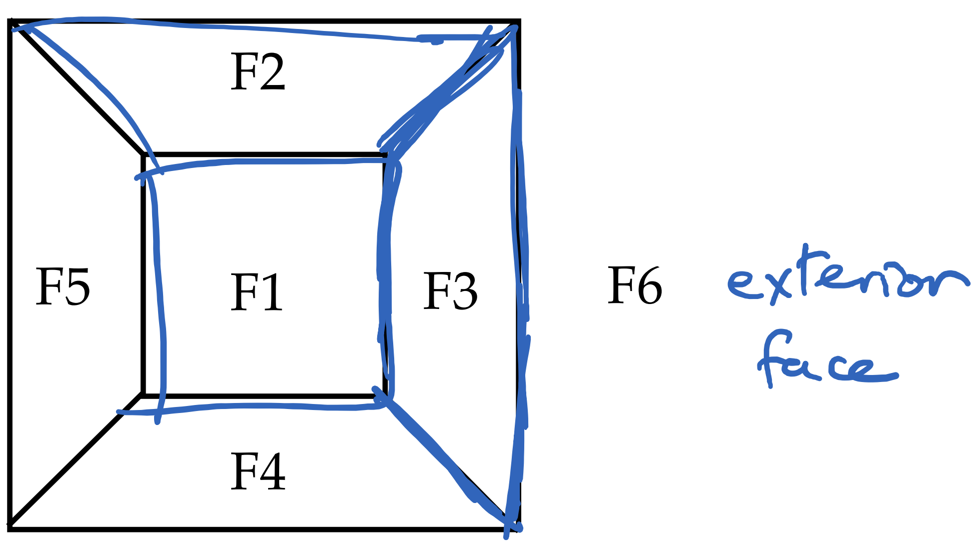
Condition 2: edges at a vertex form a single cycle

non-manifold interpenetration

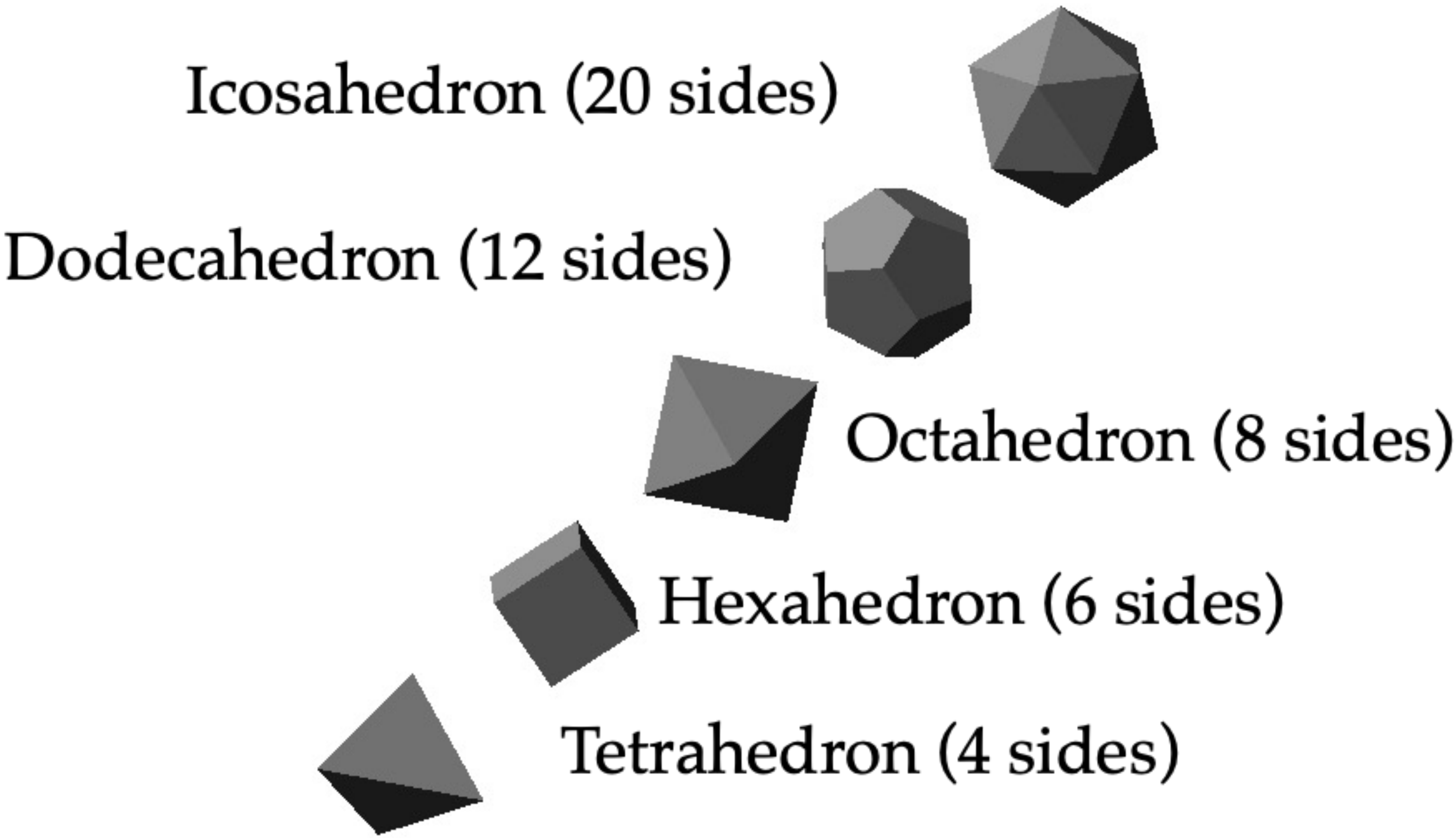
# Unwrapping Meshes

- Meshes *unwrap* to become (planar) graphs
- Choose one face and cut a small circle in it
    - Expand it to infinity
    - I.e. reverse so-called one-point unification

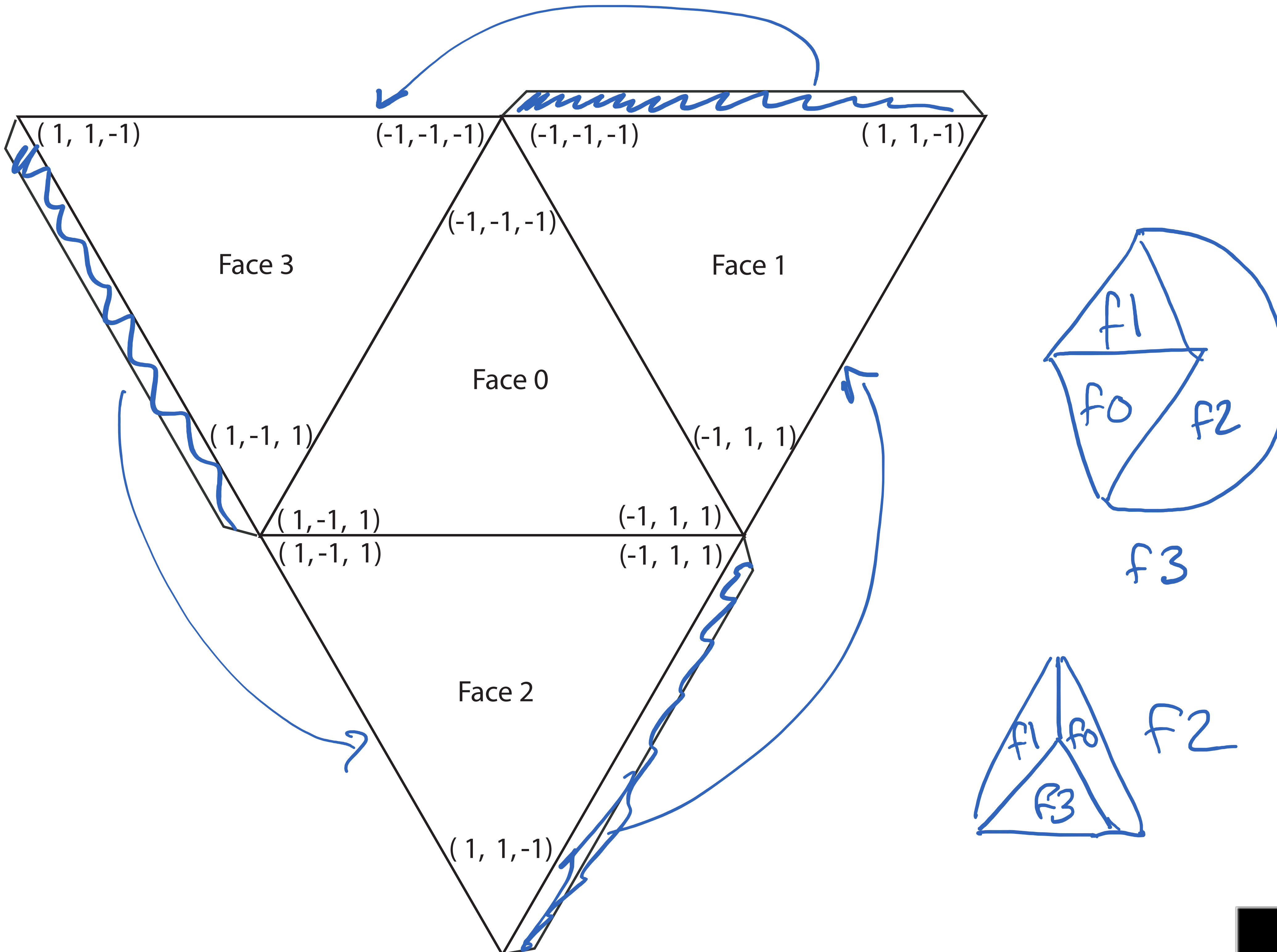


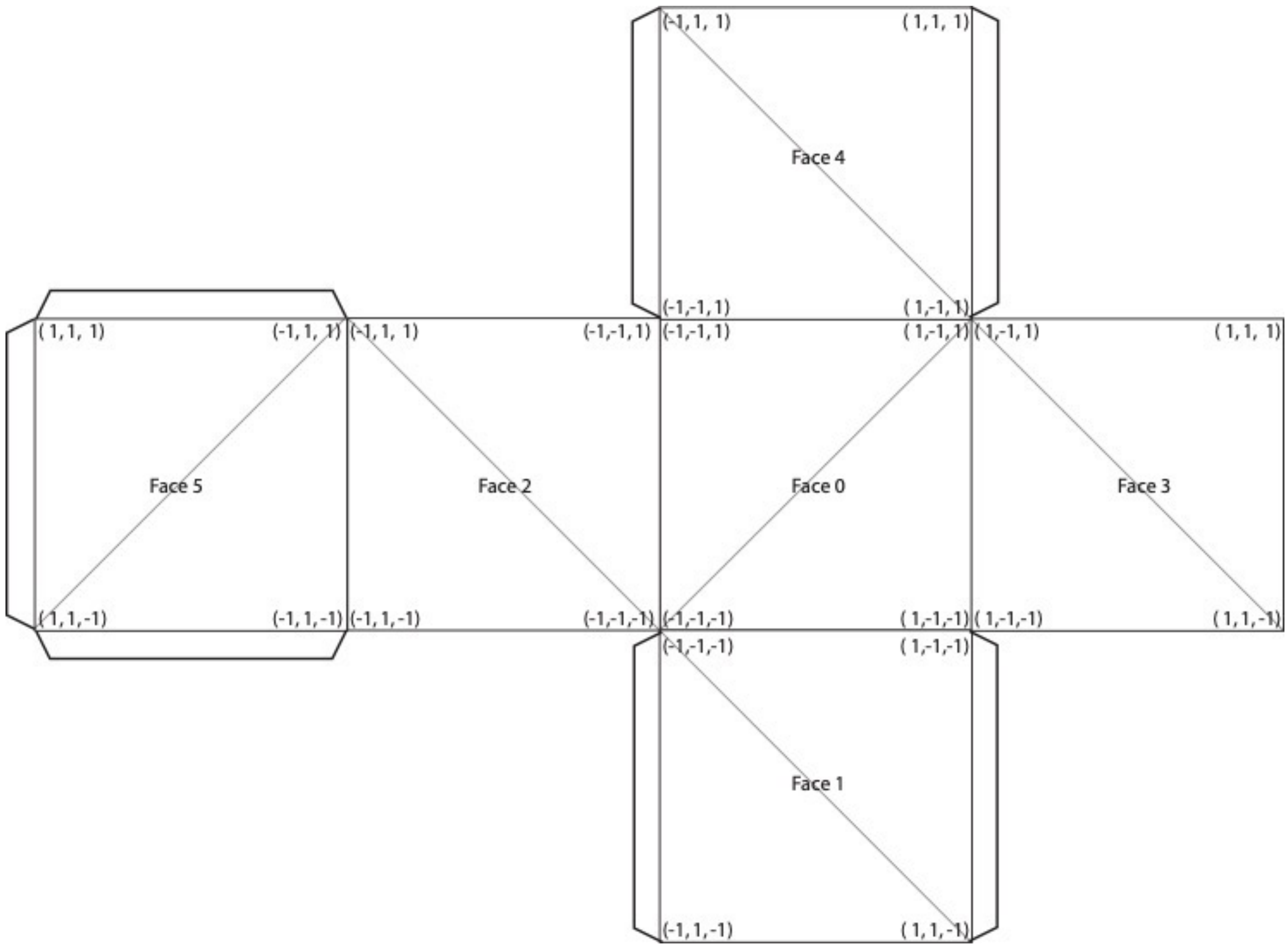
# Platonic Solids



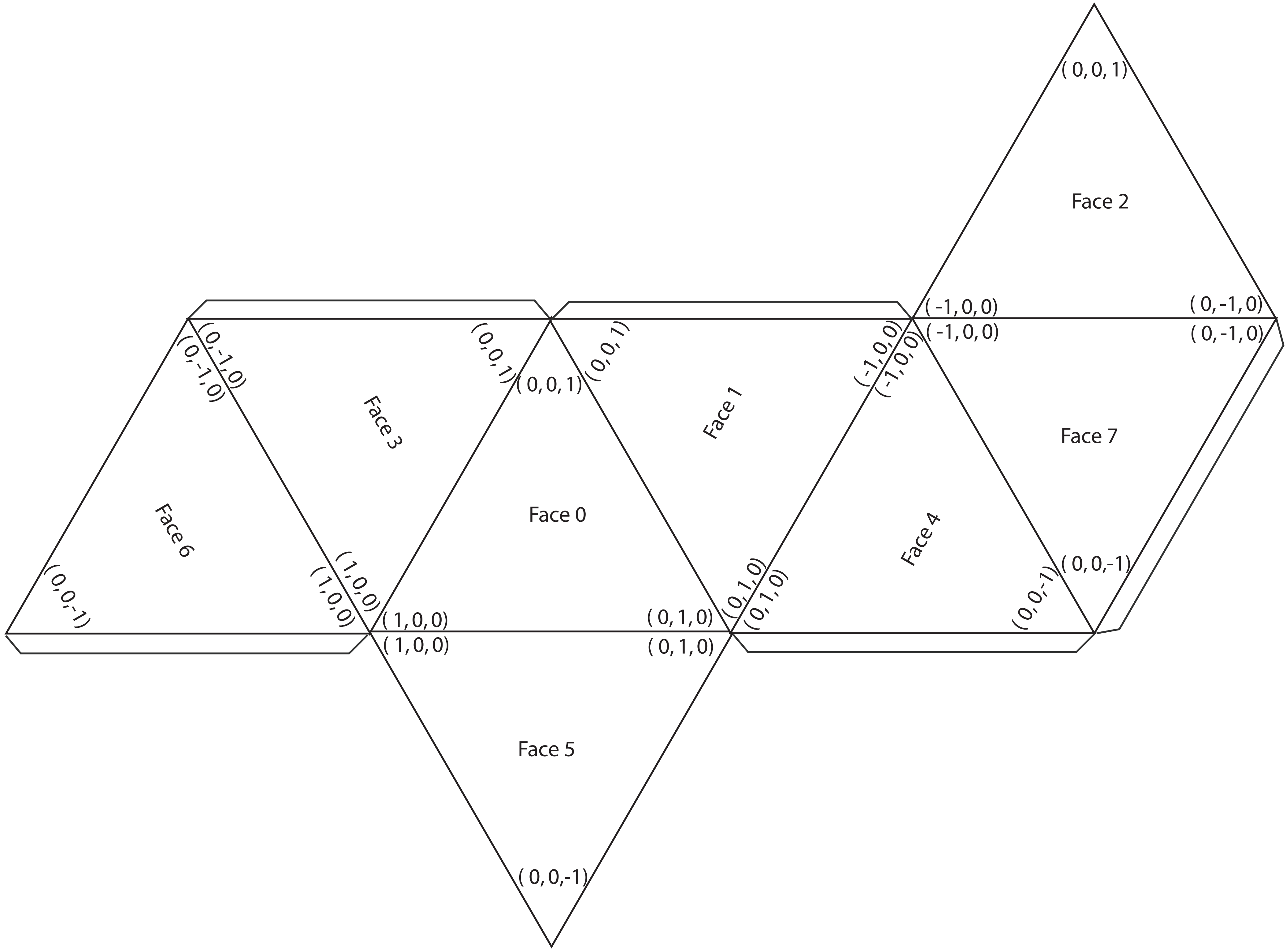
# Tetrahedron



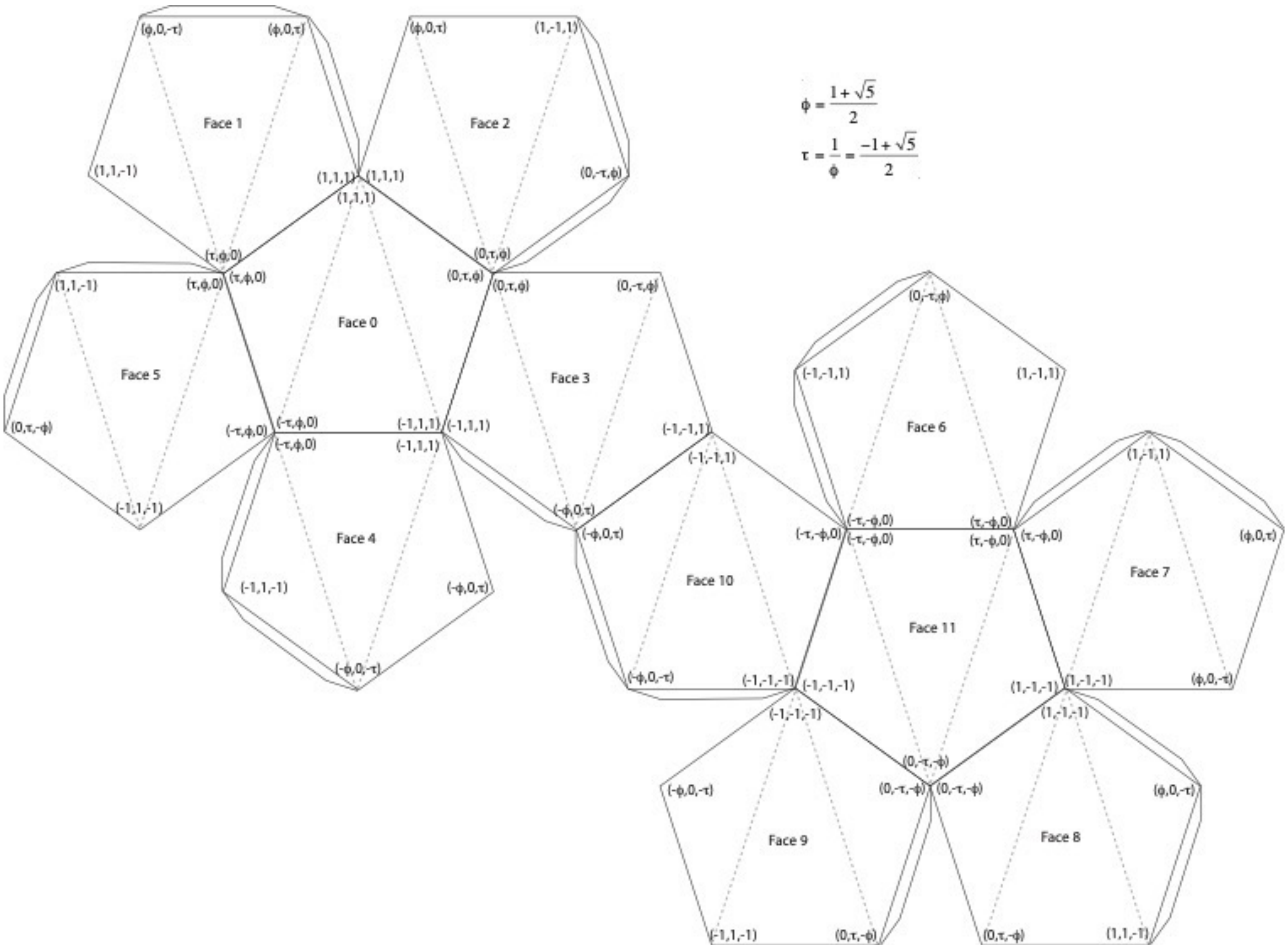
# Hexahedron (Cube)



# Octahedron


( 0, 0, 1)
Face 2
( -1, 0, 0)
( 0, -1, 0)
( -1, 0, 0)
( 0, -1, 0)
( 0, -1, 0)
( 0, -1, 0)
( 0, 0, 1)
( 0, 0, 1)
( 0, 0, 1)
( -1, 0, 0)
( -1, 0, 0)
Face 3
Face 1
Face 7
Face 6
Face 0
Face 4
( 0, 0, -1)
( 1, 0, 0)
( 1, 0, 0)
( 1, 0, 0)
( 0, 1, 0)
( 0, 1, 0)
( 0, 1, 0)
( 0, 0, -1)
( 0, 0, -1)
( 1, 0, 0)
( 0, 1, 0)
Face 5
( 0, 0, -1)

# Dodecahedron

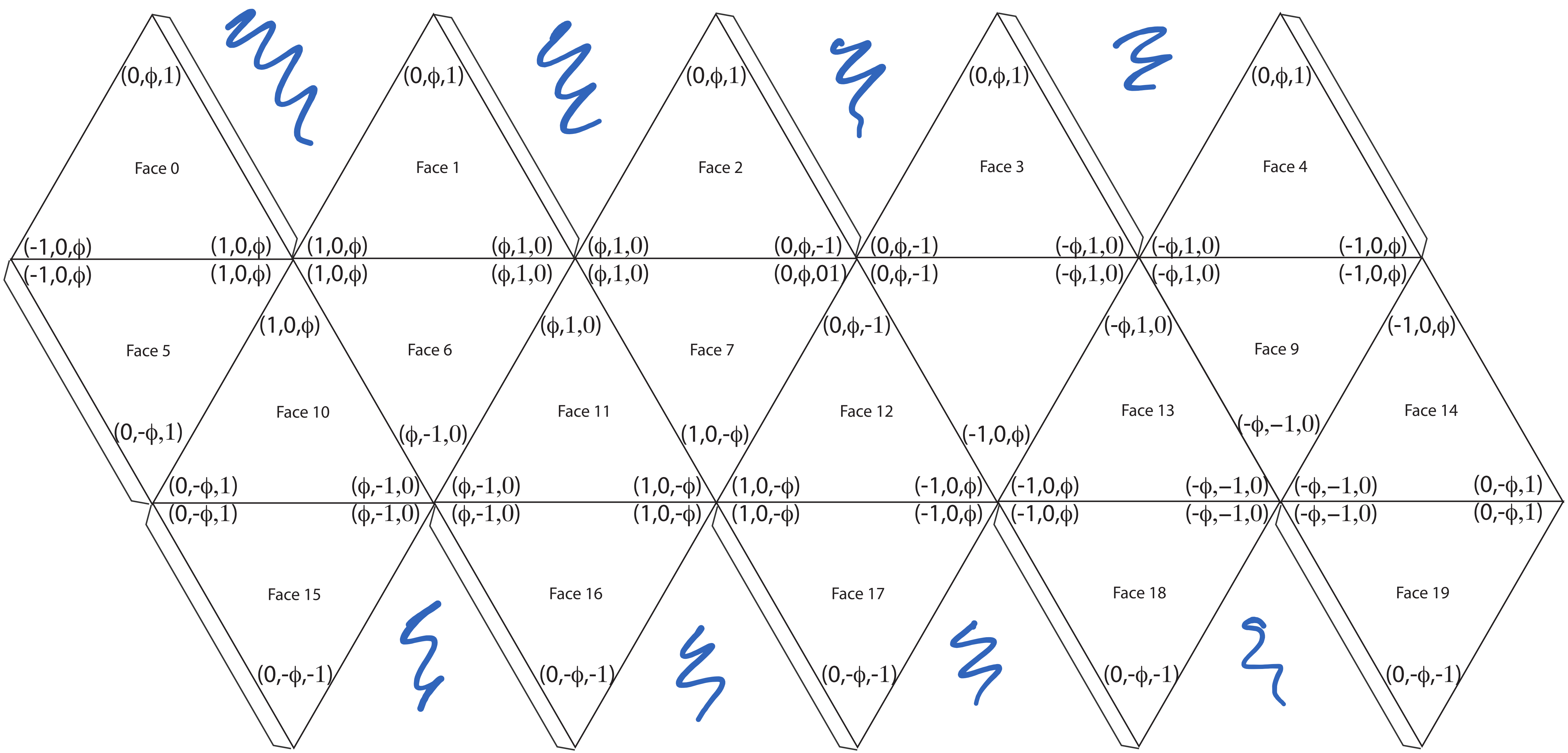



$$\phi = \frac{1+\sqrt{5}}{2}$$

$$\tau = \frac{1}{\phi} = \frac{-1+\sqrt{5}}{2}$$

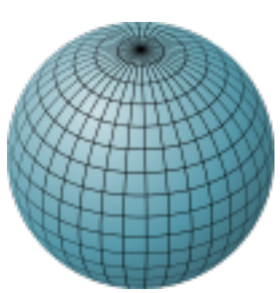
# Icosahedron

$$\phi = \frac{1+\sqrt{5}}{2}$$



# Simple Polyhedra ⇔ Planar Graphs

- Euler (18th c.) Oiler
- But not all polyhedra are simple
  - There is a phenomenon called *genus*
  - The number of *handles* through the surface
  - Or the number of independent cycles

ball  torus coffee cup    handle 

$g=0$ $g=1$ $g=2$ $g=3$

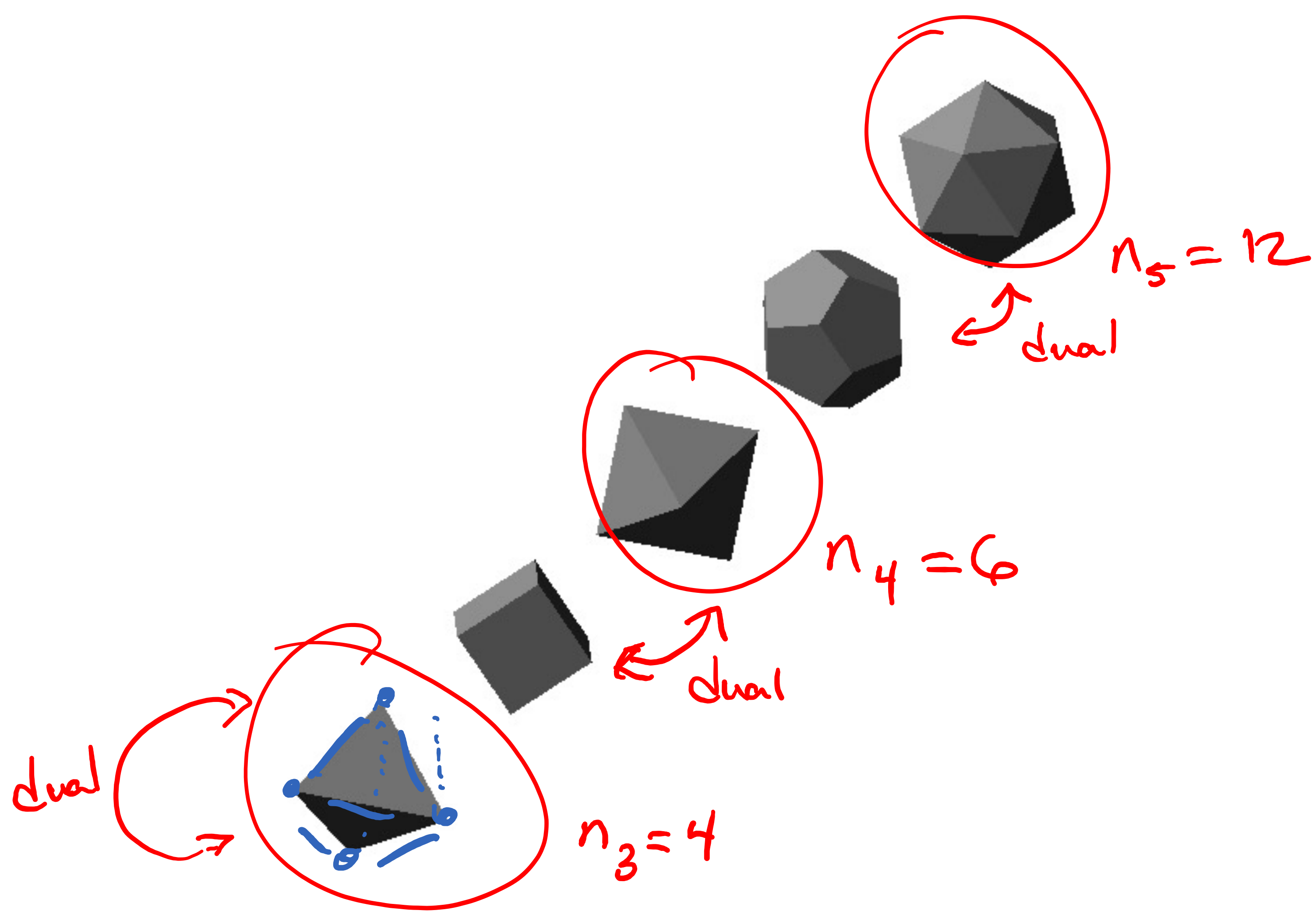
from wikipedia ...

# Euler's Formula

- In any orientable mesh/polyhedron:
  - $v - e + f = 2 - 2g$
    - v is number of vertices
    - e is number of edges
    - f is number of faces
    - g is the genus

# Some Examples

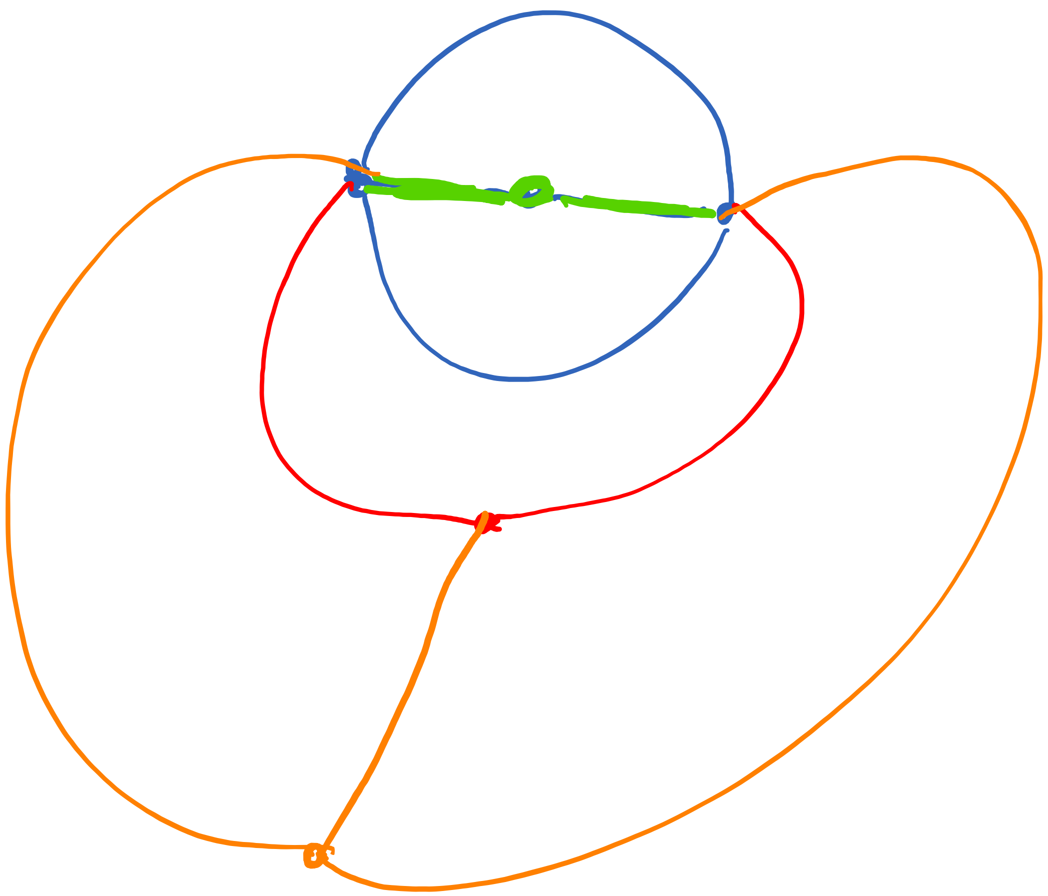
| | v | e | f | g |
|---|---|---|---|---|
| Tetrahedron | 4 | 6 | 4 | $= 2 - 2(0)$ |
| Cube | 8 | 12 | 6 | 0 |
| Octahedron | 6 | 12 | 8 | 0 |
| Dodecahedron | 20 | 30 | 12 | 0 |
| Icosahedron | 12 | 30 | 20 | 0 |



# Sketch of Proof

- Start with a single cycle with 2 faces:
- Add in vertices one at a time
  - Keeping track of vertex count, &c.
- Leads to inductive proof
- The basis of *nearly all* planar graph algorithms
  - and *all* polygon mesh data structures, &c.

# Sketch of Proof



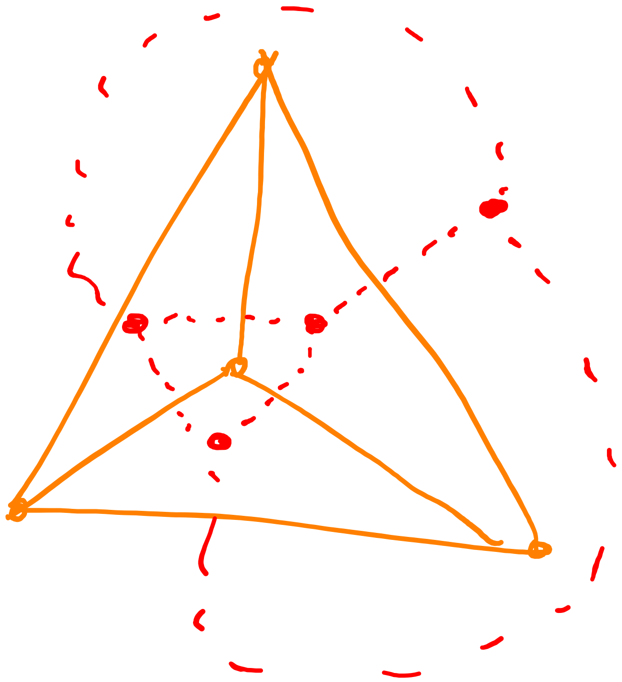
| v | e | f | g |
|---|---|---|---|
| 2 | − 2 | + 2 | = 2 − 2(0) |
| 3 | − 4 | + 3 | = 2 − 2(0) |
| 4 | − 6 | + 4 | = 2 − 2(0) |
| 5 | − 9 | + 6 | = 2 − 2(0) |

# Exercise

- Convert each Platonic solid to a planar graph
- Compute f, e, v and g
- Remove one vertex & re-triangulate face
- Repeat until done

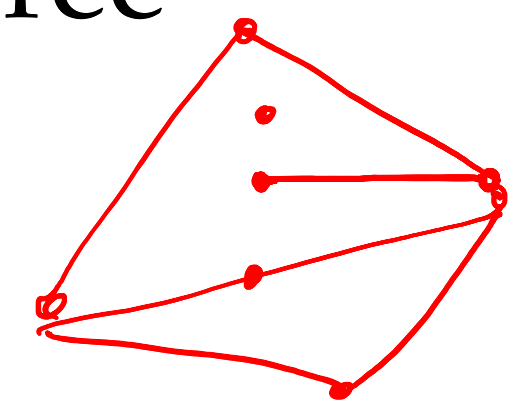
# Vertex-Face Duality

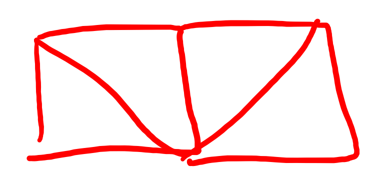
- Put a vertex at the centre of each face
- Connect faces if adjacent
- Constructs new faces around vertices
- Vertices & faces are *dual*
- Tetrahedron is self-dual
- Cube is dual to octahedron
- Dodecahedron is dual to icosahedron



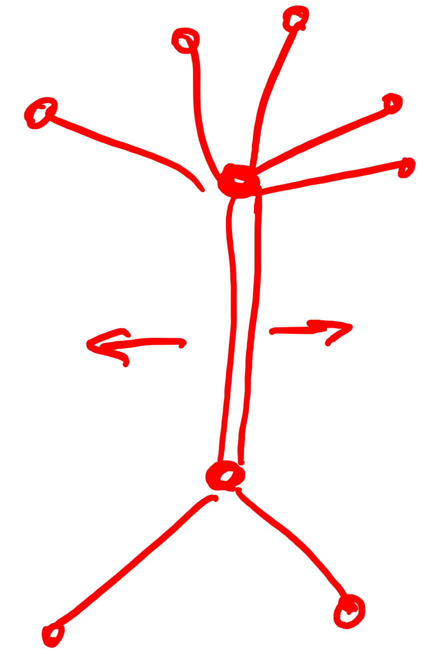
# Triangulation

- A mesh *all* of whose faces are triangular
  - And therefore easy to work with
- Vertices can be arbitrary degree
  - But not 0, 1, or 2
- Can *always* be constructed
  - e.g. barycentric face refinement
- Therefore, the base assumption for everything





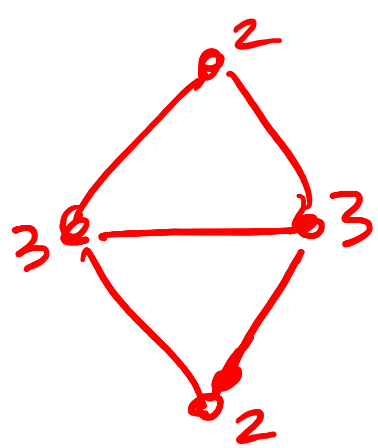
# Edges & Faces



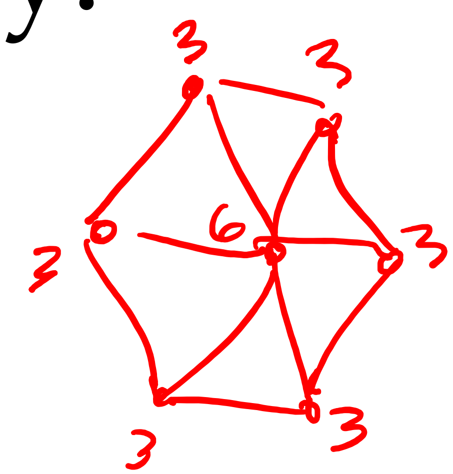
- Each edge belongs to 2 faces
- Each face has 3 edges, so:
  - $3f = 2e$
- Each edge also has 2 vertices
- Each vertex has an arbitrary number of edges
  - So the sums get a bit messier

# Edges & Vertices

- So sum up the vertex degrees:
  - $2e = \underbrace{\sum_{i=1}^{v}}_{\text{all vertices}} \underbrace{\delta(v_i)}_{\text{vertex degree}}$
- Alternately, count vertices of each degree:
  - $2e = \sum_{j=3}^{\infty} j \cdot n_j$
- And count the vertices separately:
  - $v = \sum_{j=3}^{\infty} n_j$




$n_2 = 2$
$n_3 = 2$




$n_3 = 6$
$n_4 = 0$
$n_5 = 0$
$n_6 = 1$

# Substitutions

$$
\begin{aligned}
v - e + f &= 2 - 2g \\
3v - 3e + 3f &= 6 - 6g \\
3v - 3e + 2e &= 6 - 6g \\
3v - e &= 6 - 6g \\
6v - 2e &= 12 - 12g \\
6\sum_{i=3}^{\infty} 1 \cdot n_i - \sum_{i=3}^{\infty} i \cdot n_i &= 12 - 12g \\
\sum_{i=3}^{\infty} (6 - i) \cdot n_i &= 12 - 12g \\
\sum_{i=3}^{6} (6 - i) \cdot n_i &= 12 - 12g - \sum_{i=7}^{\infty} (6 - i) \cdot n_i \\
\sum_{i=3}^{6} (6 - i) \cdot n_i &= 12 - 12g + \sum_{i=7}^{\infty} (i - 6) \cdot n_i \\
3n_3 + 2n_4 + 1n_5 + 0n_6 &= 12 - 12g + 1n_7 + 2n_8 + 3n_9 + \ldots
\end{aligned}
$$

$\times 3$

$\times 2$

# Consequences

$$3n_3 + 2n_4 + 1n_5 + 0n_6 \quad = \quad 12 - 12g + 1n_7 + 2n_8 + 3n_9 + \ldots$$

- There is *always* a vertex of degree 3, 4, or 5
  - at least in a planar graph / genus 0 surface
- The only ways the mesh can get bigger
  - by adding degree 6 vertices
  - or balancing high & low degree vertices
- If genus is low (and it *always* is)
  - so we mostly ignore it

# More Consequences

$$3n_3 + 2n_4 + 1n_5 + 0n_6 = 12 - 12g + 1n_7 + 2n_8 + 3n_9 + \ldots$$

- 3v per face, 6f per vertex means that:
  - f ~ 2v, e ~ 3v
  - and everything scales on the # of vertices
- The formula is a checksum for correctness!
- A genus 1 surface (torus) can be regular
  - i.e. all vertices the same degree (6)
  - No others can, except small ones

$n_3 = 4$ or

$n_4 = 6$ or

$n_5 = 12$

# Quad Meshes

- A quad mesh is similar, starting with:
  - 4f = 2e
- Similar substitutions lead to:
  - $n_3 + 0\, n_4 = 8 \; - 8g + n_5 + 2n_6 + 3n_7 + \cdots$
- A quad mesh mostly has degree 4 vertices
  - Other vertices are *extraordinary*
  - See subdivision surfaces . . .