Exercises for Advanced Graphics (Lectures 1-4)

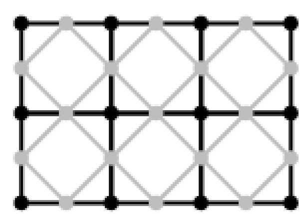
All work to be submitted by email in PDF format, no less than 48 hours before supervision.

These exercises are partly drawn from past exam questions.

1. **B-Splines and Beziers**
    a. Show that the B-spline with k=3 and knot vector [0 0 0 1 1 1] is equivalent to the quadratic Bezier curve.
    b. Give a knot vector and value of k which would describe a uniform B-spline equivalent to a cubic Bezier curve.

2. **B-splines**
    a. Derive the formula of and sketch a graph of N3,3(t), the third of the quadratic B-spline basis functions, for the knot vector [0 0 0 1 3 3 4 5 5 5].
    b. We have already seen that, for a given order, k, there is only one basis function for uniform B-splines. Every control point uses a shifted version of that one basis function. How many different basis functions are there for open-uniform B-splines of order k=2? Order k=3? For the general case, that is: for arbitrary k?
    c. Imagine a type of B-spline where adjacent knots are separated by either 1 or 0. This gives uniform knot separation with the added possibility of multiple knots. e.g. (0,0,0,0,1,2,3,3,3); (0,0,1,1,2,2); (0,1,1,1,1,2,2,3,4,4).How many different basis functions will be needed for this type of B-spline for k=2? For k=3? For arbitrary k? (Hint: you may like to start by considering the case k=1).

3. **Doo-Sabin and Reif-Peters Subdivision**

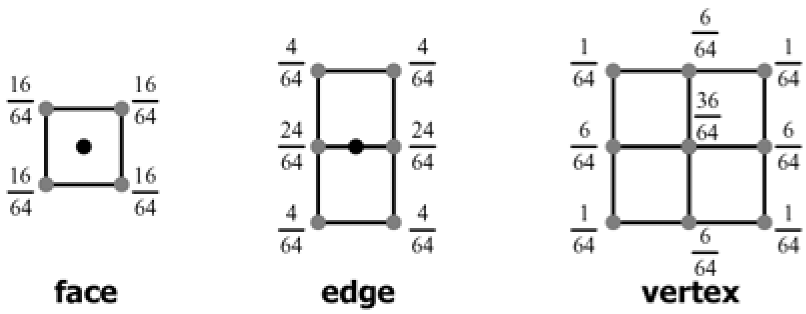
    The Reif-Peters subdivision scheme is illustrated at right. Reif-Peters uses a different approach to Doo-Sabin: in it the new points are generated halfway between existing points and connected up into a mesh as illustrated in the diagram on the right (black: original mesh; grey: new mesh). Note that (i) the existing points do not form part of the new mesh and (ii) each new point's position is simply the average of the positions of the two existing points at either end of the corresponding line segment.
    a. Doing the Reif-Peters subdivision twice produces a mesh which looks similar to that produced by doing the Doo-Sabin subdivision once. You can treat two steps of Reif-Peters as if it were a single

step of a Doo-Sabin-like subdivision scheme. Calculate the weights on the original points for each new point after two steps of Reif-Peters.

b. For the Reif-Peters scheme, explain what happens around extraordinary vertices and what happens around extraordinary polygons, giving examples.

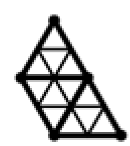
4. **Catmull-Clark Subdivision**

The Catmull-Clark bivariate subdivision scheme is a bivariate generalisation of the univariate 1/8 [1, 4, 6, 4, 1] subdivision scheme. It creates new vertices as blends of old vertices in the following ways:



a. Provide similar diagrams for the bivariate generalisation of the univariate four-point interpolating subdivision scheme 1/16 [-1, 0, 9, 16, 9, 0,-1].

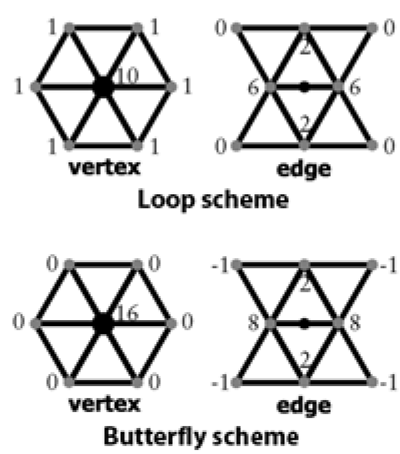
b. Explain what problems arise around extraordinary vertices (vertices of valency other than four) for this bivariate interpolating scheme and suggest a possible way of handling the creation of new edge vertices when the old vertex at one end of the edge has a valency other than four.



every triangle is split into four

5. **Triangular Subdivision**

The Loop and Butterfly subdivision schemes can both operate on triangular meshes, in which all of the polygons have three sides. Both schemes subdivide the mesh by introducing new vertices at the midpoints of edges, splitting every original triangle into four smaller triangles, as shown at right. Each scheme has rules for calculating the locations of the new "edge" and "vertex" vertices based on the locations of the old vertices. These rules are shown below. All weights should be multiplied by 1/16.



a. Which of the two schemes produces a limit surface which interpolates the original data points?
b. Which of the four rules must be modified when there is an extraordinary vertex? For each of the four rules either explain why it must be modified or explain why it does not need to be modified.
c. Suggest appropriate modifications where necessary to accommodate extraordinary vertices.

6. **Implicit Surfaces**

Assume that you have built an implicit surface renderer which renders isosurfaces of the set of points P in space where the sum of all forces at P is exactly equal to 1.

a. You have two point forces positioned at $(-\sqrt{2}, 0, 0)$ and $(\sqrt{2}, 0, 0)$. What is the value of Jim Blinn's Metaballs equation (a=1,b=3) at (0, 0, 0)? (0, 1, 0)?

A point force is an expression written in terms of radius. Assuming that metaball() is the function for Jim Blinn's Metaballs expression, here is the pseudocode of a point source centered at the origin:

```
function float F(P) {
  return metaball(length(P));
}
```

b. Give an expression for a point source centered at (1, 1, 1).
c. Give an expression for a *meta line*. This should be a line which radiates force uniformly in all directions. Its isosurface would be a cylinder section with hemispherical caps (like a jelly bean).
d. Give an expression for a *meta torus*.

We can also choose more blending functions with different traits. As observed in lecture, metaball() is only one possible blending function.

e. Suggest a blending function which would cause two point sources near to each other to produce an isosurface of two spheres intersecting, instead of the "blobby" blending that metaball() presents.

7. **Voronoi Diagrams**
a. What is *equiangularity*?
b. What is the *empty circle property*?
c. Describe how to use hardware acceleration to swiftly compute Voronoi diagrams. What are the limitations of this approach?

8. **Curvature**

The *one-ring* of a vertex is the (usually ordered) set of vertices which lie exactly one edge away from a given vertex on a polyhedral surface. Given a vertex V with one-ring $\{v_0,...,v_{n-1}\}$, give a formula for the discrete curvature of the surface at V.