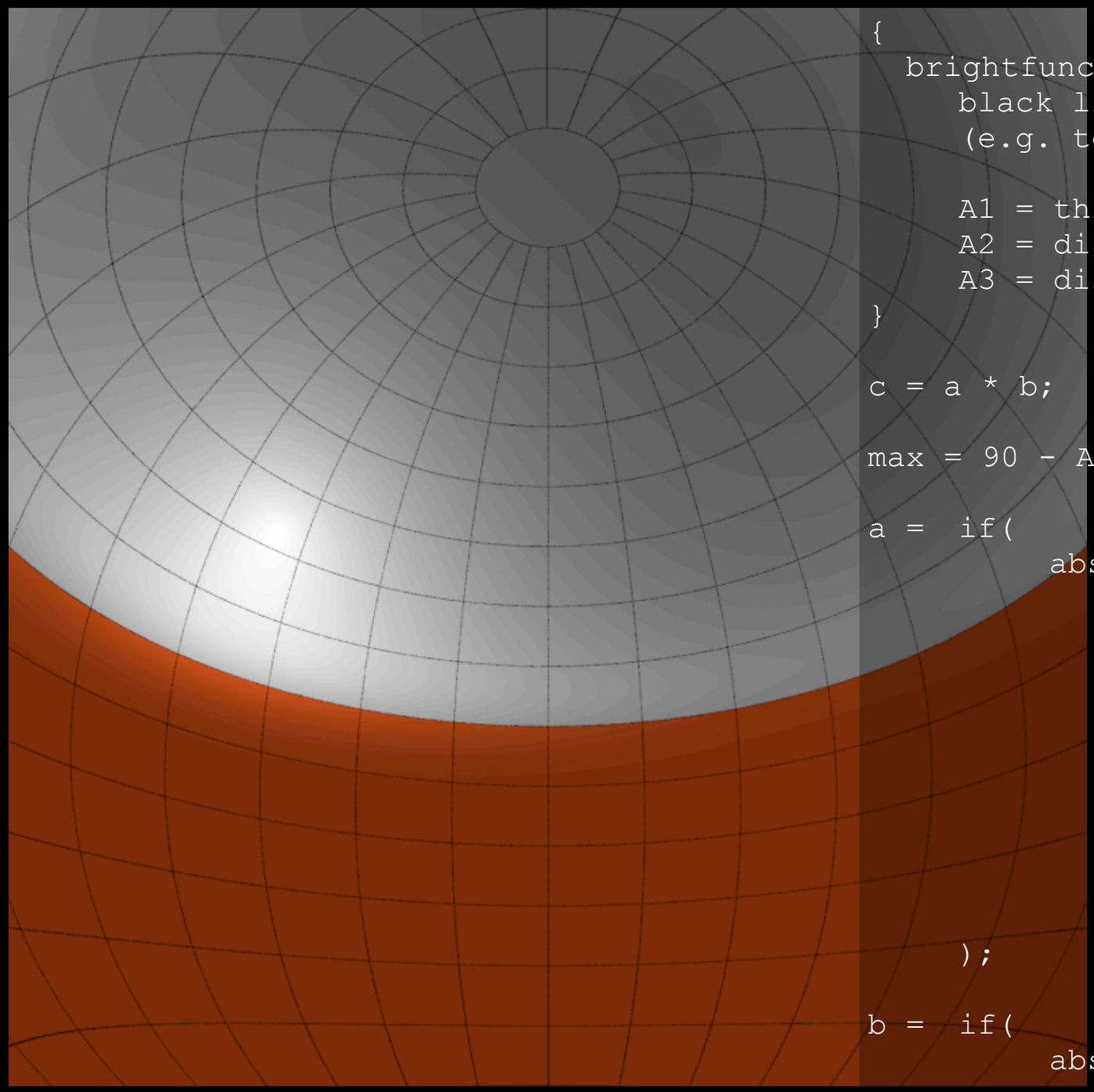
# cal files and daylight studies

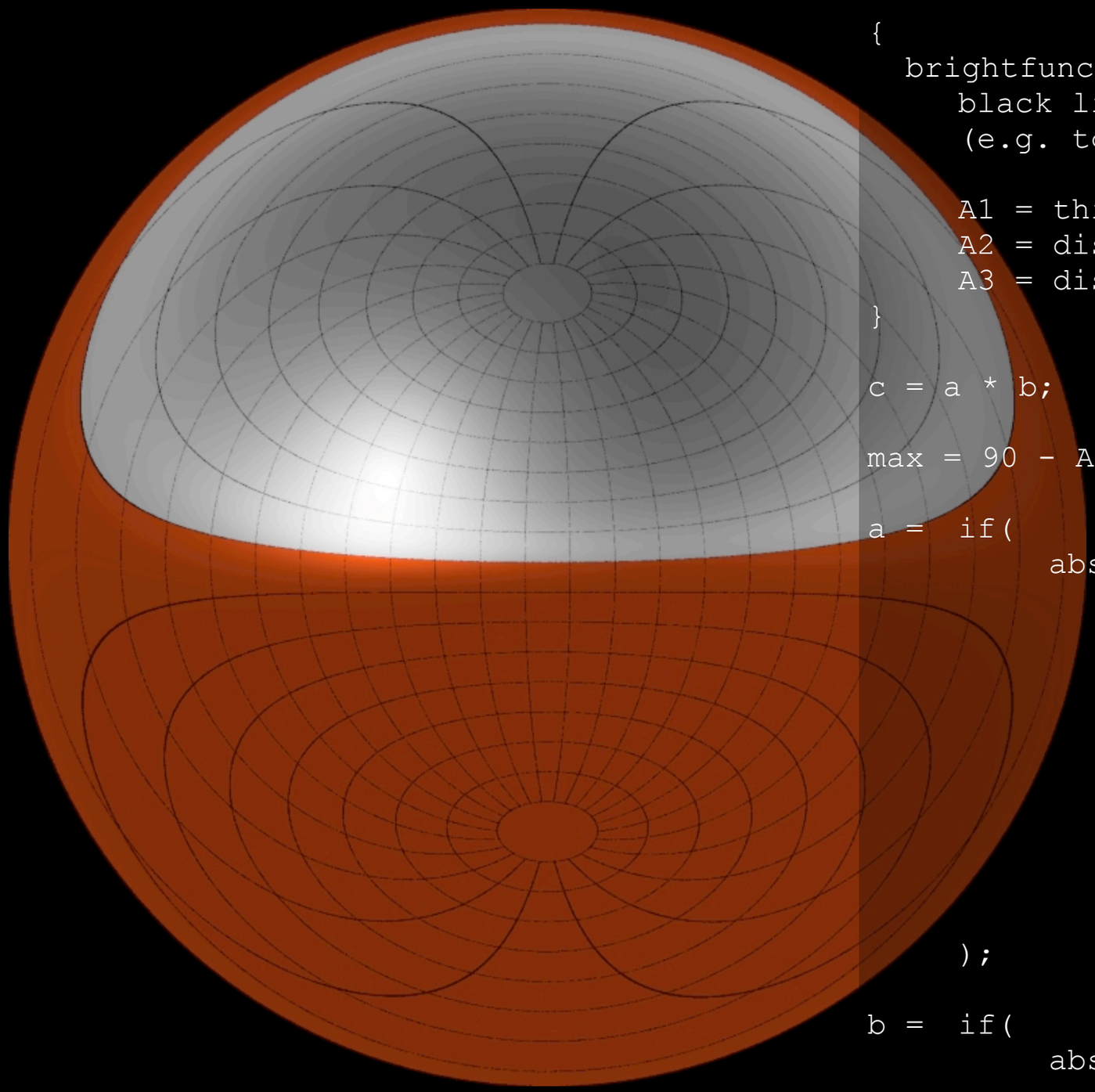
9th International Radiance Workshop
September 2010
Santiago Torres - Arup | Lighting

# alt-azimuth grid



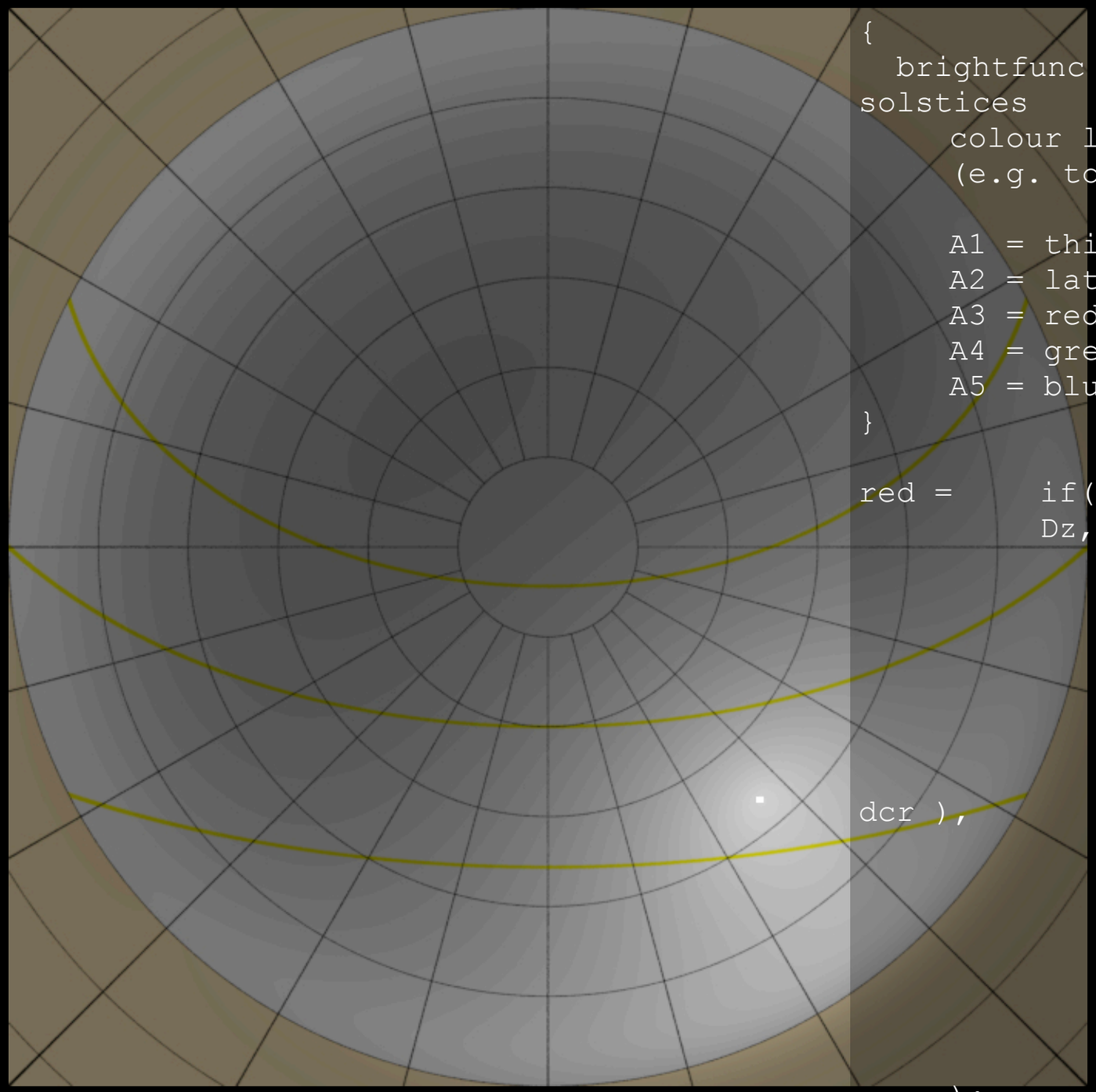
```
{
  brightfunc to generate a polar grid in the sky
     black lines on clear background
     (e.g. to be used in conjunction with a CIE sky)

     A1 = thickness of lines
     A2 = distance in degrees between lines in azimuth
     A3 = distance in degrees between lines in altitude
}

c = a * b;

max = 90 - A3;

a =  if(
          abs(asin(Dz)*180/PI) - max - v*A3 ,
               1,
               if(
                    h - ( az - floor(az) ) ,
                         0,
                         if(
                              h - ( ceil(az) - az ),
                                   0,
                                   1
                         )
               )
     );

b =  if(
          abs(asin(Dz)*180/PI)- max - v*A3 ,
```

# alt-azimuth grid



```
{
  brightfunc to generate a polar grid in the sky
     black lines on clear background
     (e.g. to be used in conjunction with a CIE sky)

     A1 = thickness of lines
     A2 = distance in degrees between lines in azimuth
     A3 = distance in degrees between lines in altitude
}

c = a * b;

max = 90 - A3;

a =  if(
          abs(asin(Dz)*180/PI) - max - v*A3 ,
               1,
               if(
                    h - ( az - floor(az) ) ,
                         0,
                         if(
                              h - ( ceil(az) - az ),
                                   0,
                                   1
                         )
               )
     );

b =  if(
          abs(asin(Dz)*180/PI)- max - v*A3 ,
```

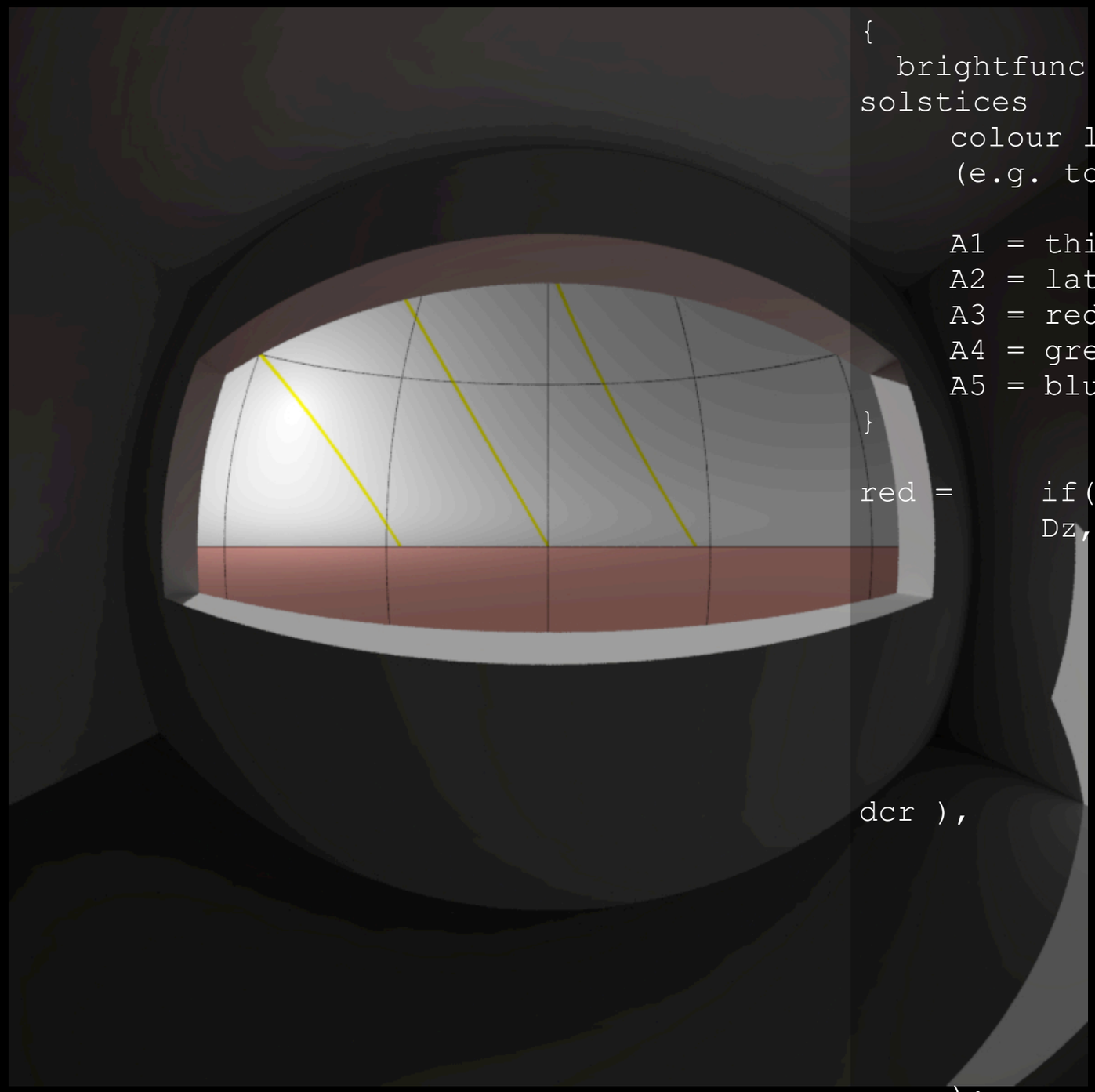
# sunpath



```
{
  brightfunc to generate sunpaths for equinox and
solstices
      colour lines on clear background
      (e.g. to be used in conjunction with a CIE sky)

      A1 = thickness of lines
      A2 = latitude
      A3 = red
      A4 = green
      A5 = blue
}

red =     if(
          Dz,
                    if(
                         25 - abs(dec),
                              if(
                                   h - ( dcr - floor(dcr) ) ,
                                        A3,
                                        if(
                                             h - ( ceil(dcr) -

dcr ),
                                                  A3,
                                                  1
                                        )
                              ) ,
                              1
                    ) ,
                    1
          );


green =   if(
```

Arup | Lighting

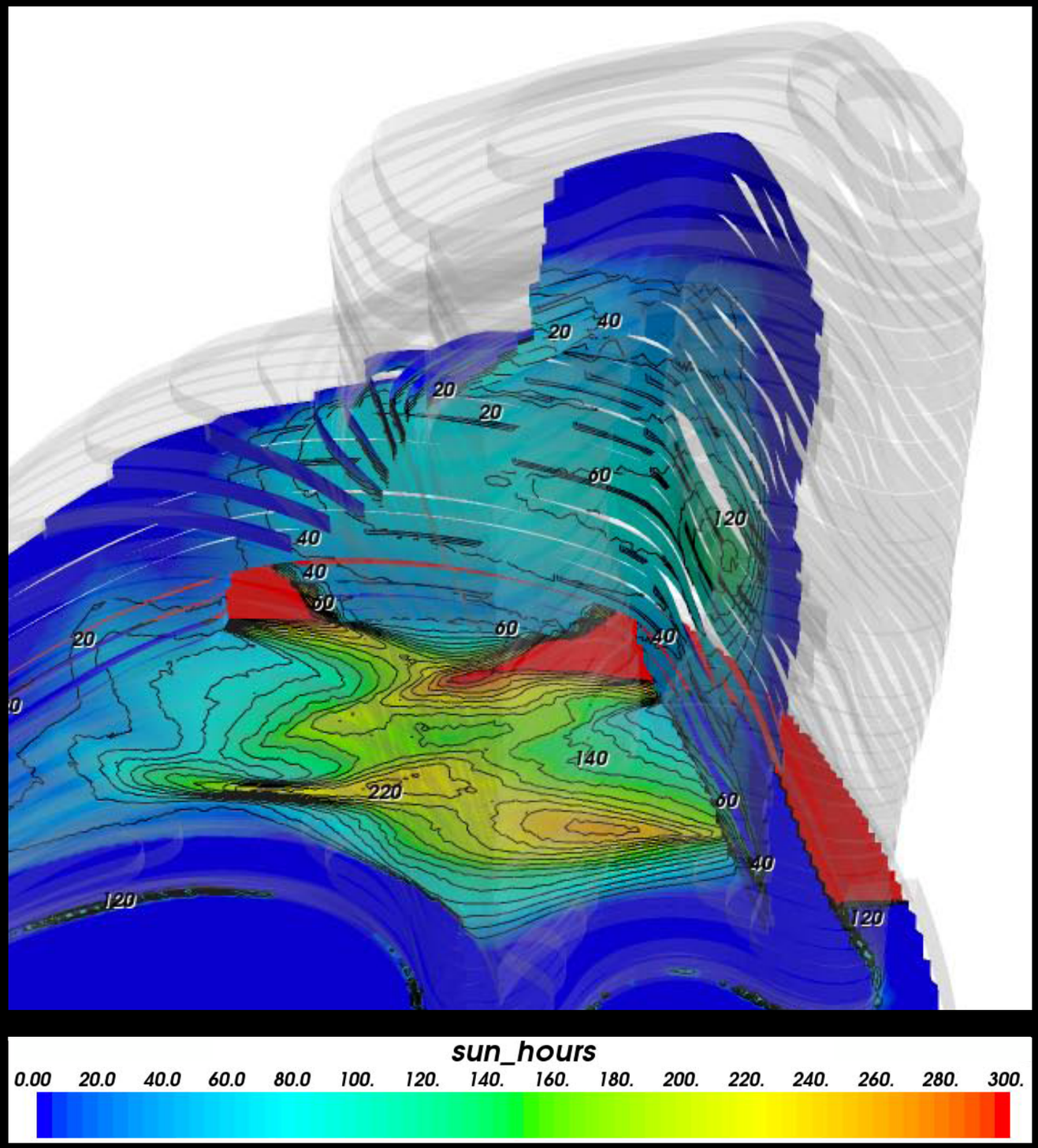
# sunpath



```
{
  brightfunc to generate sunpaths for equinox and
solstices
     colour lines on clear background
     (e.g. to be used in conjunction with a CIE sky)

     A1 = thickness of lines
     A2 = latitude
     A3 = red
     A4 = green
     A5 = blue
}

red =     if(
          Dz,
                    if(
                         25 - abs(dec),
                              if(
                                   h - ( dcr - floor(dcr) ) ,
                                        A3,
                                        if(
                                             h - ( ceil(dcr) -

dcr ),                                                  A3,
                                                        1
                                        )
                              ) ,
                              1
                    ) ,
                    1
          );

green =   if(
```

Arup | Lighting

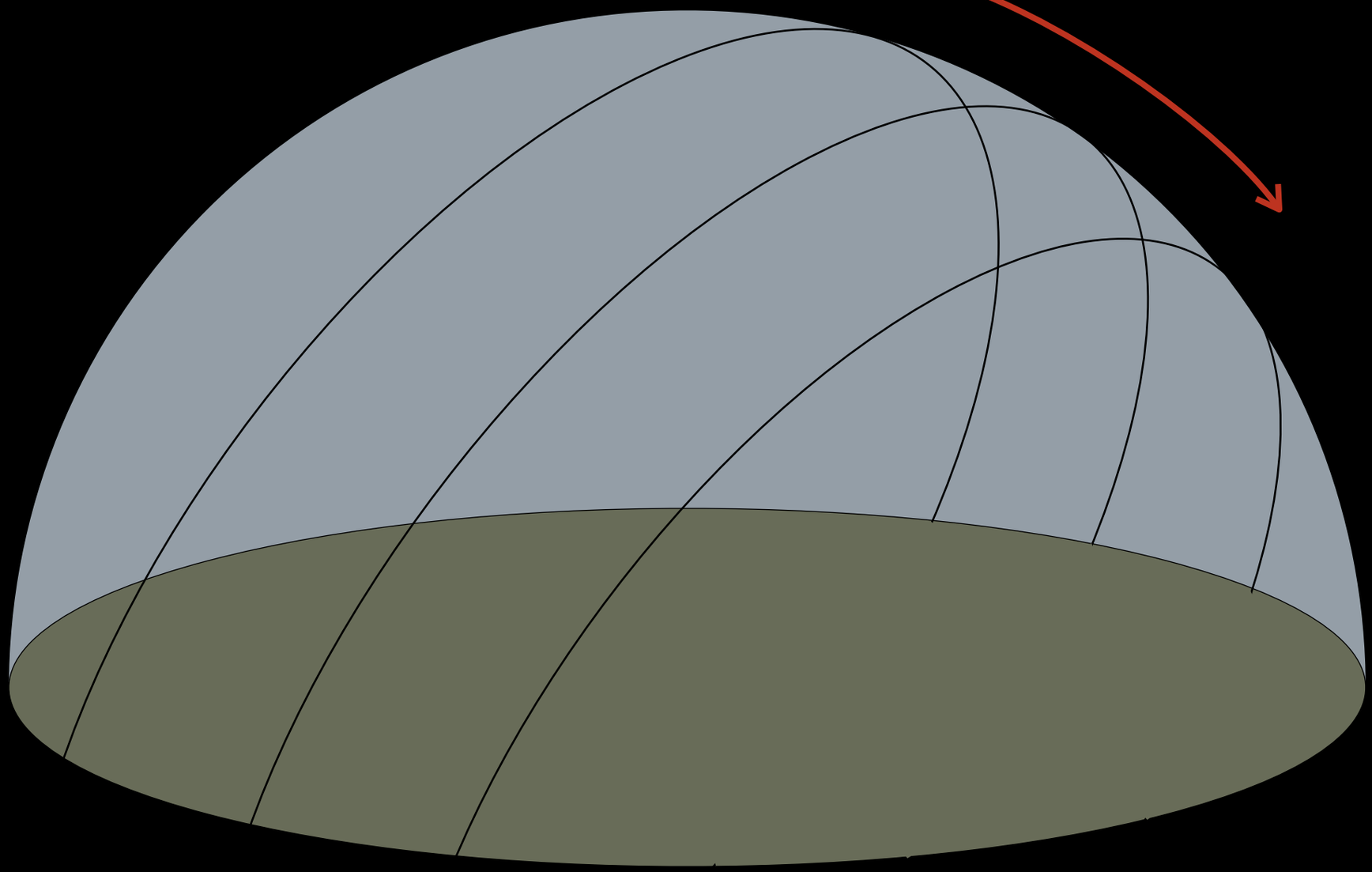
# sunhours




Arup | Lighting

# sunhours

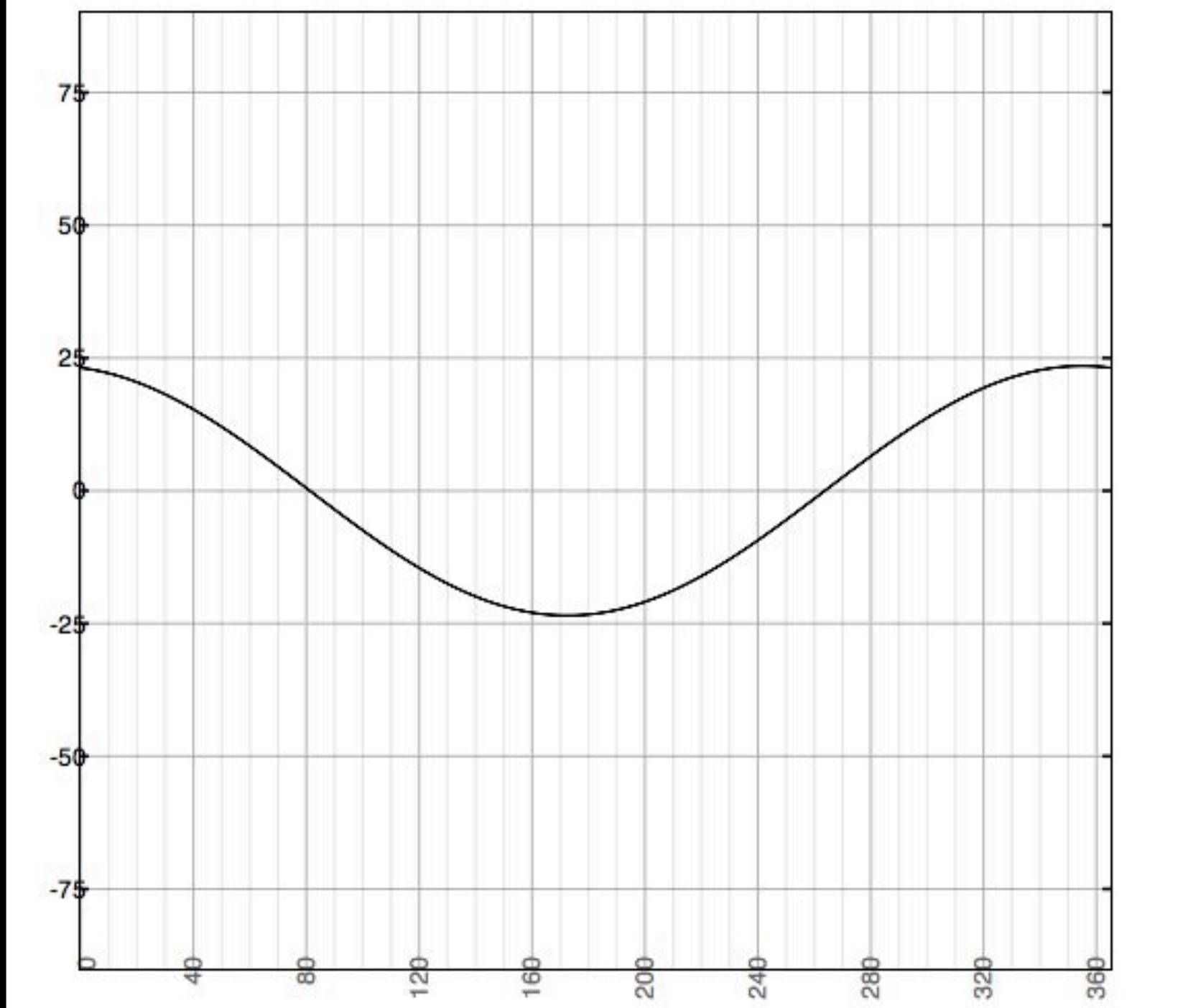
declination



Arup | Lighting

# sunhours

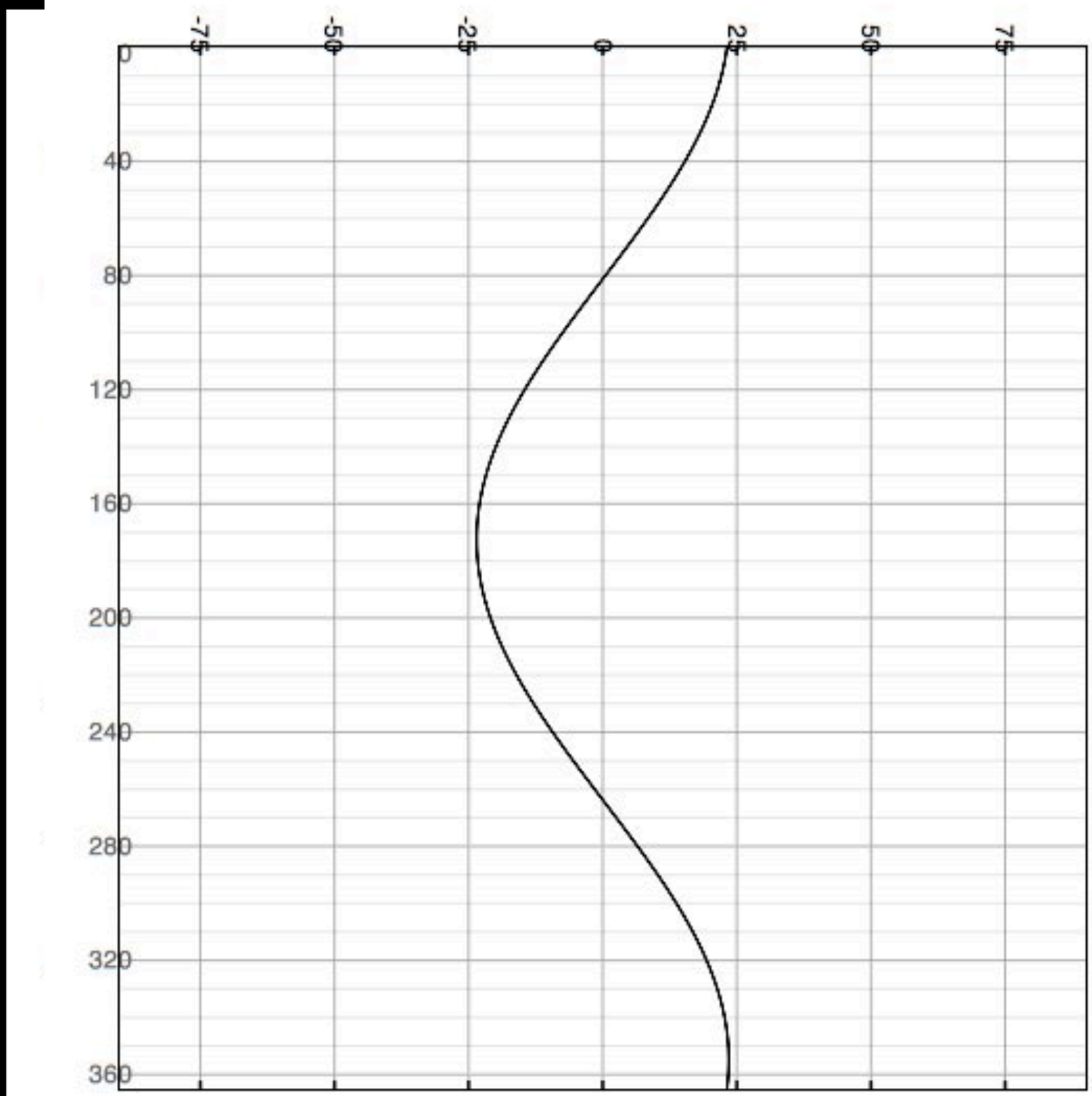
dec. as a function of DN



# sunhours

dec. as a function of DN

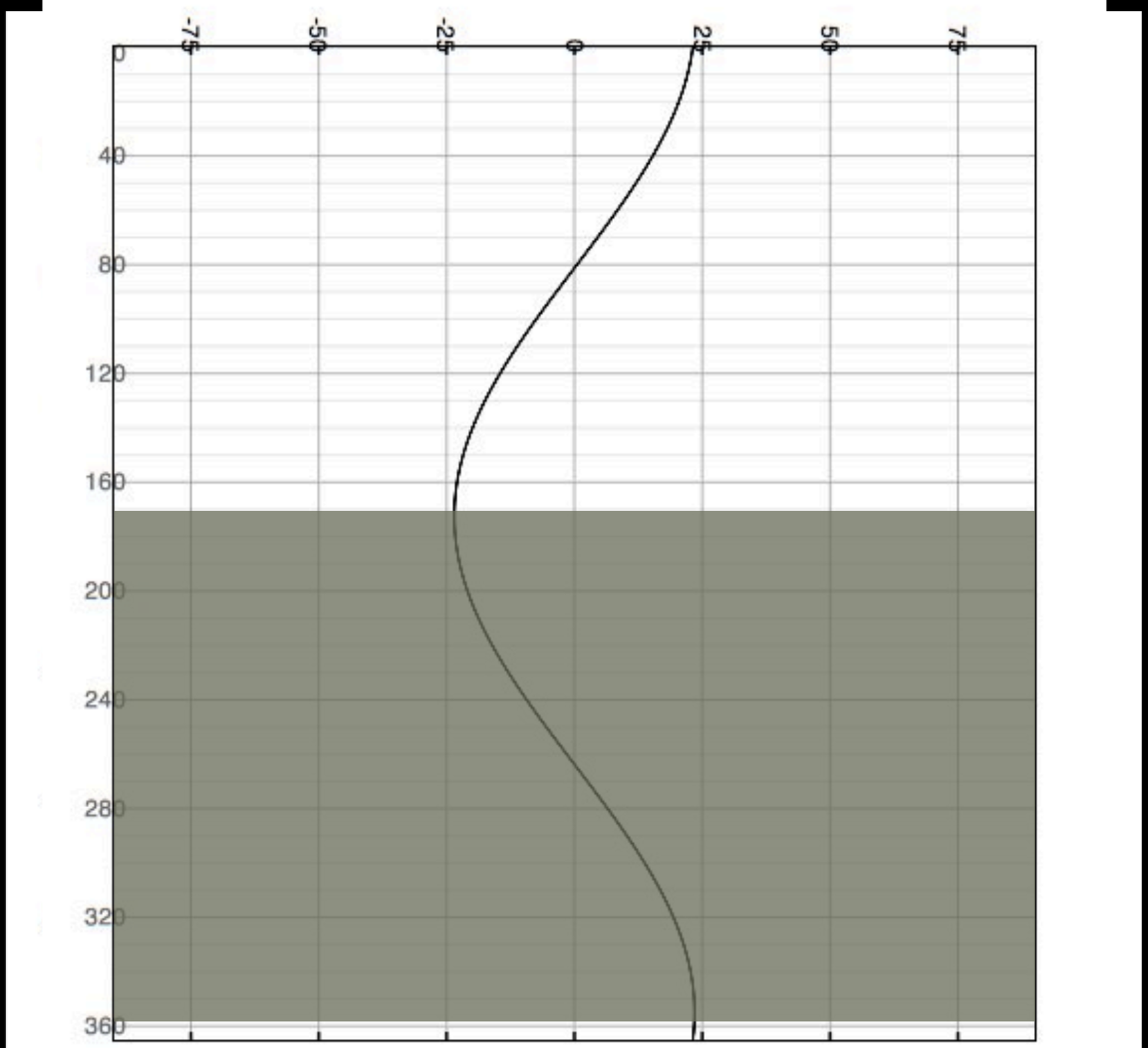
the inverse function:
DN as a function of dec.



# sunhours

dec. as a function of DN

the inverse function:
DN as a function of dec.
(only half year)

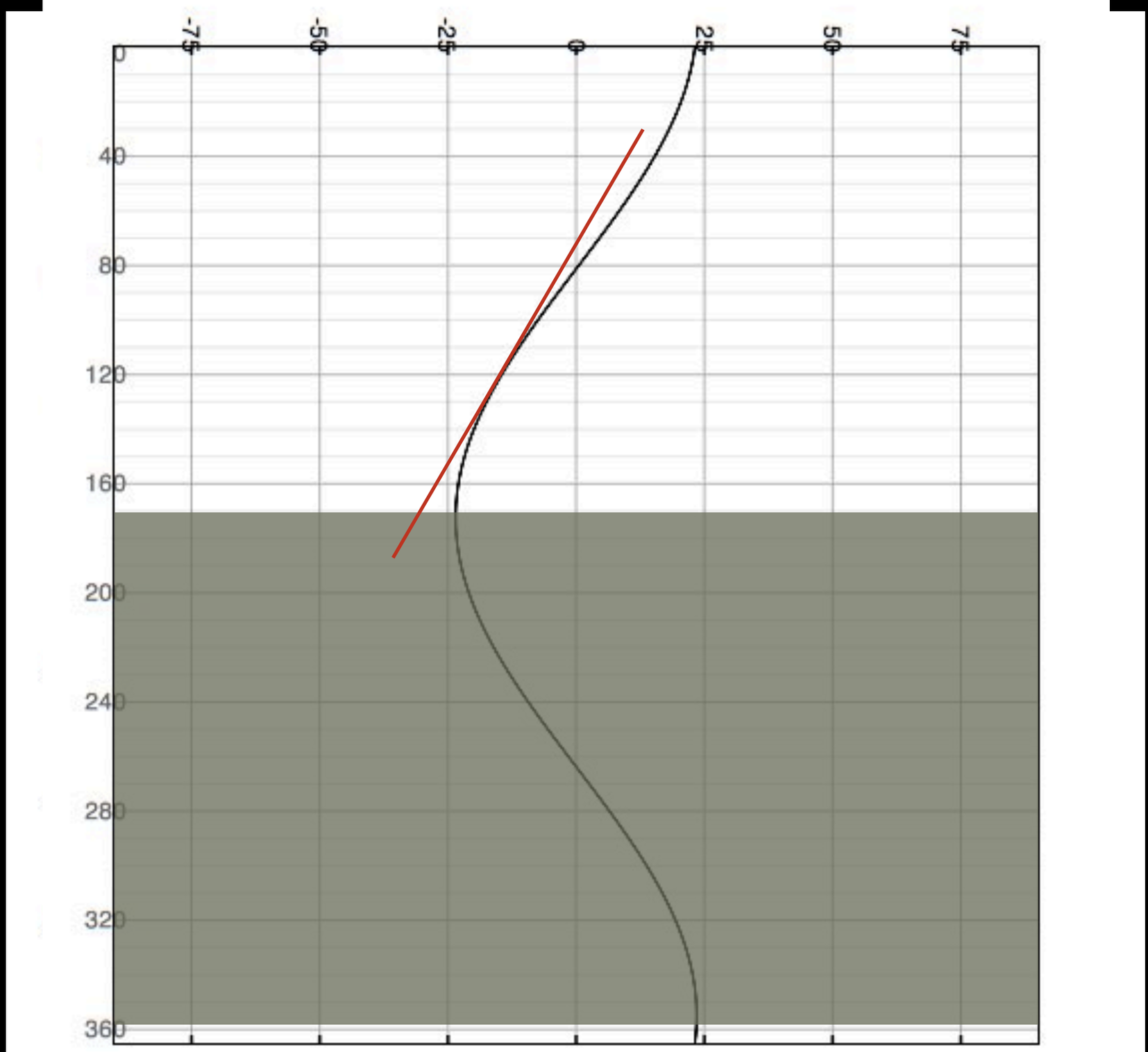


# sunhours

dec. as a function of DN

the inverse function:
DN as a function of dec.
(only half year)

derivative = number of
days per unit angle
(x2 for the full year)

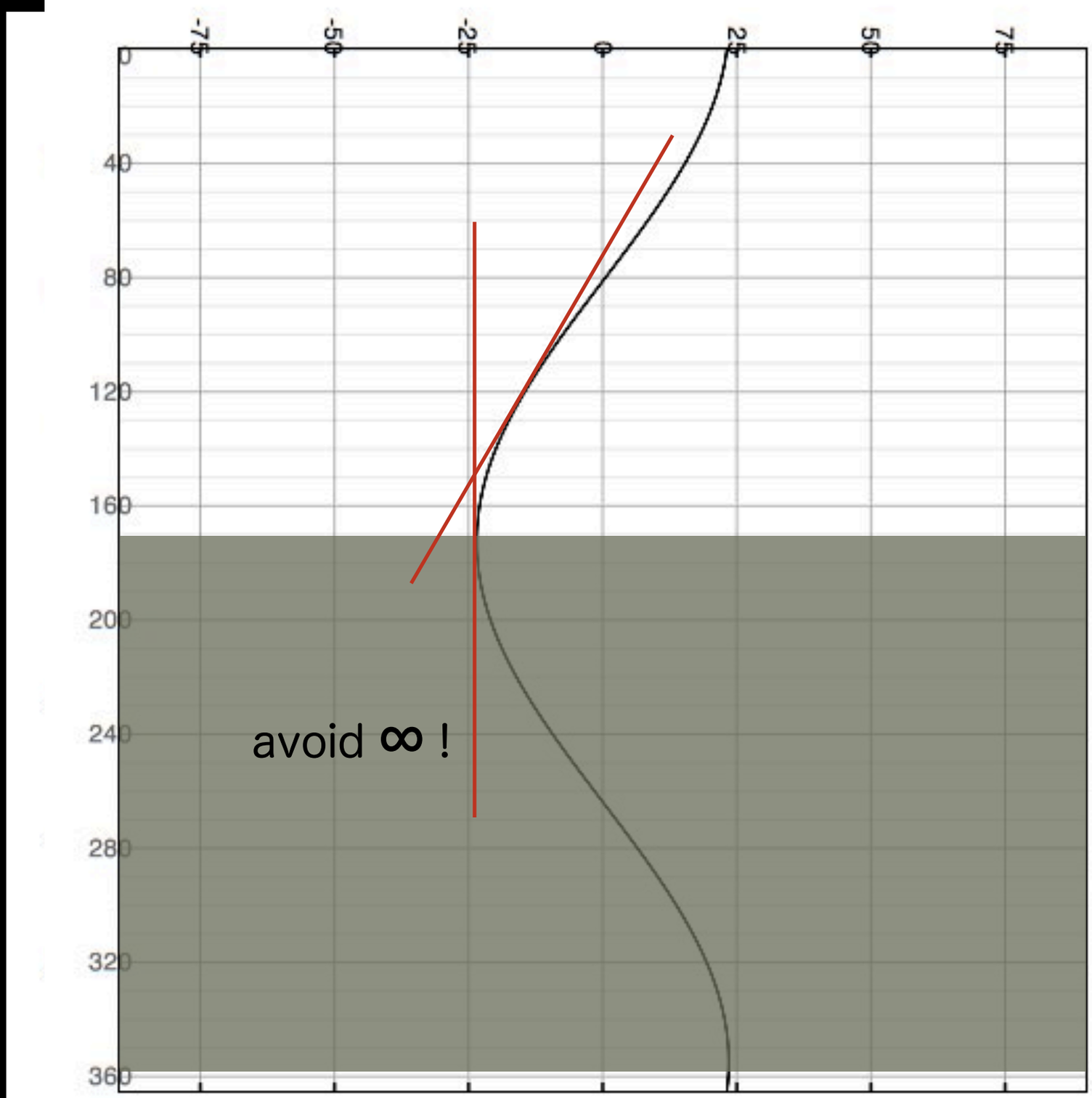


# sunhours

dec. as a function of DN

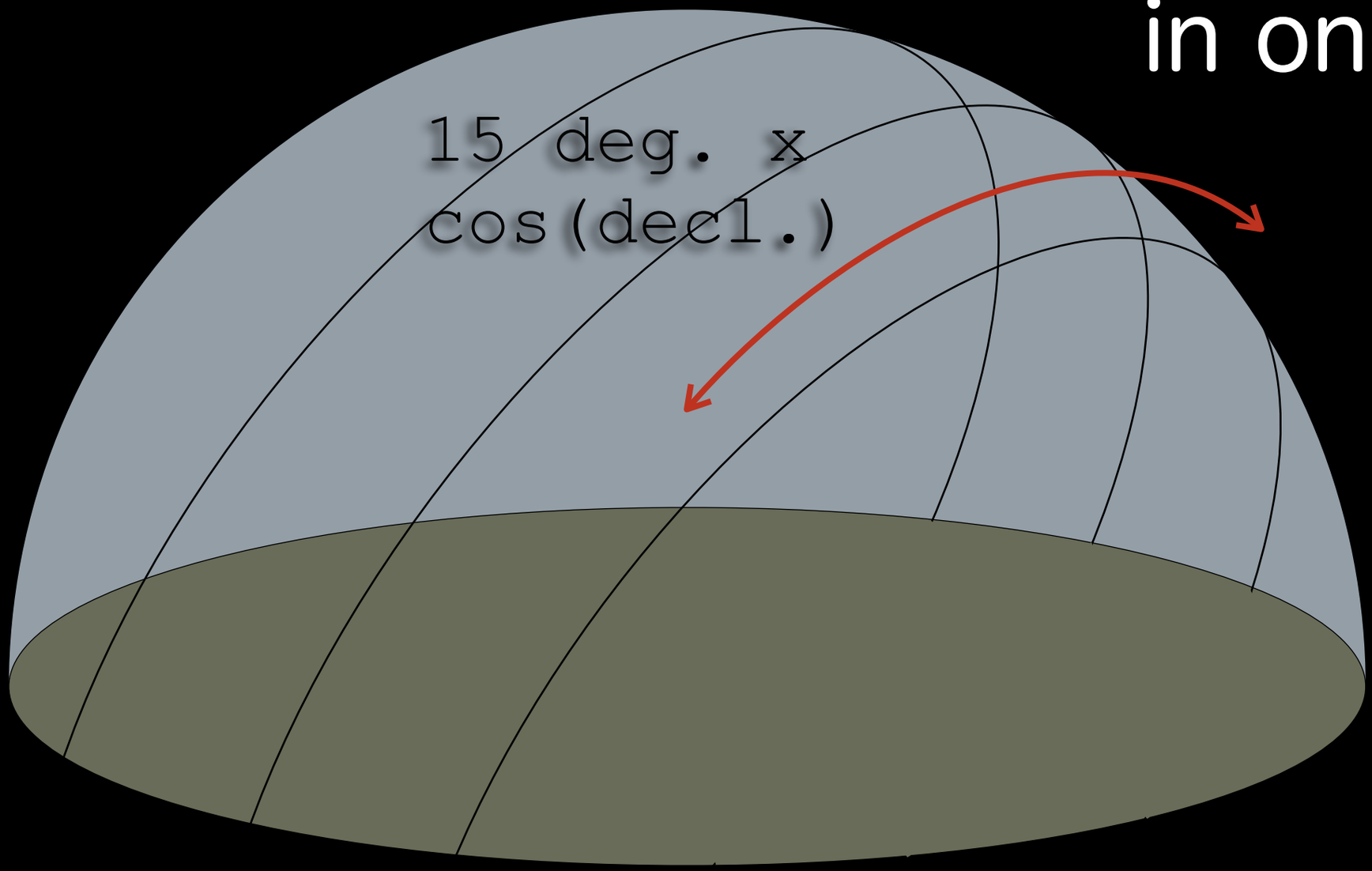
the inverse function:
DN as a function of dec.
(only half year)

derivative = number of
days per unit angle
(x2 for the full year)

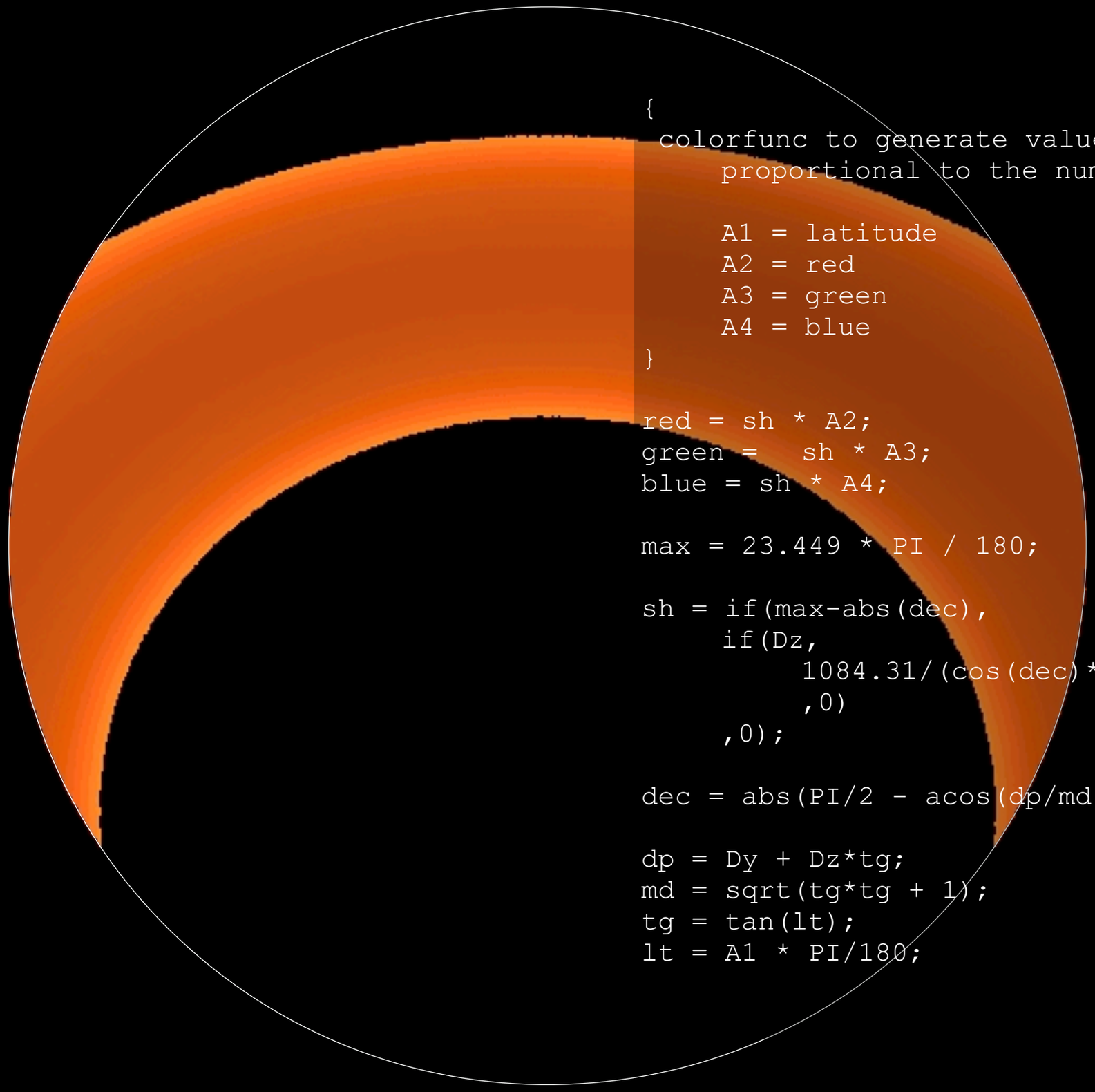


# sunhours

movement
in one day

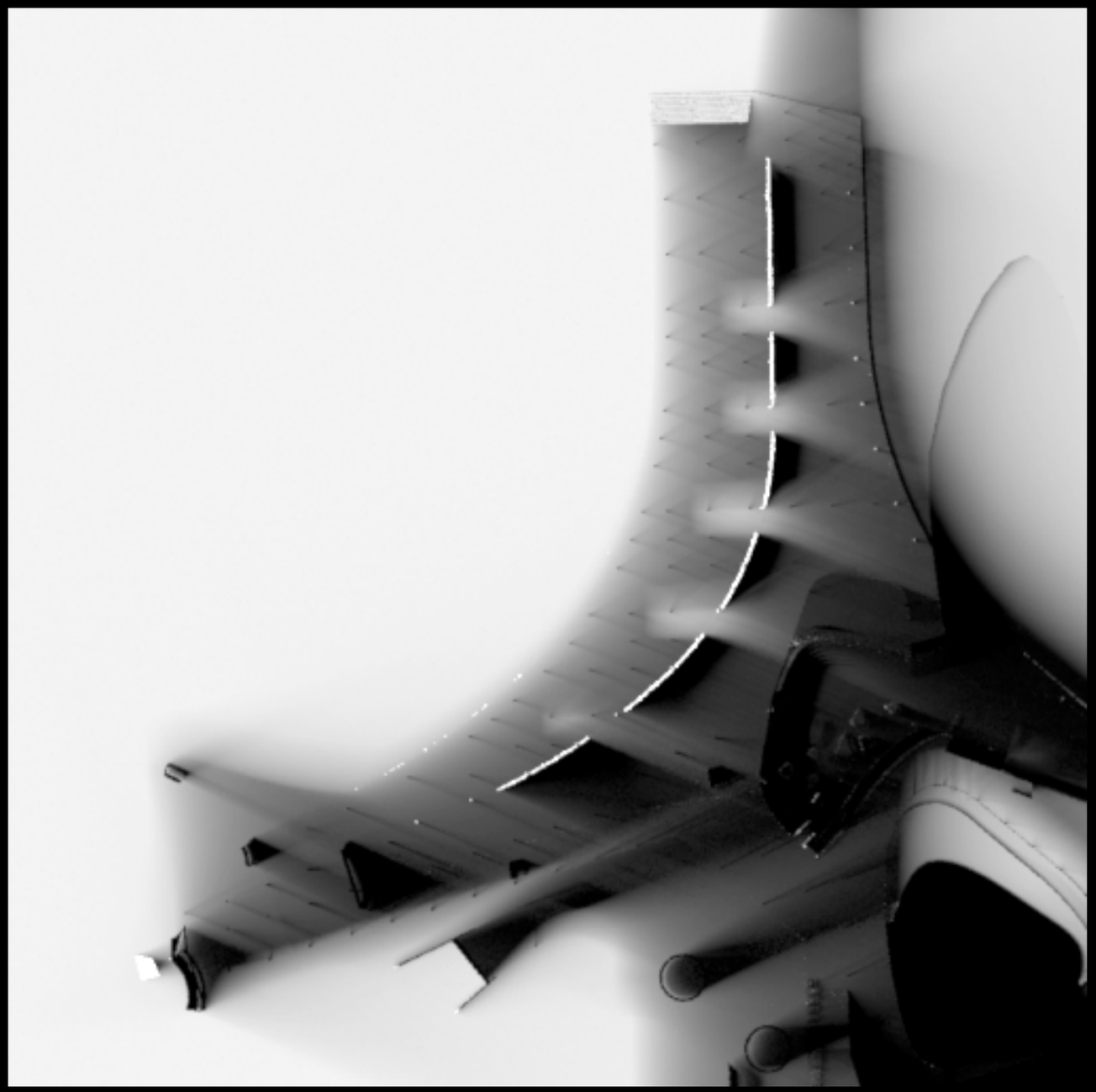


# sunhours



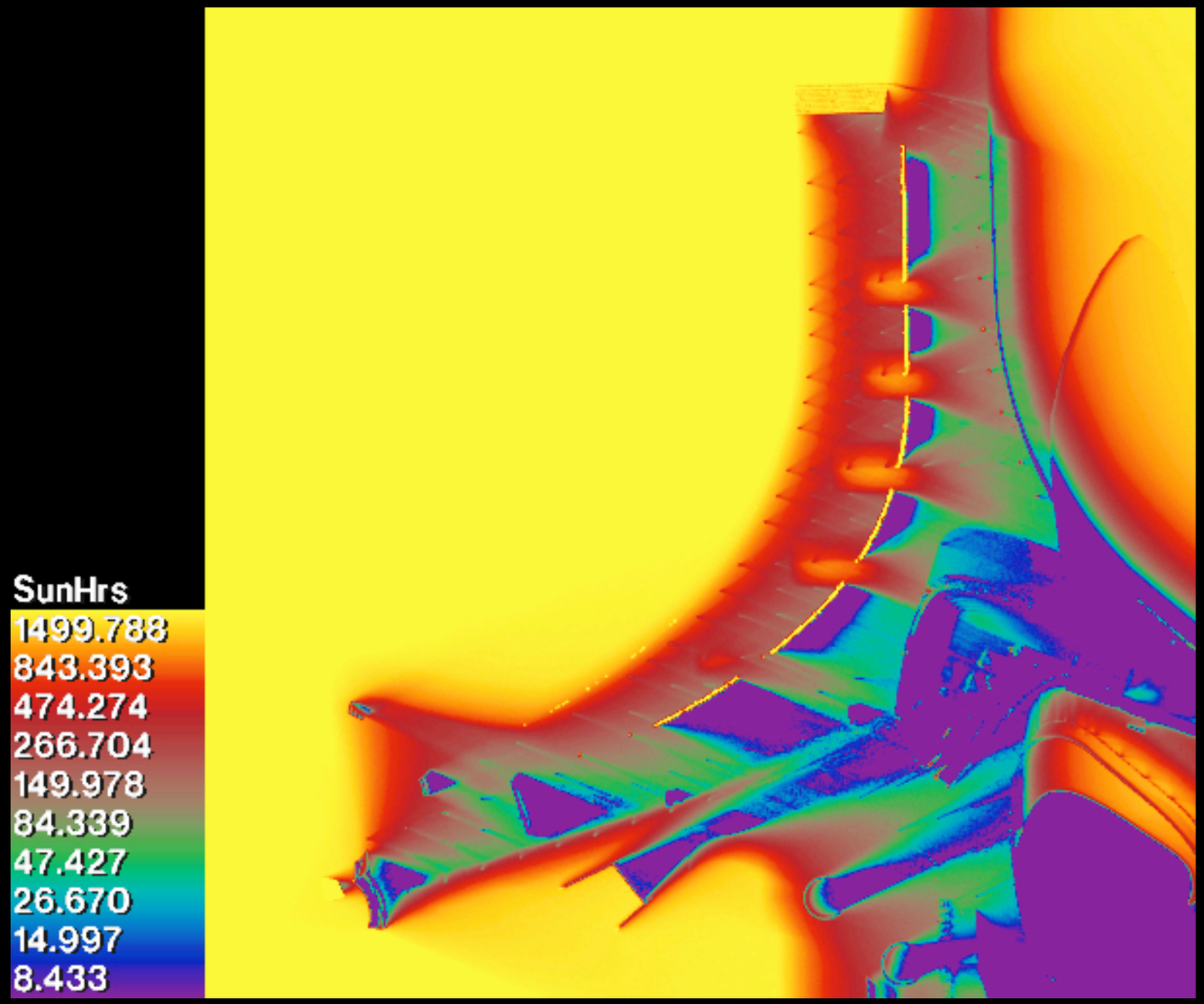
```
{
 colorfunc to generate values for the sky
     proportional to the number of sunhours

     A1 = latitude
     A2 = red
     A3 = green
     A4 = blue
}

red = sh * A2;
green =   sh * A3;
blue = sh * A4;

max = 23.449 * PI / 180;

sh = if(max-abs(dec),
     if(Dz,
          1084.31/(cos(dec)*sqrt(1-5.9698*dec*dec))
          ,0)
     ,0);

dec = abs(PI/2 - acos(dp/md));

dp = Dy + Dz*tg;
md = sqrt(tg*tg + 1);
tg = tan(lt);
lt = A1 * PI/180;
```

# sunhours



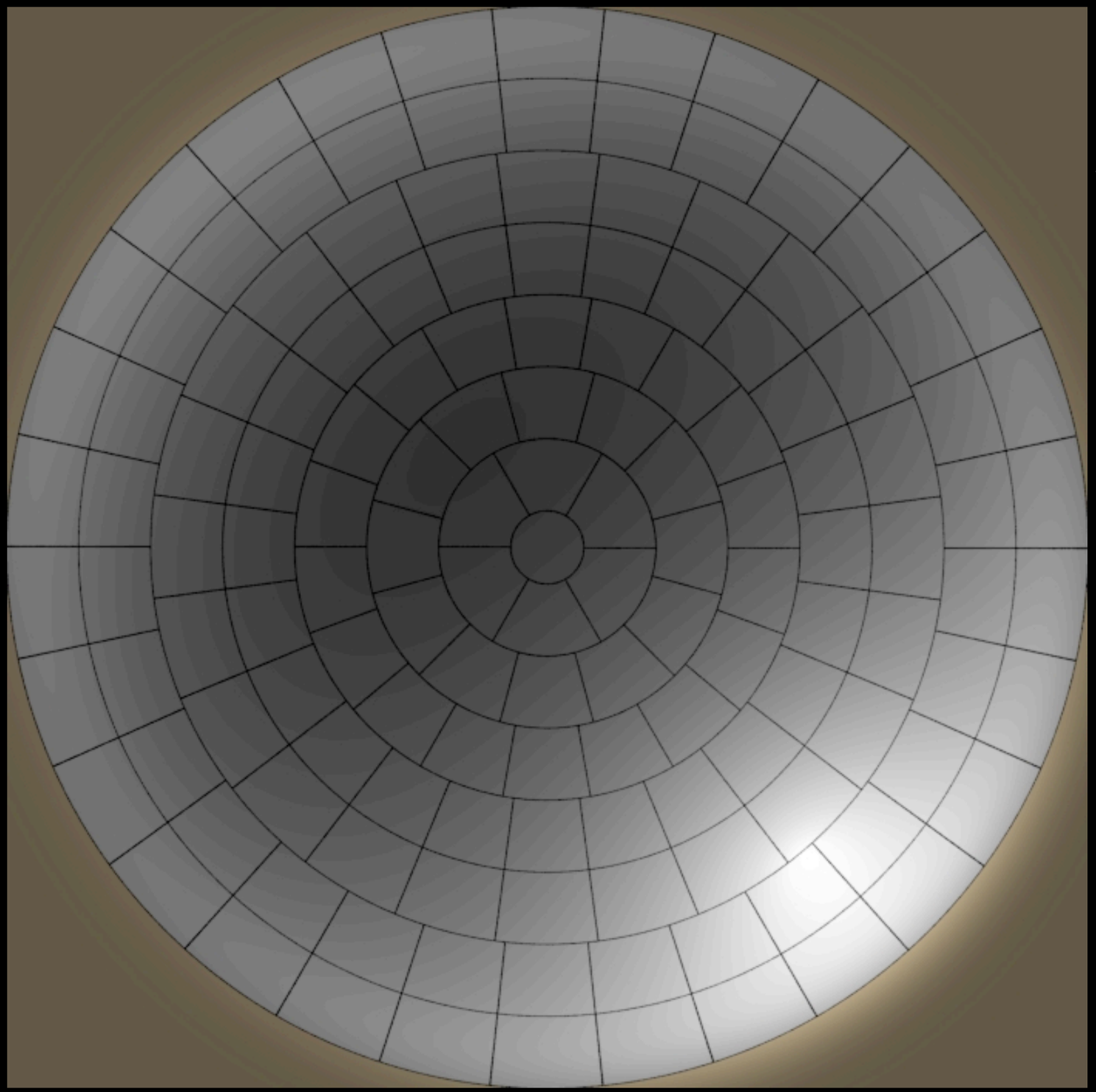
sunhours is not affected by cosine law => needs workaround
(e.g. compensating for horizontal surfaces)

# sunhours




sunhours is not affected by cosine law => needs workaround
(e.g. compensating for horizontal surfaces)

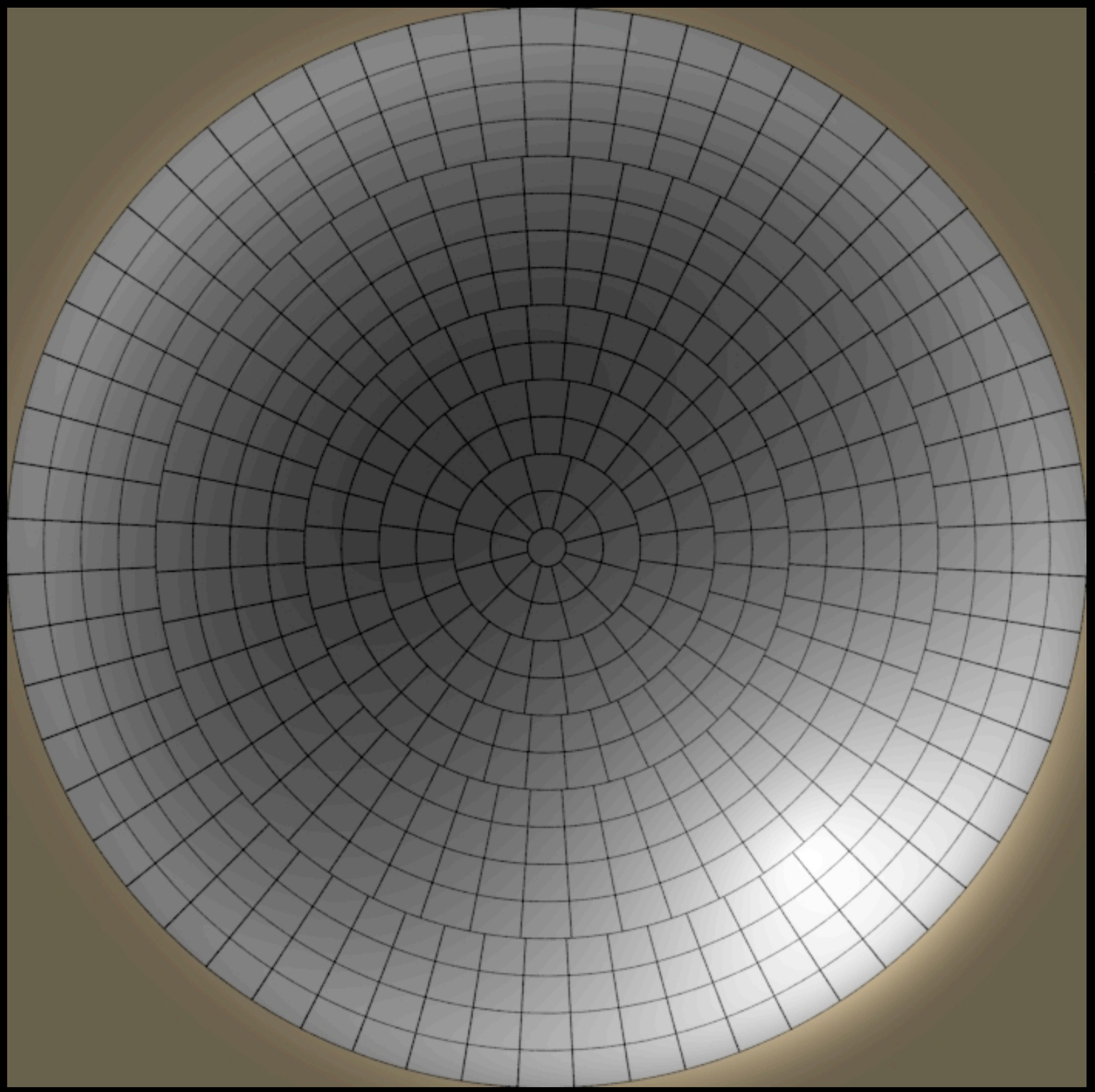
# tregenza.cal



N: 145
~12 deg.

Arup | Lighting

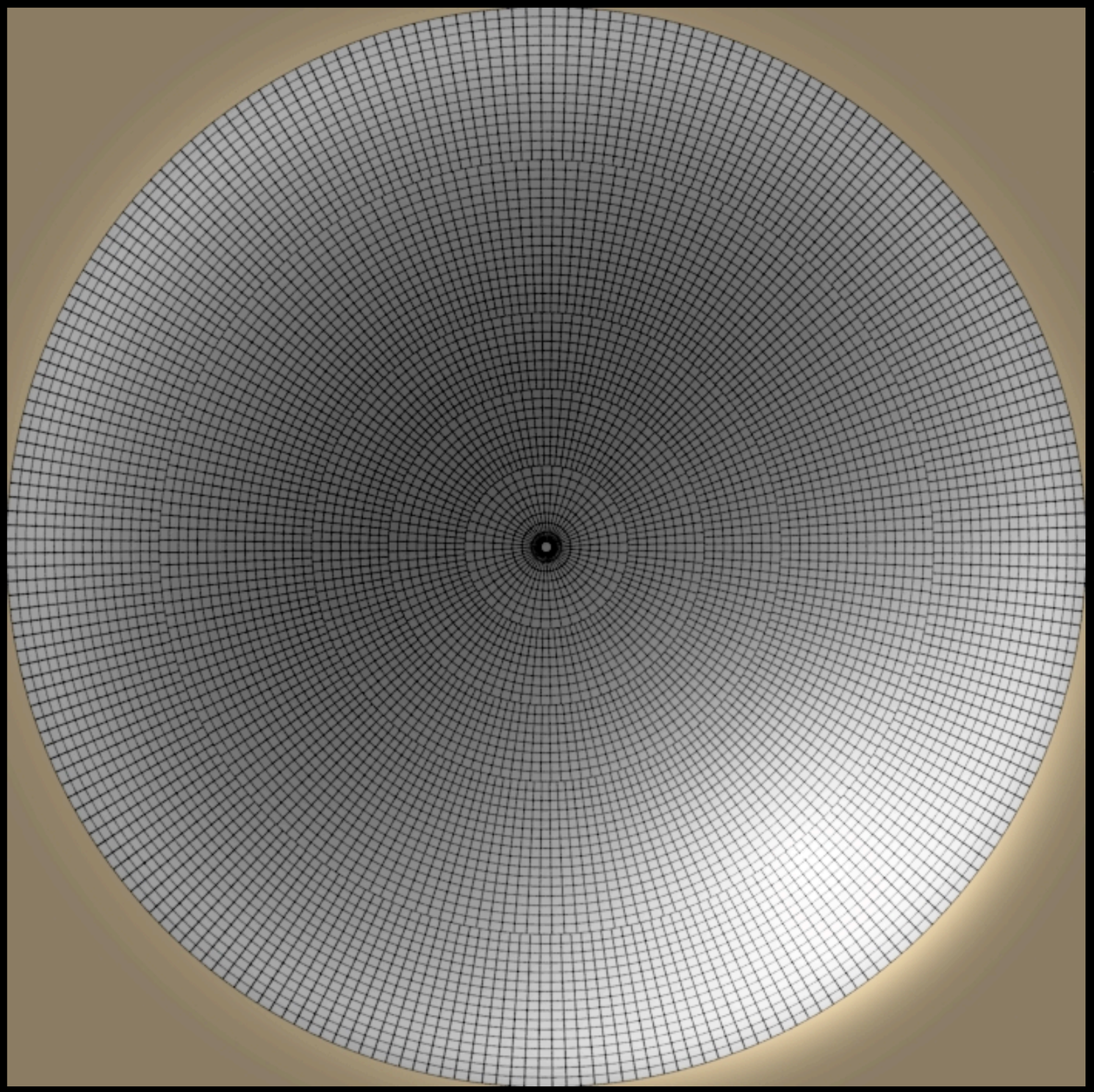
# reinhart.cal



MF:2
N: 580
~6 deg.

can produce any number multiple of 145 and a square

# reinhart.cal

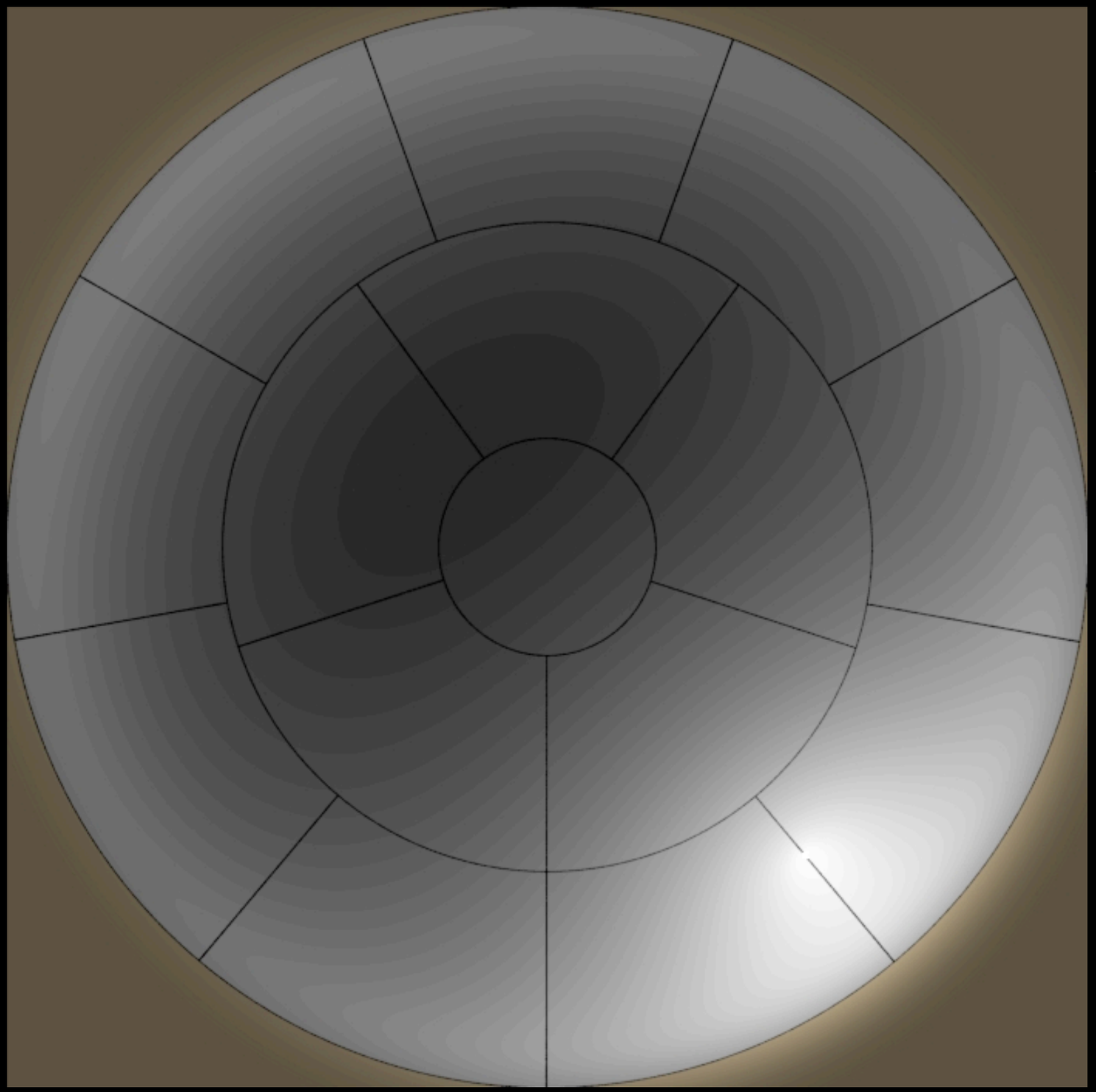


MF:2
N: 580
~6 deg.

can produce any number multiple of 145 and a square

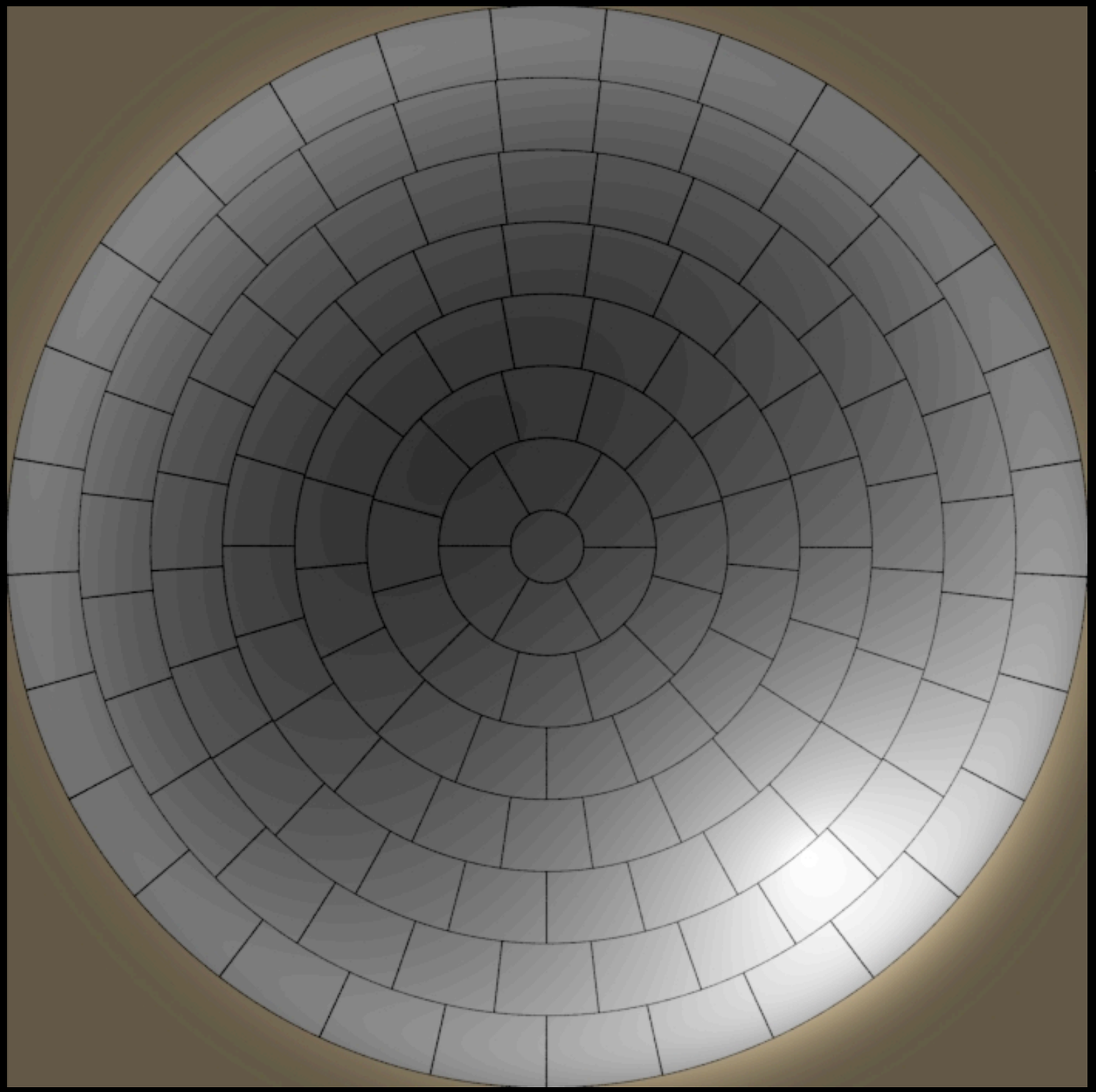
MF:8
N: 9280
~1.5 deg.

# generic



N: 16
~36 deg.

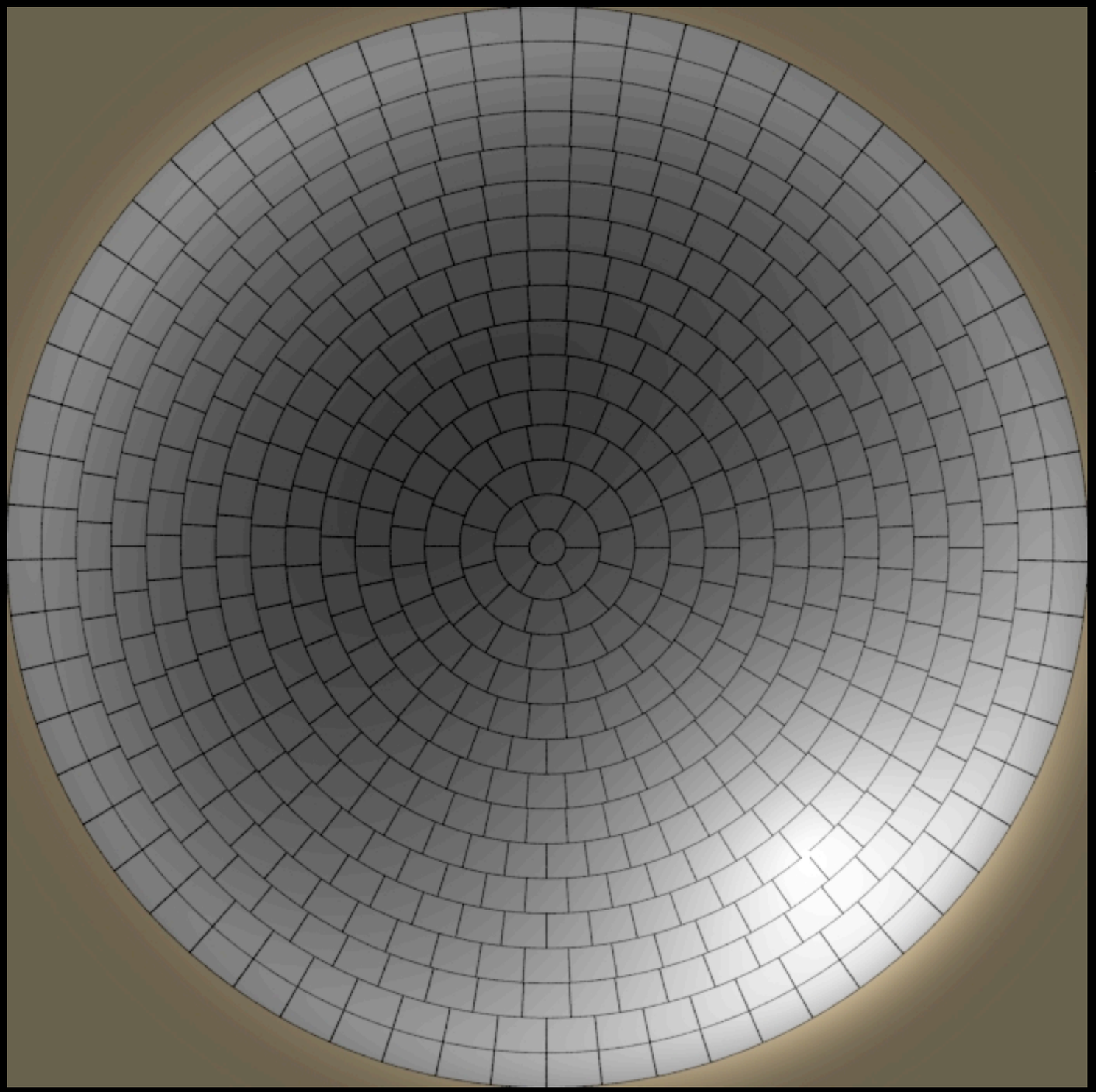
# generic



N: 16
~36 deg.

N: 141
~12 deg.

# generic

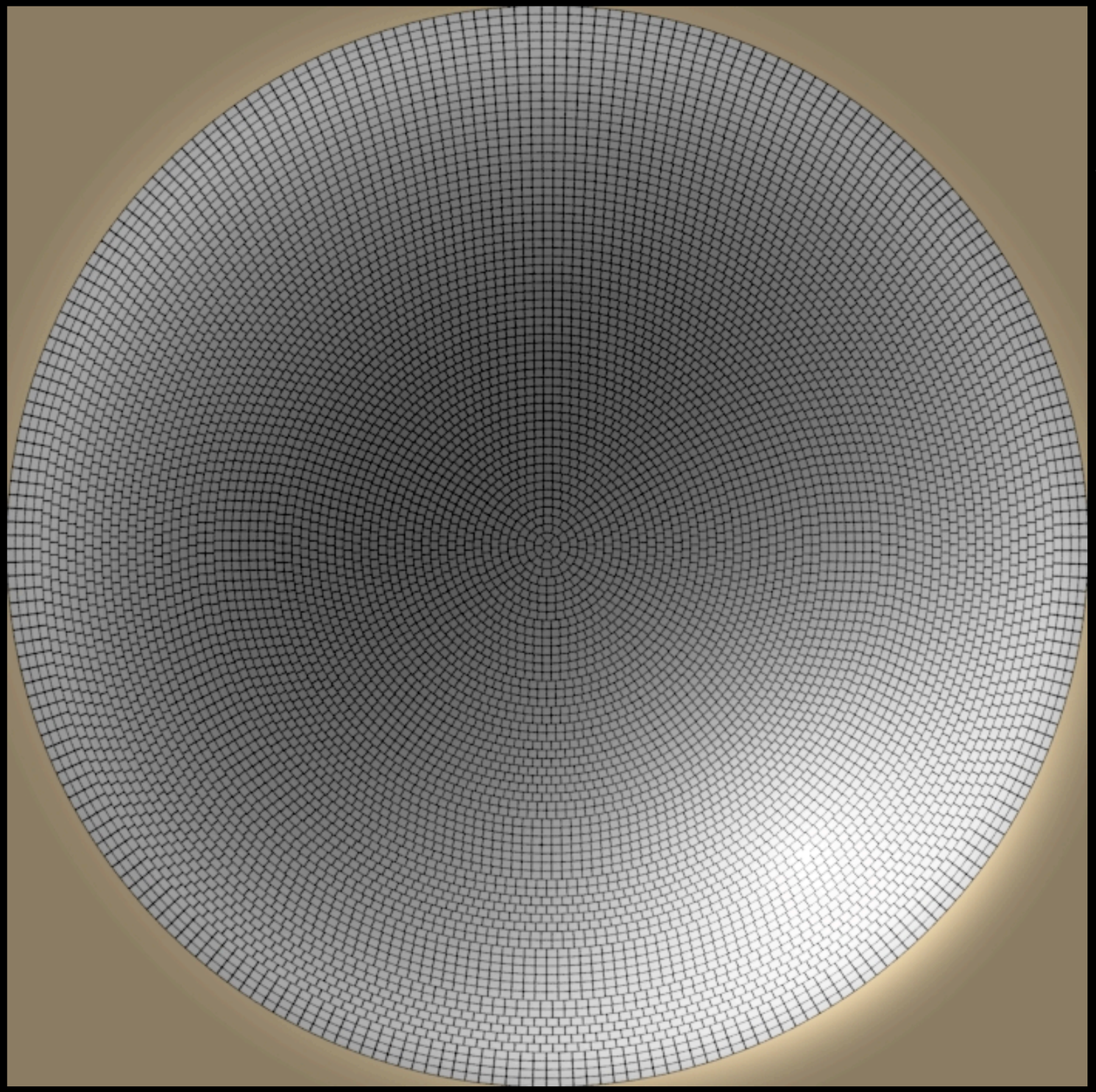


N: 16
~36 deg.

N: 141
~12 deg.

N: 607
~5.8 deg.

# generic



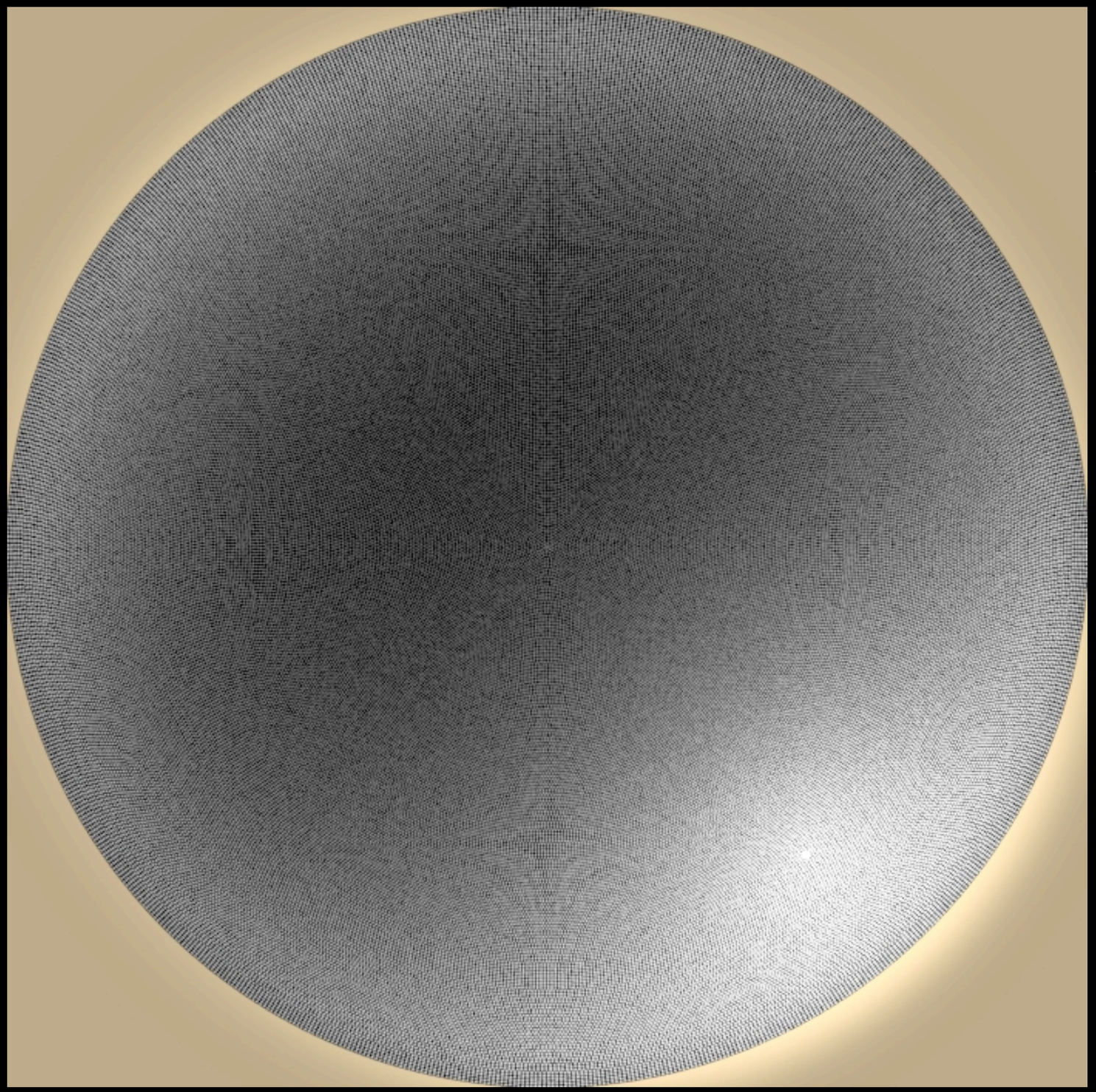
N: 16
~36 deg.

N: 141
~12 deg.

N: 607
~5.8 deg.

N: 9917
~1.44 deg.

# generic



N: 16
~36 deg.

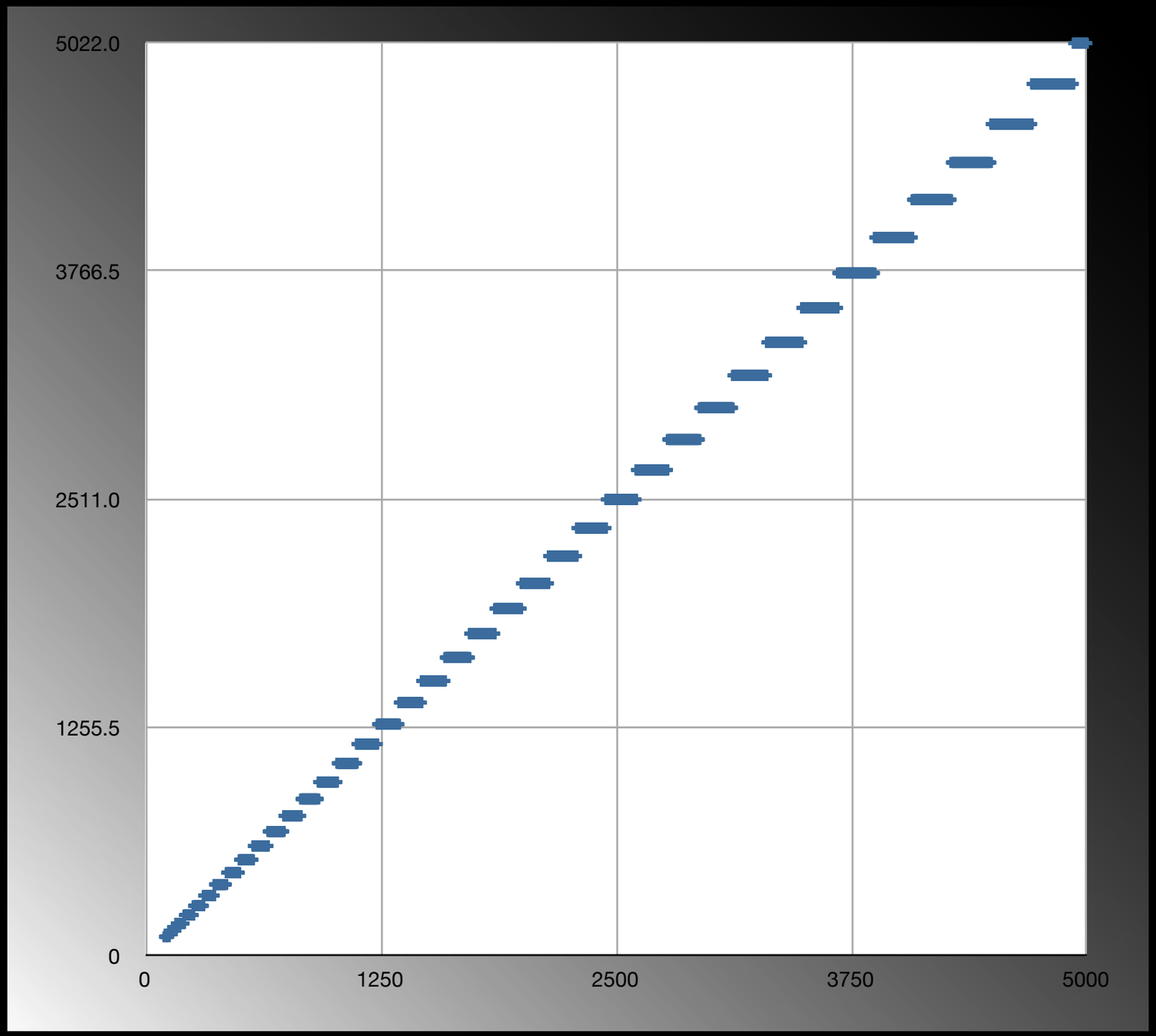
N: 141
~12 deg.

N: 607
~5.8 deg.

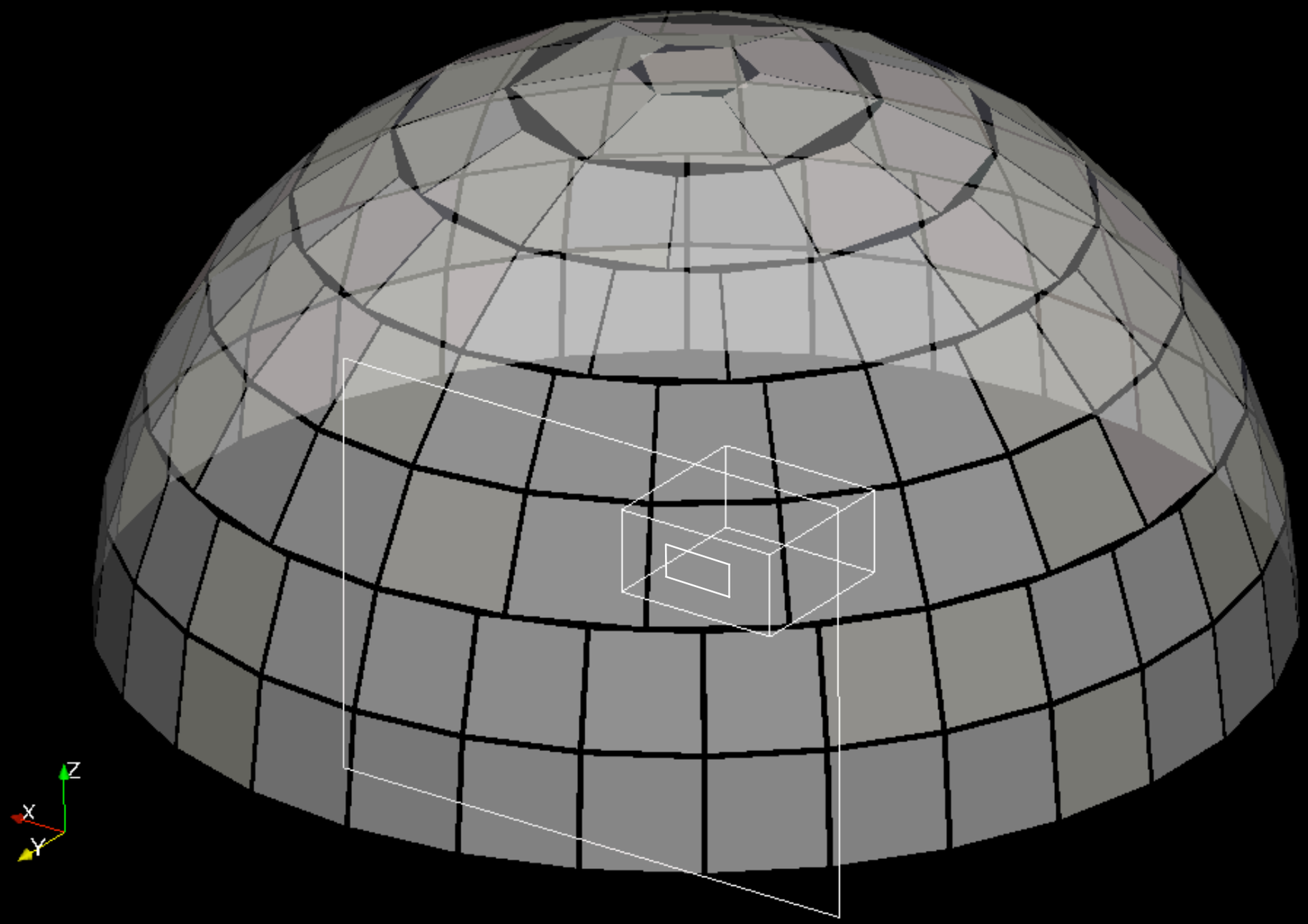
N: 9917
~1.44 deg.

N: 100236
~0.45 deg.

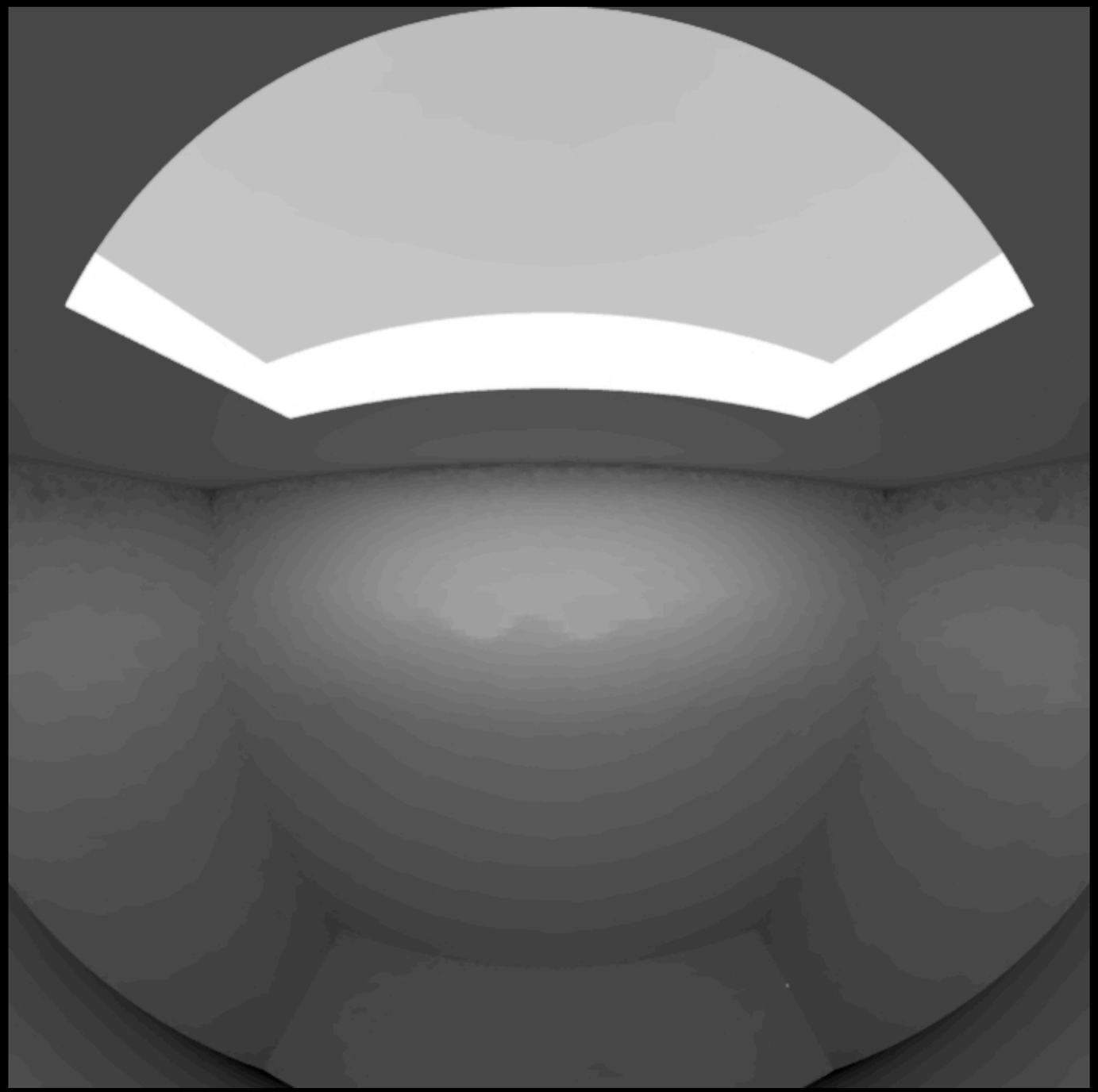
# generic



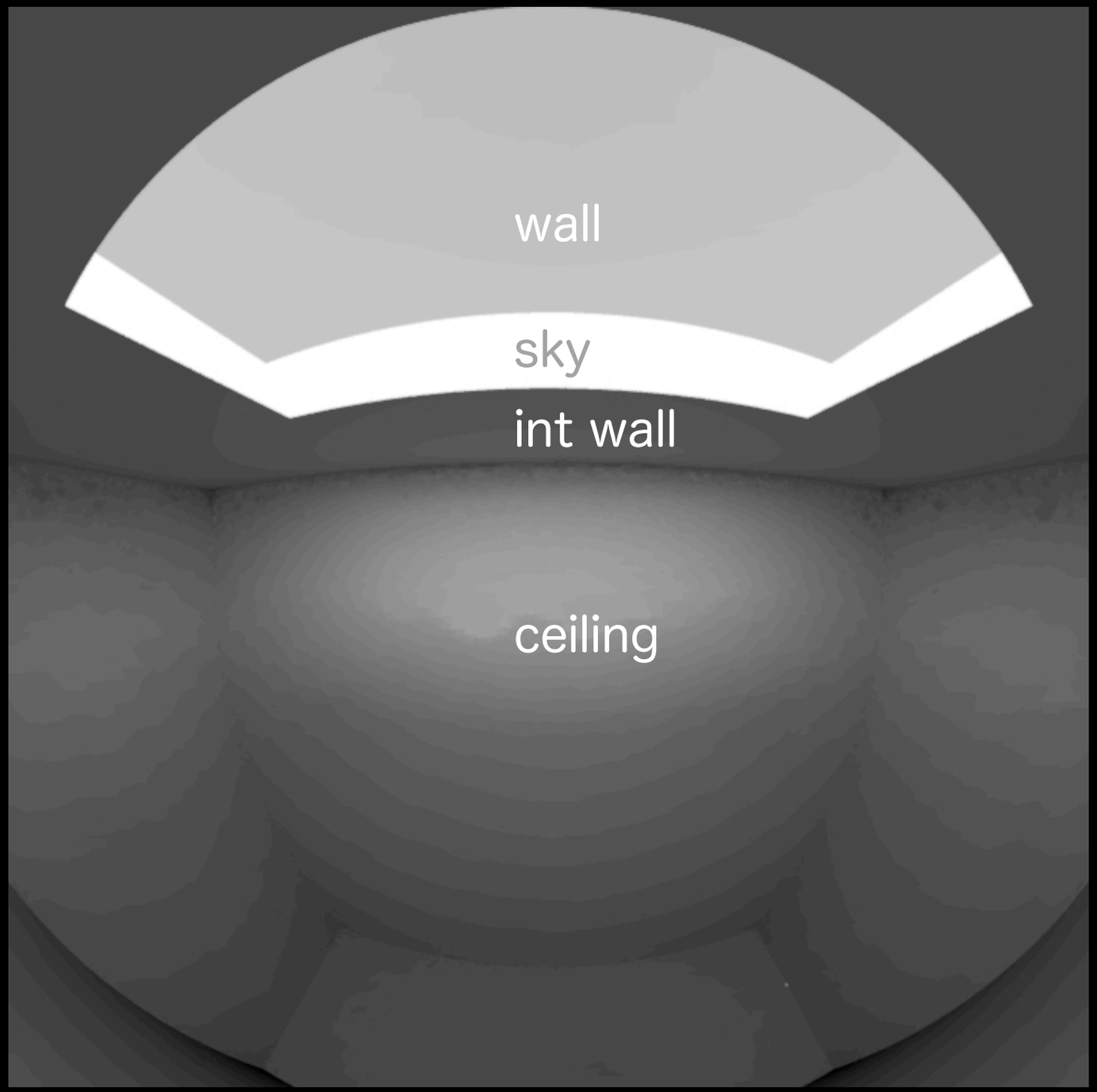
# sky coeficients



# sky coeficients

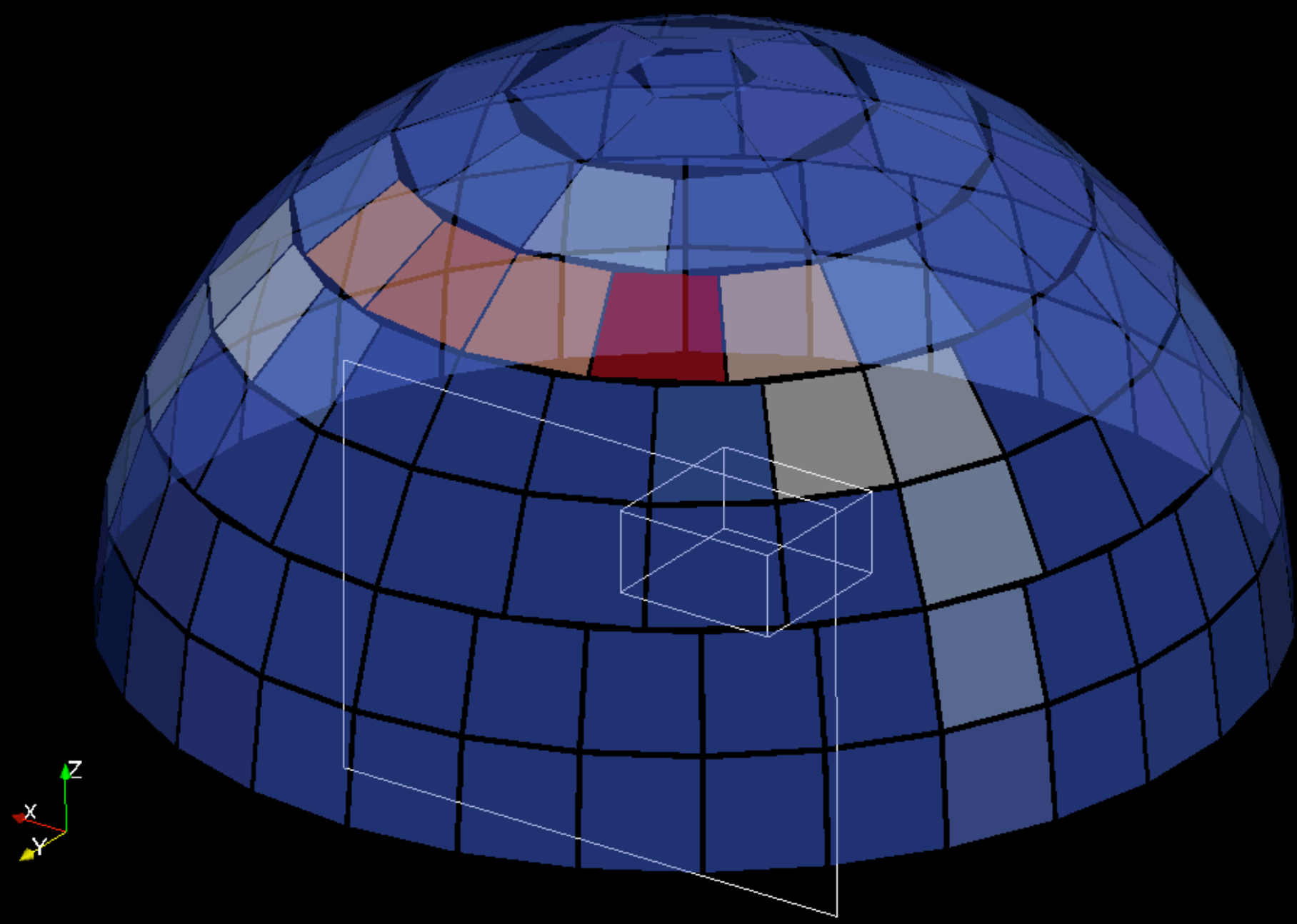


# sky coeficients

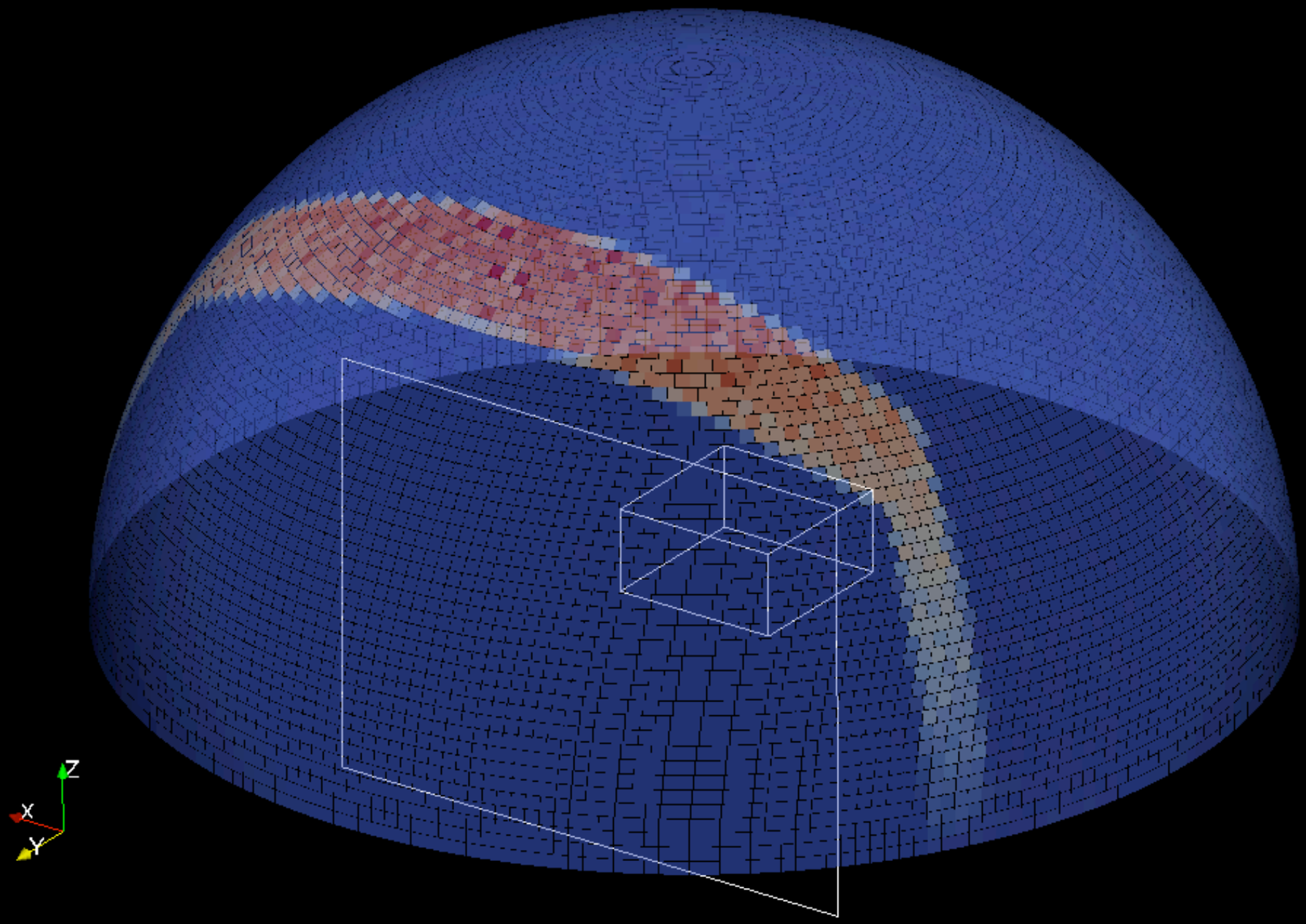



view from
calculation point

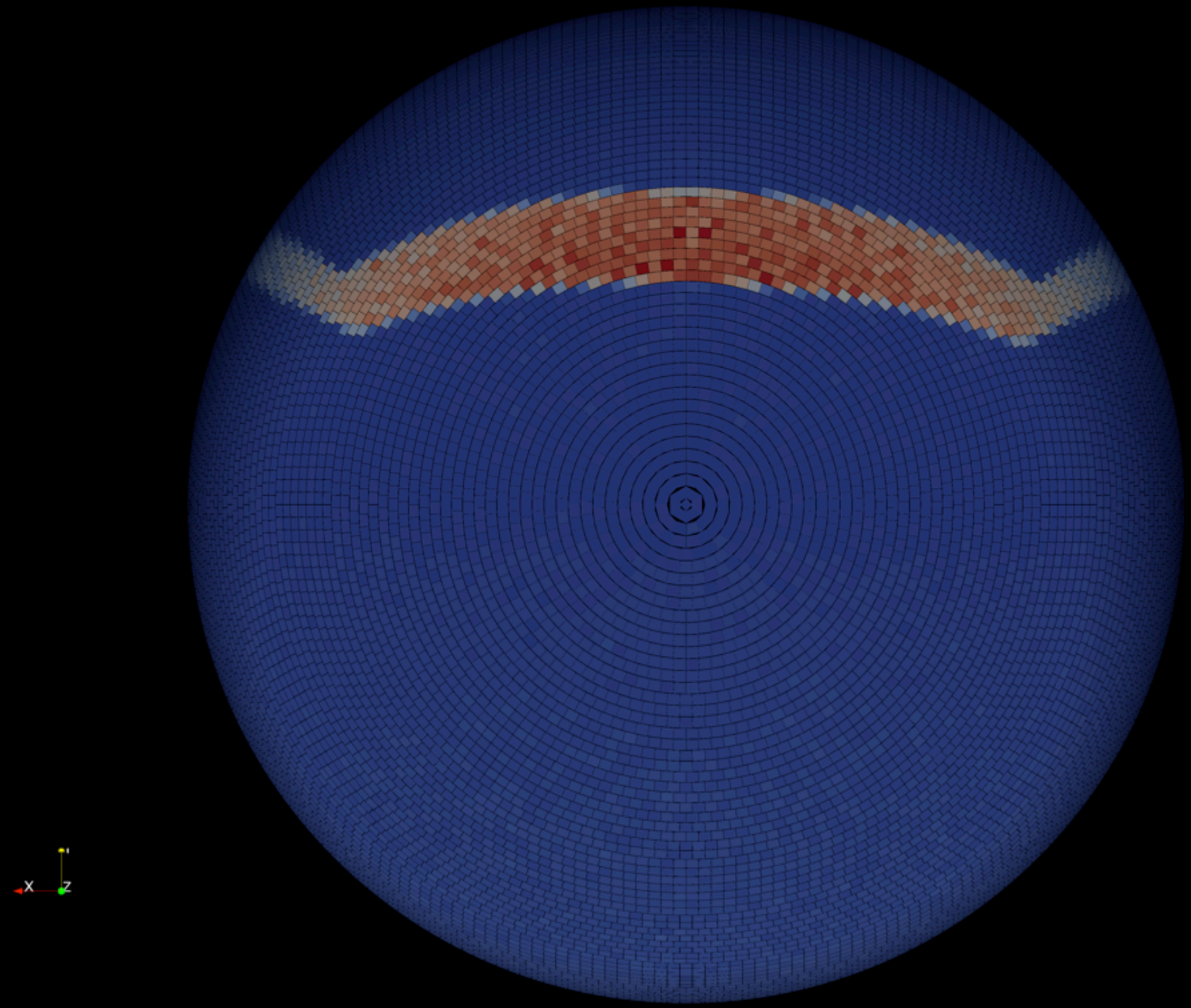
# sky coeficients



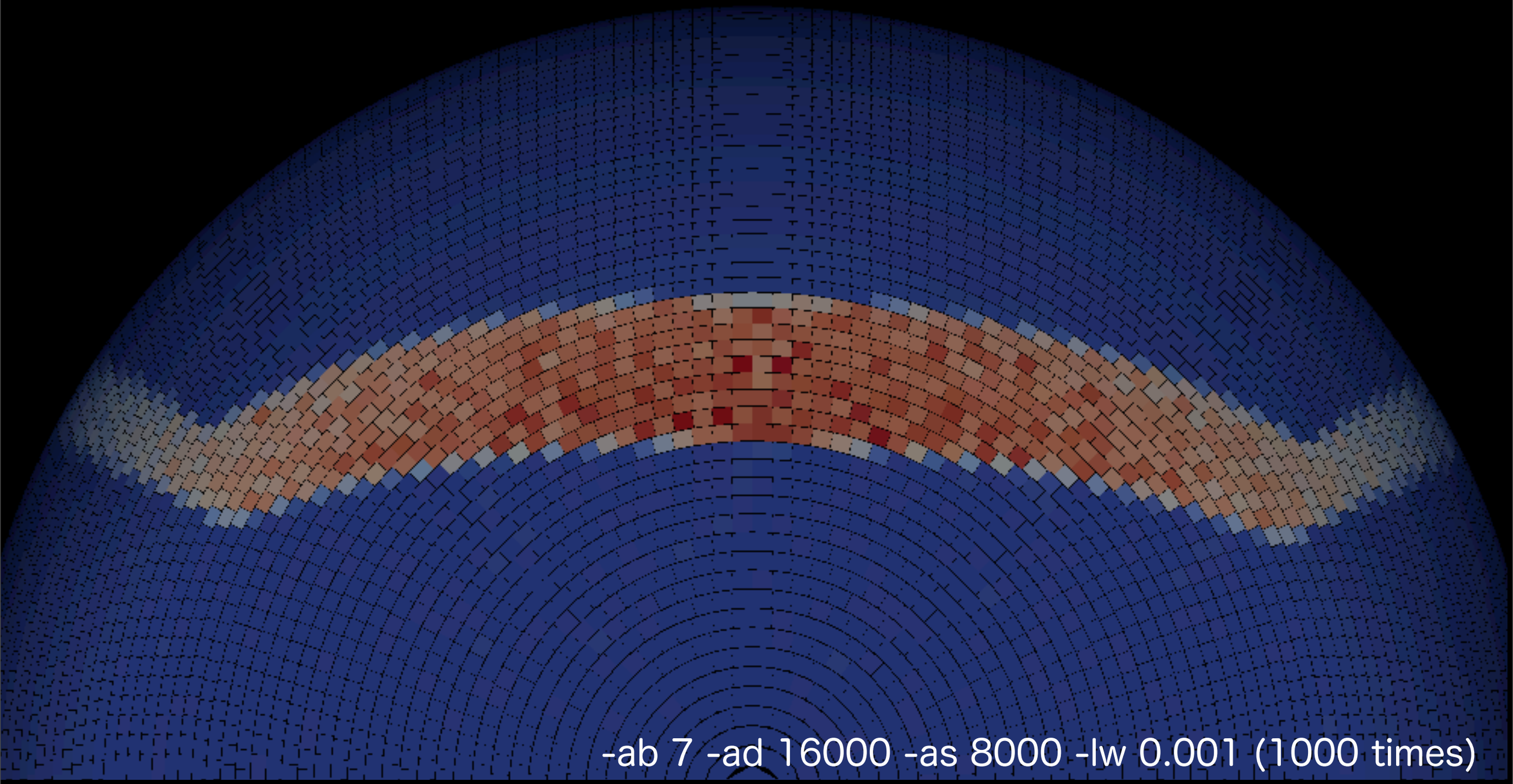
# sky coeficients



# sky coeficients



# sky coeficients




Arup | Lighting