Dome Projection using a Vertex Shader

I believe that it's only a matter of time until it becomes a routine visualization tool.
Programming a Dome display is easier when only a single projector is used

A fisheye lens distorts the image so that it spreads out across the dome. The trick is pre-distorting the image in the other direction so that it looks correct after being projected.

Dome Distortion

Move the teapot so it surrounds the audience
**Dome Projection:**

Viewing Volume = (-1,-1) to (1,1)

The edge of the circle represents the edge of the dome projection = your left, right, bottom, top as you are sitting in the theater.

**Dome Vertex Shader:**

```
const float PI = 3.14159265;

void main()
{
    vec4 pos = gl_ModelViewMatrix * gl_Vertex;
    float lenxy = length(pos.xy);

    float phi = atan(lenxy, -pos.z);
    pos.xy = (phi / (PI/2.)) * (pos.xy / lenxy);

    float phi = atan(lenxy, -pos.z);
    pos.xy = (phi / (PI/2.)) * (pos.xy / lenxy);

    gl_Position = gl_ProjectionMatrix * pos;
}
```

*Note: (pos.xy / lenxy) = (cosθ, sinθ)*
Dome Vertex Shader:

Cartesian:  
Dome:

Dome:
Flow Visualization in the Dome

Mars Panorama in the Dome
Large Lines and Polygons Need to be Tessellated

Bounding Box edges were not tessellated. Straight lines on the monitor produced curved lines on the dome.

Note: This edge does not pass through the flow vectors!

Bounding Box edges were tessellated. Curves lines on the monitor produced straight lines on the dome.

Note: This edge does pass through the flow vectors!