

```

100
101
102 if (cellTime > 0.) {
103     // wedge
104     float ins = .25;
105     p.z += ins;
106     vec3 n = normalize(vec3(1,0,.35));
107     wedge = -dot(p, n);
108     wedge = max(wedge, dot(p, n * vec3(1,1,-1)));
109     wedge = smax(wedge, p.z - len*1.12 - ins, len);
110     p.z -= ins;
111
112     // wedge2
113     ins = .2;
114     p.z += ins;
115     n = normalize(vec3(1,0,.4));
116     float wedge2 = -dot(p, n);
117     wedge2 = max(wedge2, dot(p, n * vec3(1,1,-1)));
118     wedge2 = smax(wedge2, p.z - len*.95 - ins, len*.6);
119     p.z -= ins;
120
121     float r = len / 8.;
122
123     float top = p.y - len * .5;
124     float curve = smoothstep(0., .2, cellTime);
125
126     len *= mix(1.5, .65, curve);
127     pR(p.zy, -mix(.2, .7, curve));
128     slice = length(p - vec3(0,len,0)) - len;
129     d2 = abs(slice) - .05;
130     d2 = max(d2, top);
131
132     float d3 = smax(d2, wedge, .05);
133     float d4 = smax(d2, wedge2, .05);
134     wedges = smin(wedge, wedge2, .01);
135     d3 = smin(d3, d4, .01);
136     d = d3;
137
138     p = pp;
139     len = llen;
140     vec2 uv = p.xz / len;
141     return Model(d, p, uv, cell, wedges, slice, len);
142 }
143
144 return Model(d, p, vec2(0), vec2(0), 0., 0., 0.);
145 }
146
147
148 vec3 calcAlbedo(Model model) {
149     vec3 col = vec3(.15,.15,.4);
150
151     vec3 p = model.p;
152     float len = model.len;

```


