CENTER FOR ASTROPHYSICS HARVARD & SMITHSONIAN

Development of a Telescope Design System

Sukhman Singh (Yale University)

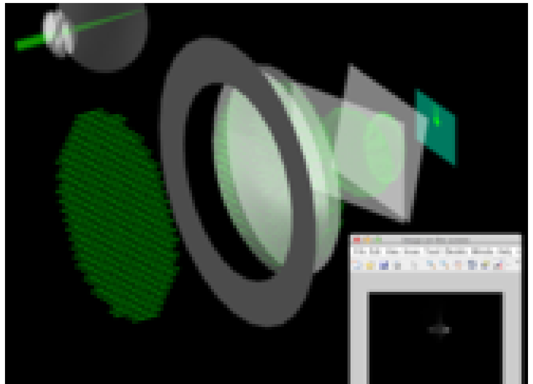
Peter Cheimets and Edward Hertz (The Center for Astrophysics | Harvard & Smithsonian)

Overview

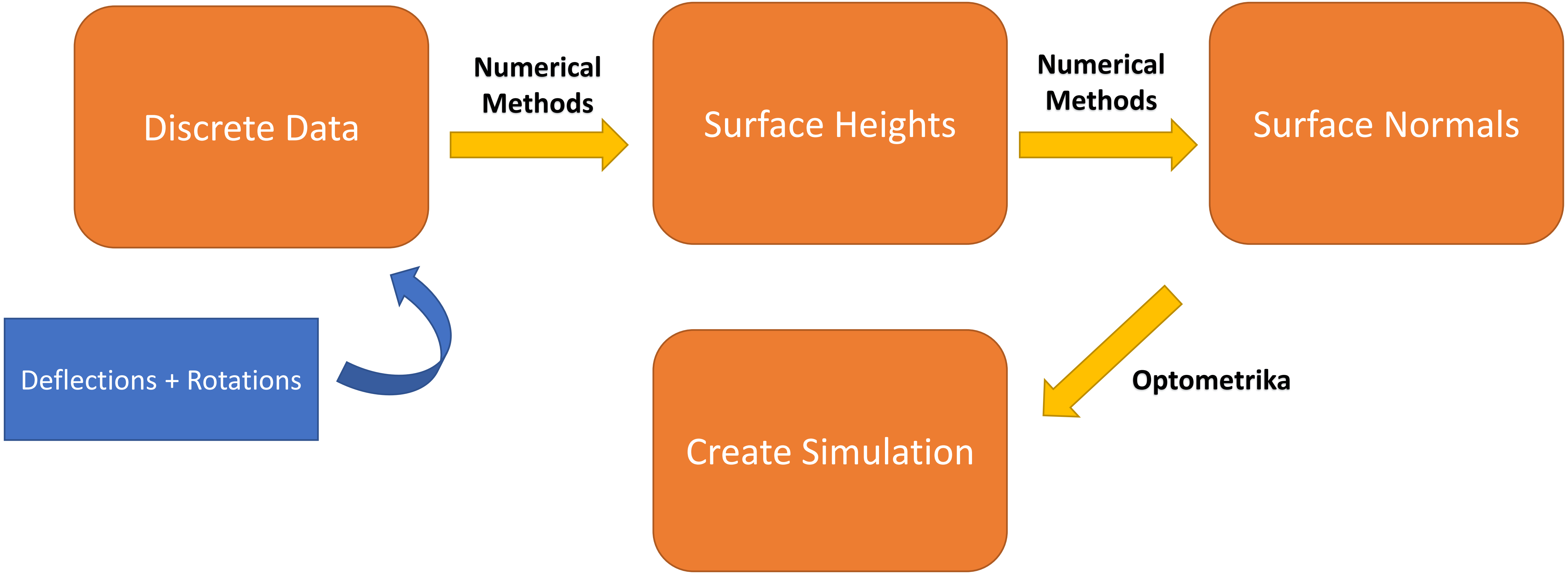
- **<u>Goal</u>**: To understand the relationship between off-nominal conditions and telescope performance
- **<u>Method</u>**: Modify a software program to simulate the optical performance of lenses and mirrors with non-computational surfaces
  - Surface heights and normals of computational surfaces can be found everywhere by their equations (e.g., sphere)
  - Non-computational surfaces are defined by a discrete set of points

    ➔ **Problem**: Surface heights and normals are known only at those points, but we need to know them everywhere

    ➔ **Solution**: Use numerical techniques like interpolation for calculating these parameters and creating simulations



Layout



Discrete Data
Numerical Methods
Surface Heights
Numerical Methods
Surface Normals
Deflections + Rotations
Optometrika
Create Simulation

Example: Discrete Data for a Sphere

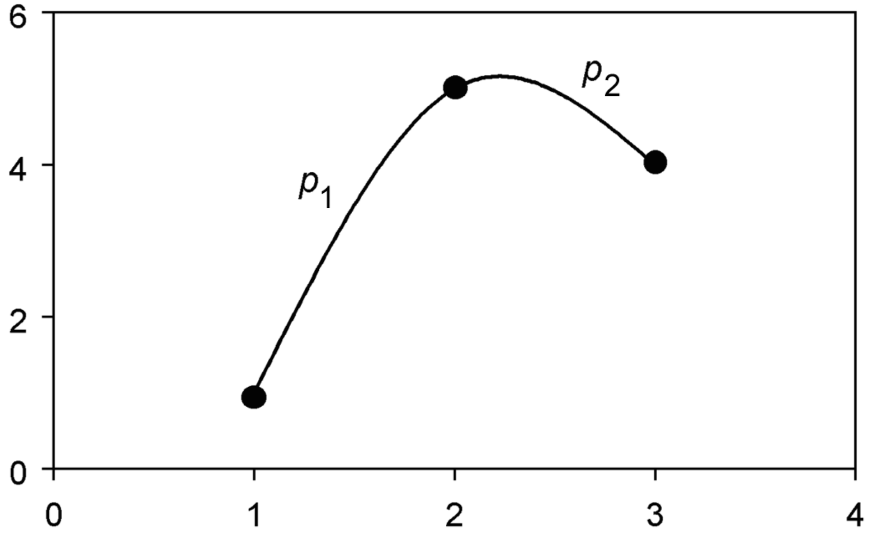
- Given a regular mesh of y and z data points, we can find surface heights (i.e., x values) using the equation of a sphere:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$$

- Suppose we treat the sphere instead as a collection of discrete points
  - Given set of y and z data, we need to find the x values at intermediate points in this data **without using the above equation**
  - Ex: Suppose y and z have the range (-1, -0.5, 0, 0.5, 1) ➔ Find x at (-0.4, 0.9)
  - Solution: Interpolation

Surface Heights

- Interpolation – A numerical estimation method that calculates an unknown value between two known values
- Cubic Spline Interpolation: Each interval between (x_i, y_i) and (x_{i+1}, y_{i+1}) is connected by a piecewise cubic polynomial
  - The polynomial must match the values at the data points
  - The first and second derivatives are continuous

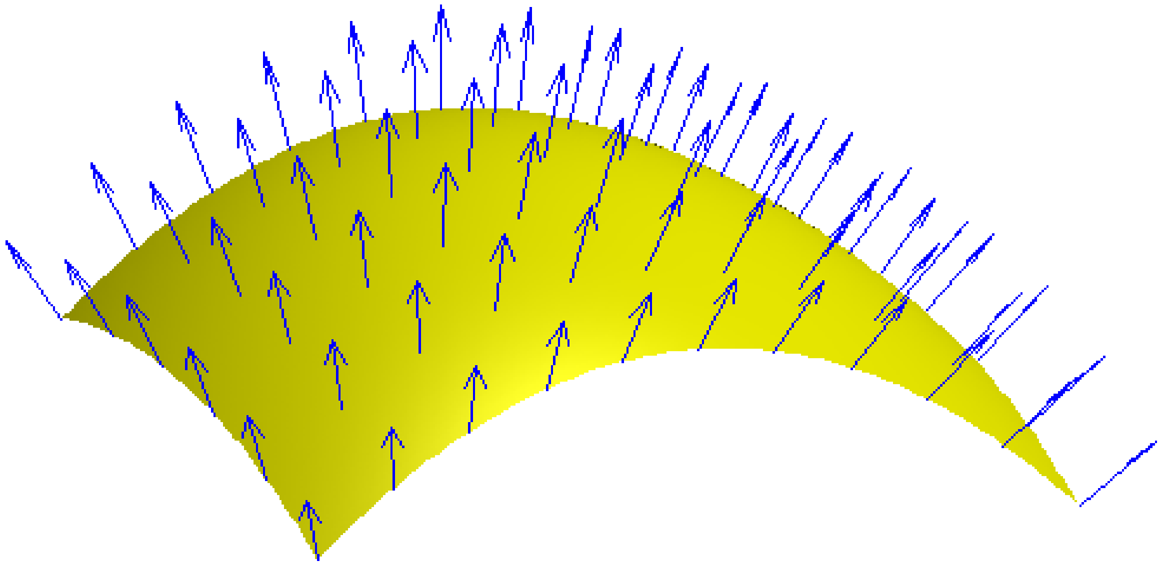


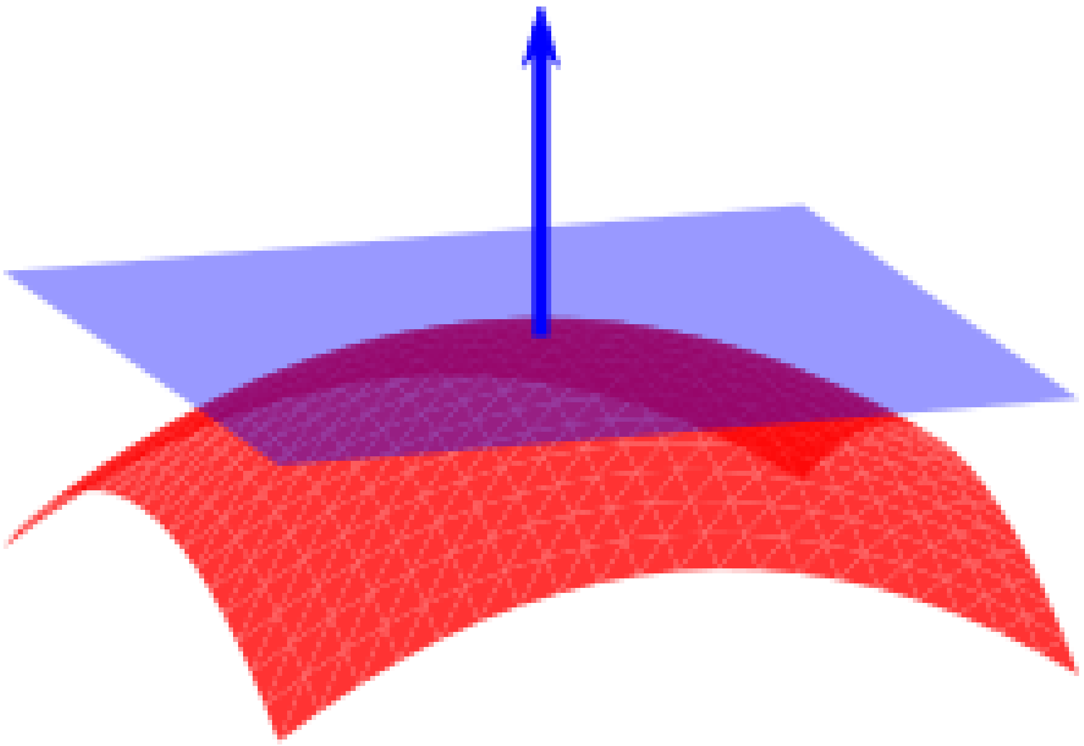
Two cubic polynomials create a cubic spline between the points (1,1), (2,5), and (3,4).

Surface Normals

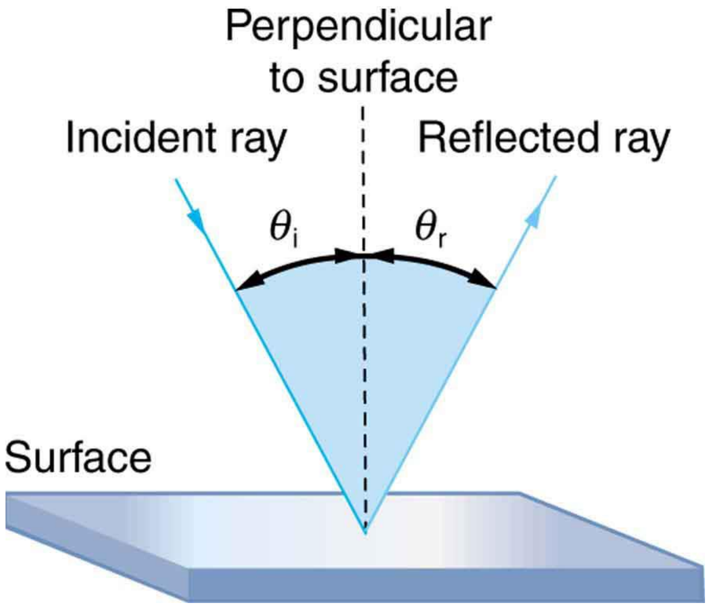
- Surface normal = normal vector at any point on a surface
- Need to find vector representation of these normals at any given point

➔ Determine path of reflected rays



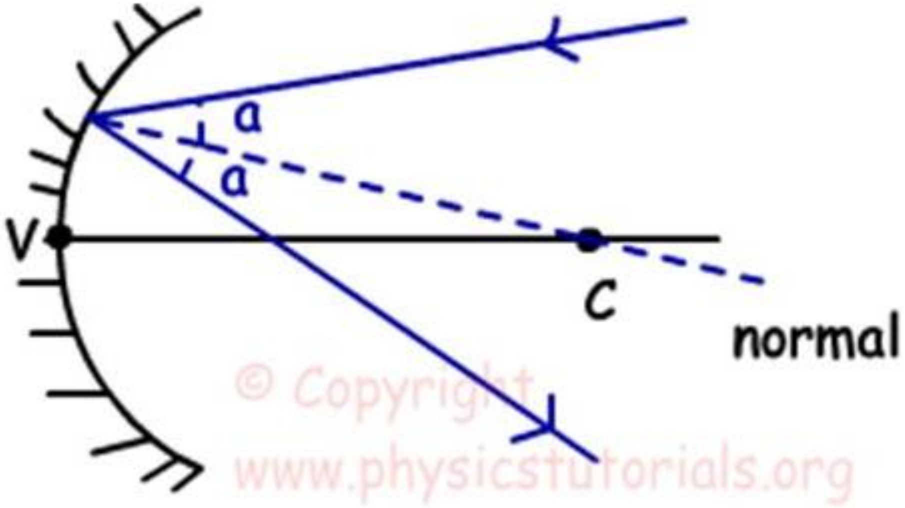


Law of Reflection



- Angle of Incidence = Angle of Reflection

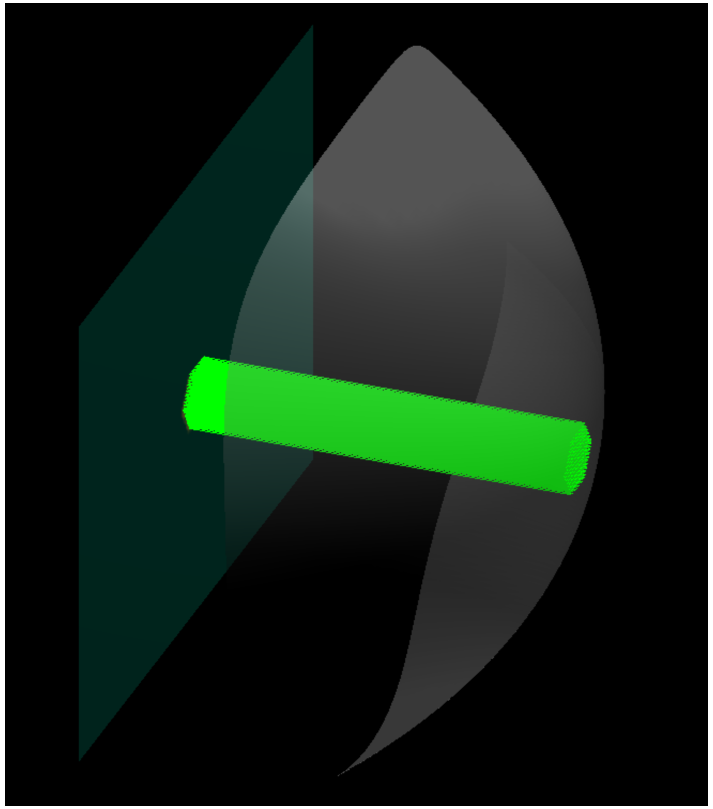
- For a spherical concave mirror:
  - Normal extends from point of incidence to center of curvature
  - Angles of incidence + reflection measured from this normal



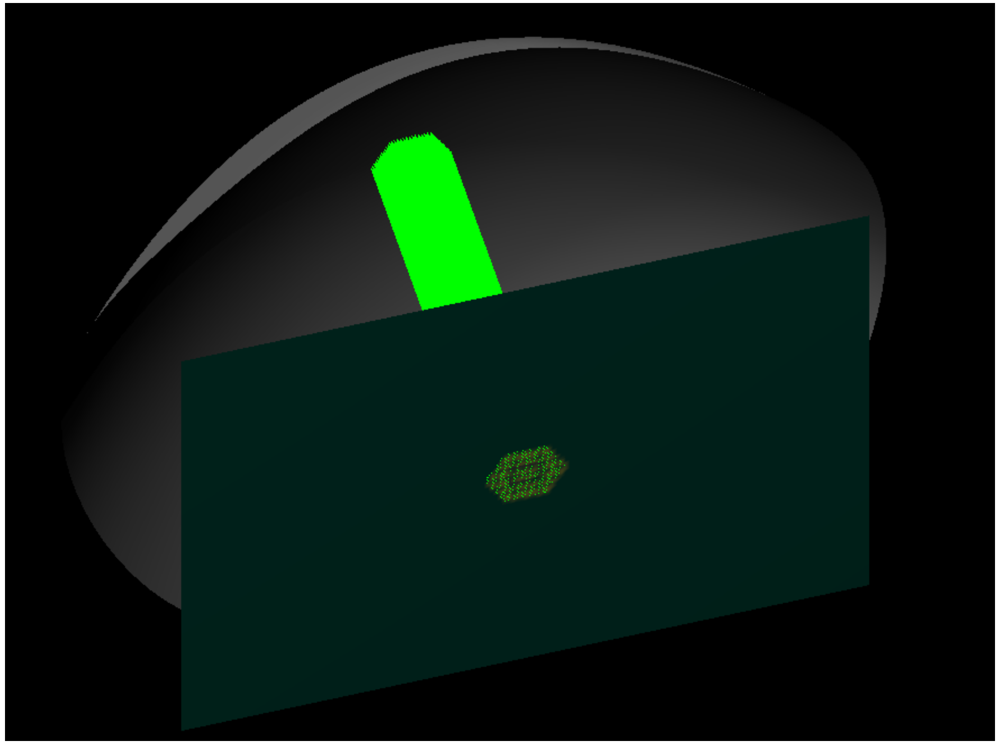
Constructing Surface Normals

1) Treating the sphere as a computational surface:

- Normal at each point is the gradient vector at that point (with an orientation), i.e., take derivatives



500 rays focused on a screen



Same simulation from a different orientation

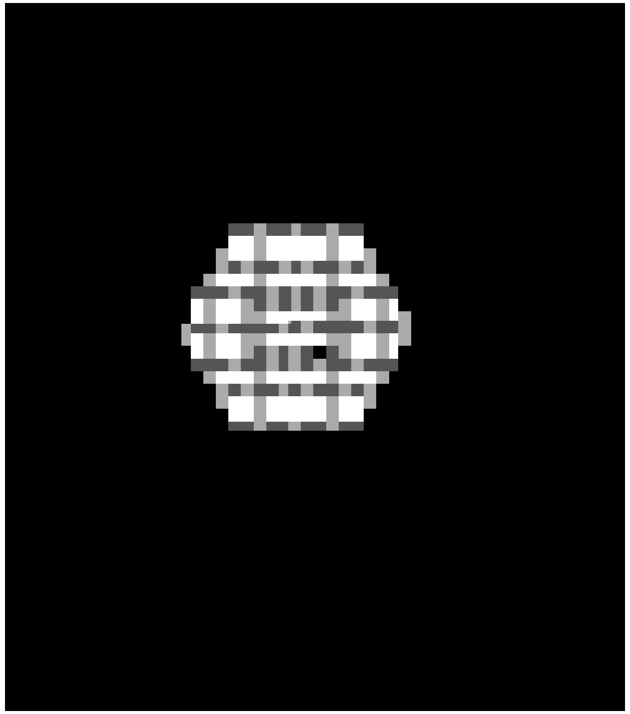


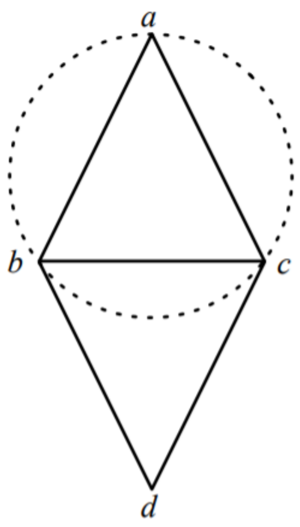
Image of focused rays on screen

2) Treat the sphere as a non-computational surface:

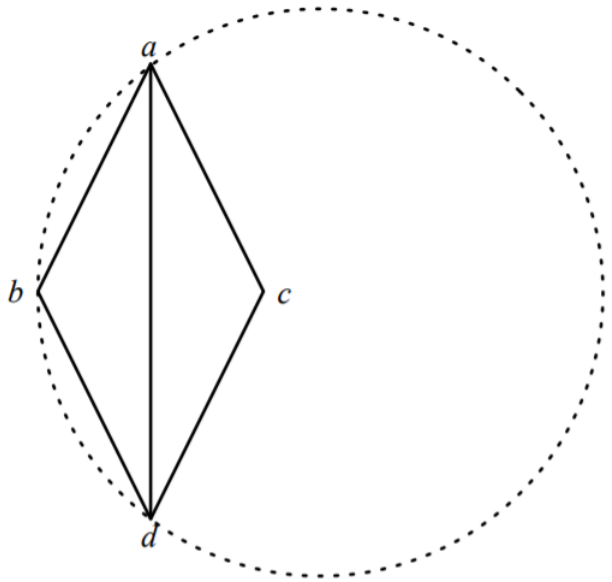
- Delaunay Triangulation (one of the unsuccessful numerical methods)

➔In 2D, a set of points has a Delaunay Triangulation if the circumcircle of each triangle formed from three points contains no points in its interior.

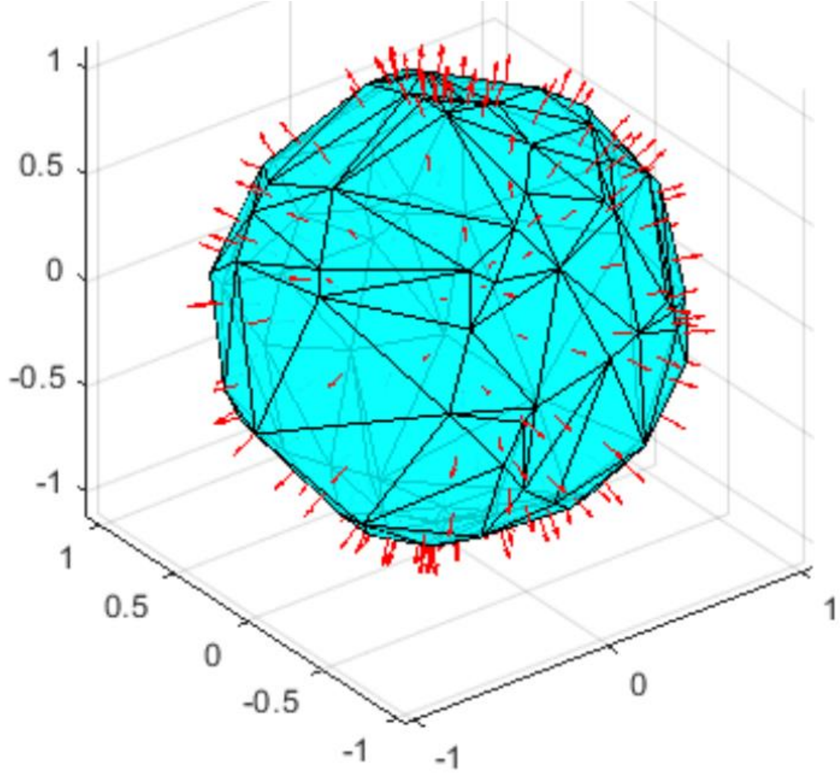
- Find normal through center of each triangle



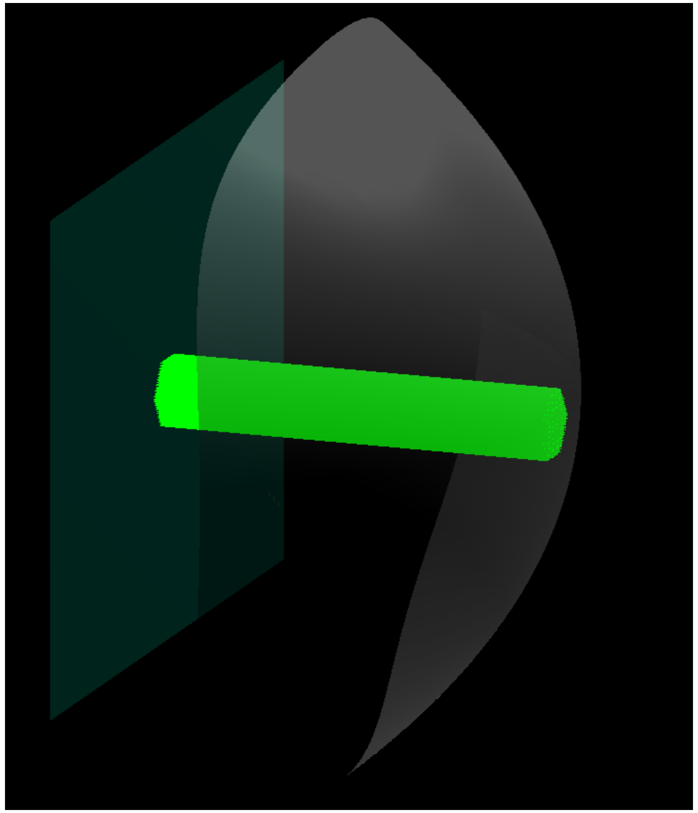
Delaunay Triangulation



Not a Delaunay Triangulation



Results



Rays appear to be focusing on screen

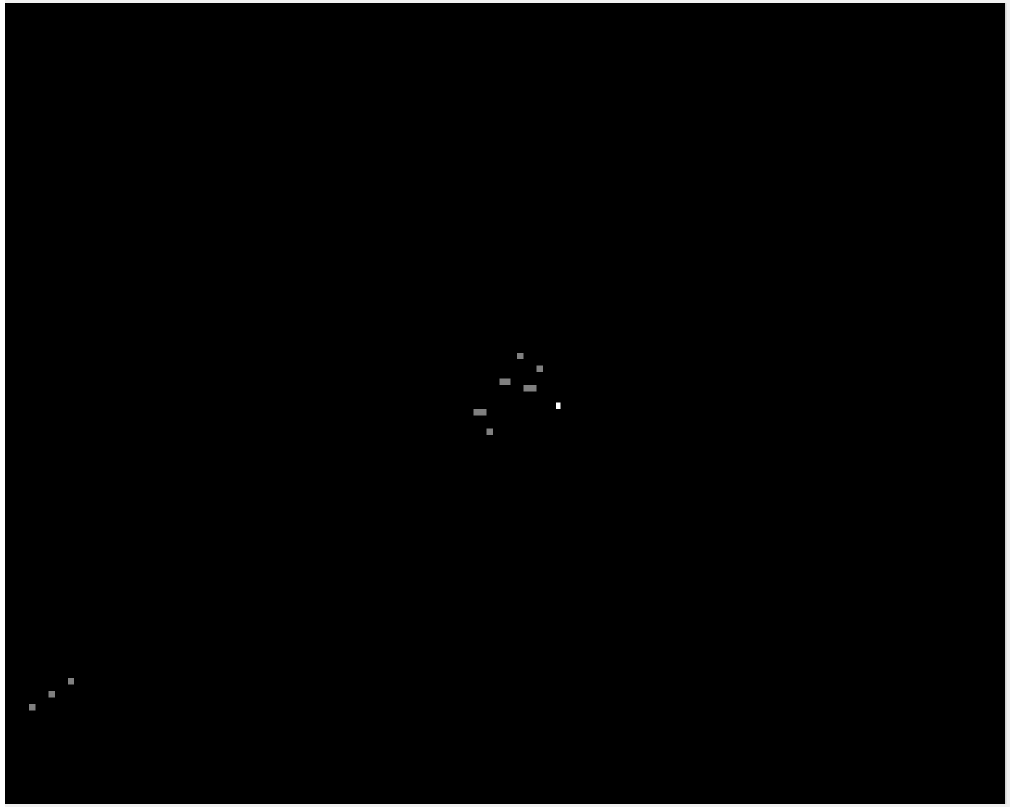
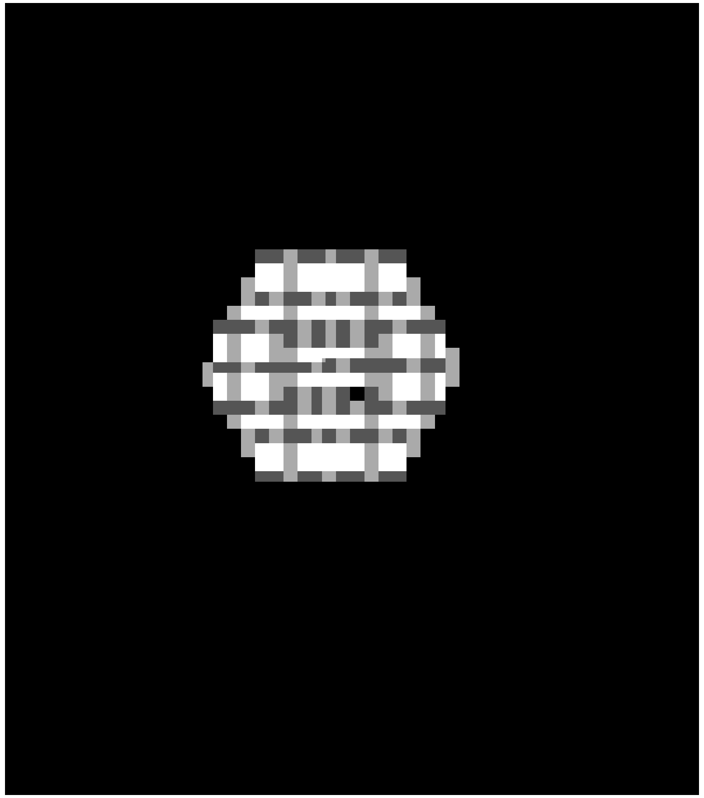
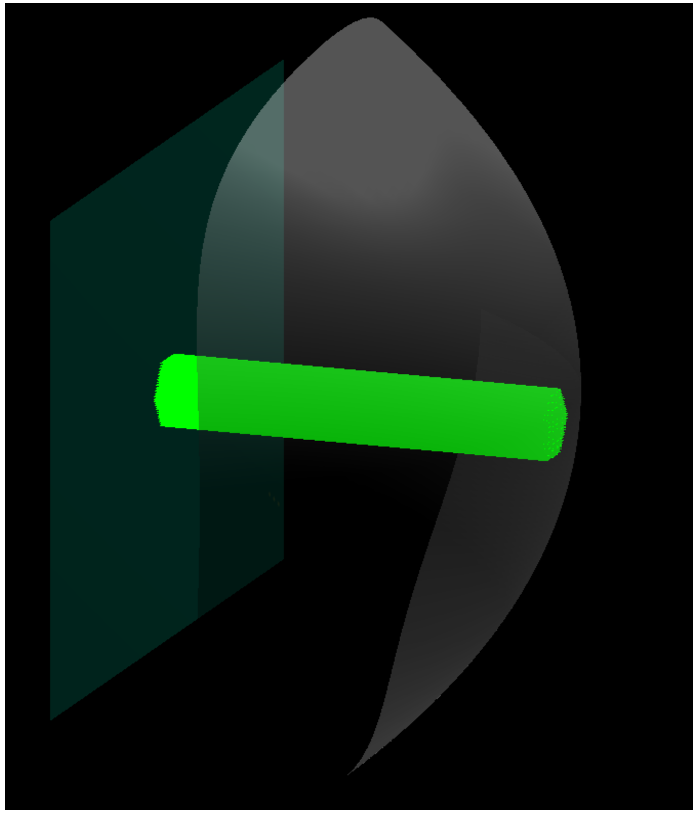


Image of focused rays on screen



Standard of Comparison

Results



Rays appear to be focusing on screen

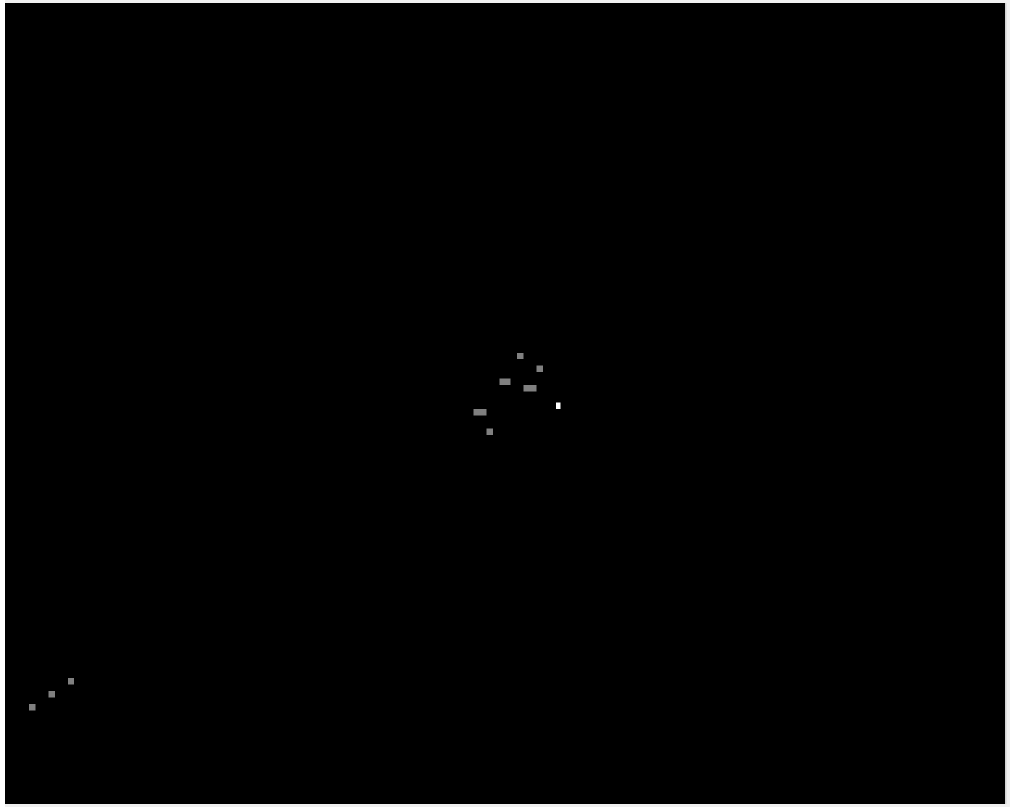
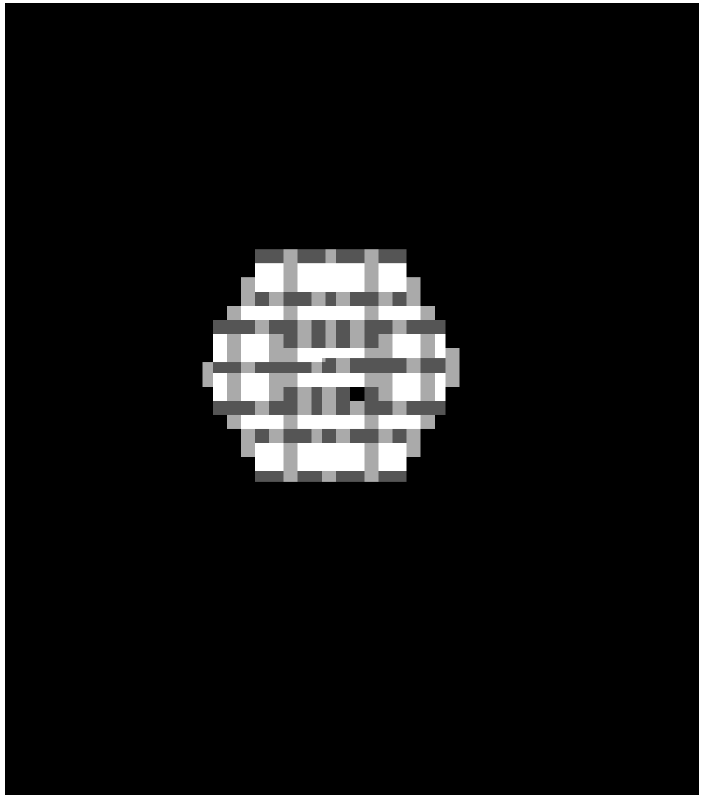


Image of focused rays on screen



Standard of Comparison

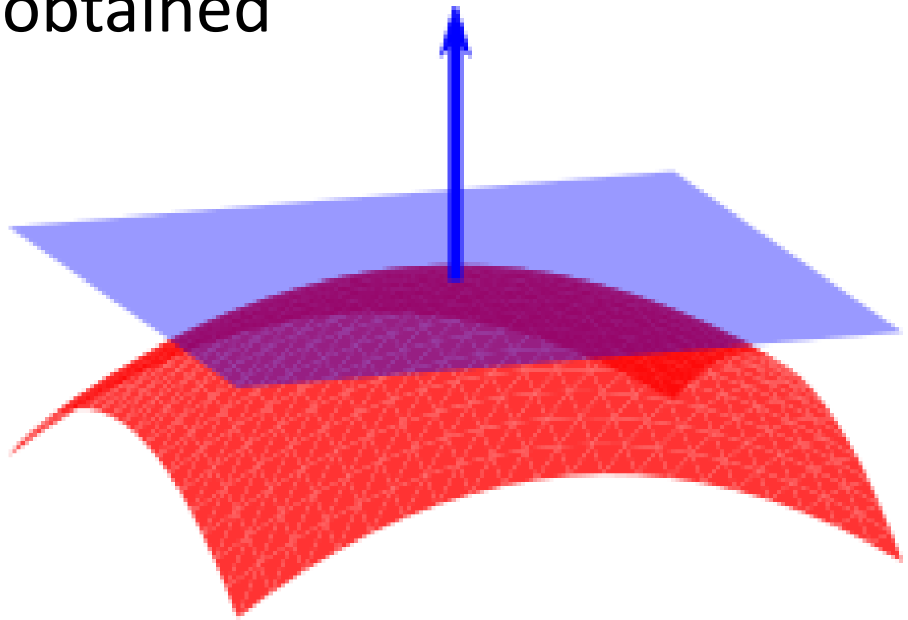
Problem: The surface normals aren't being drawn at the query points dictated by the program. They are instead being drawn at the centers of the triangles.

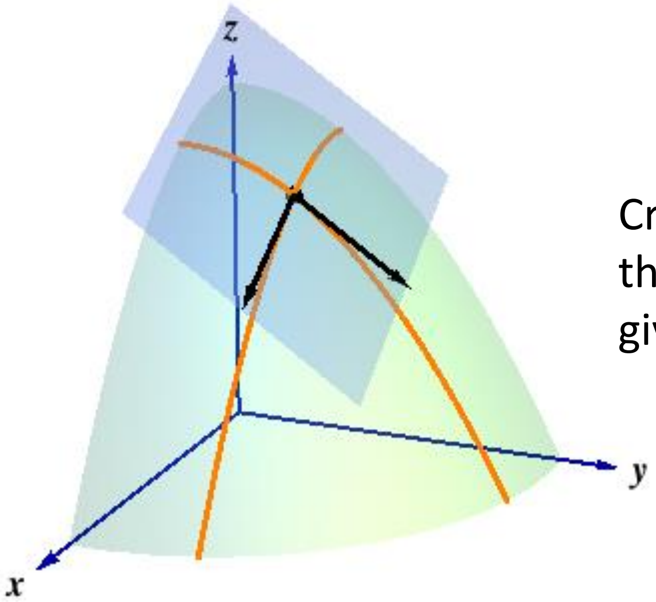
A Brute Force Algorithm

- Any three points uniquely define a plane

Algorithm:

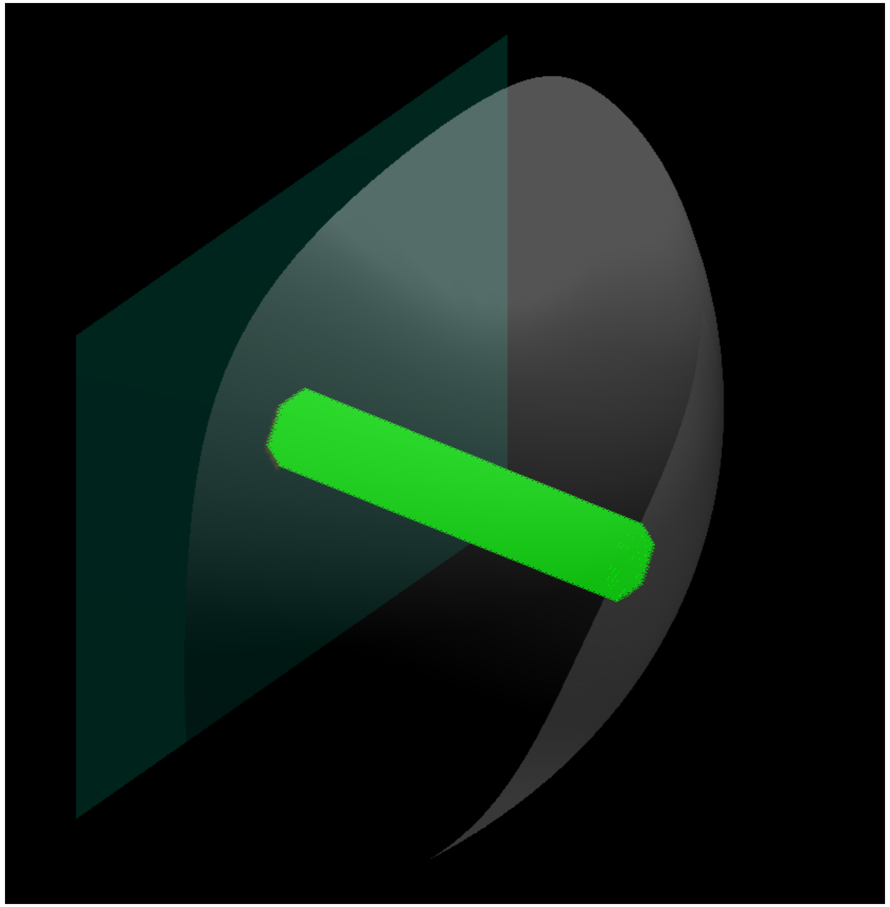
1) From three (interpolated) data points, find equation of their plane
2) Calculate normal vector to that plane
3) Keep shuffling down data set until desired number of normals are obtained





Cross Product of the black vectors gives the normal

Results



Rays appear to be focused on the screen

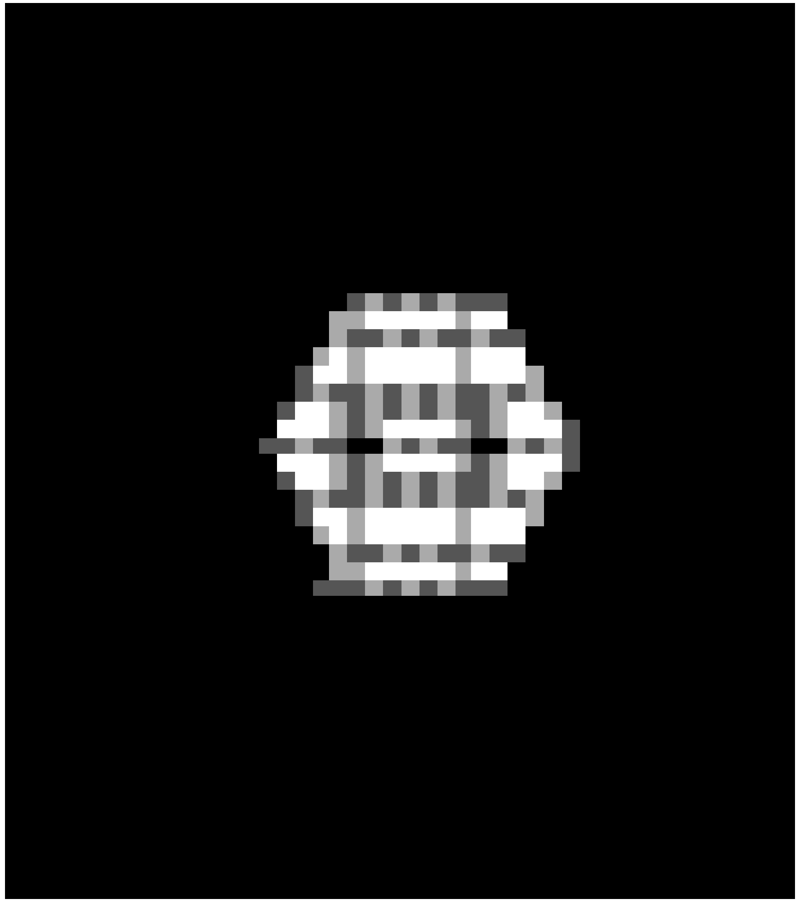
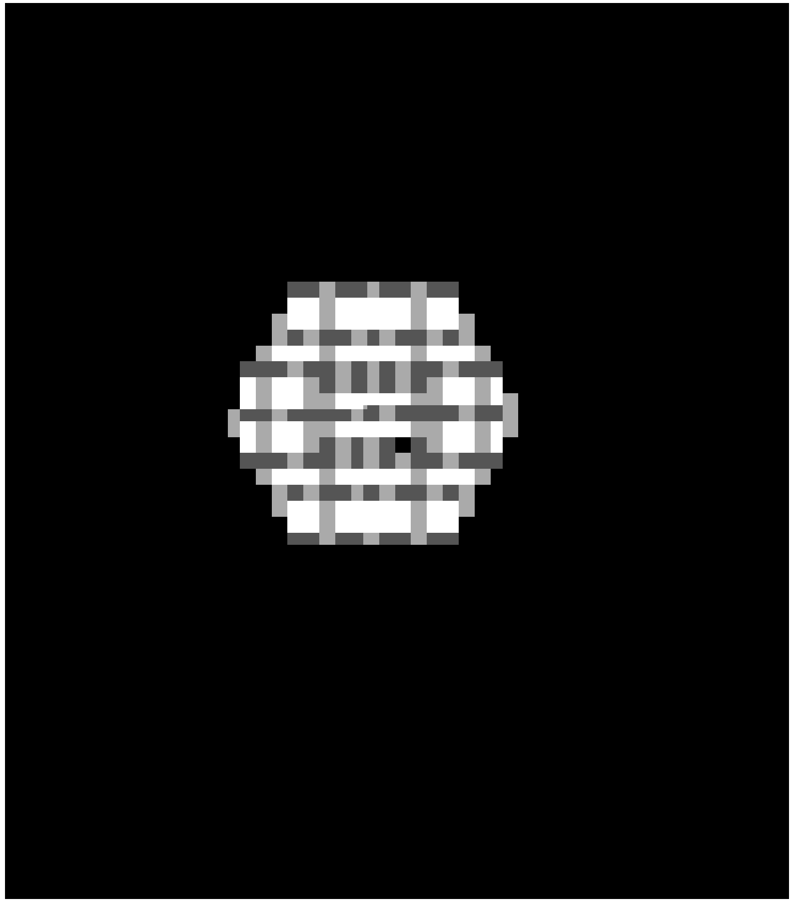
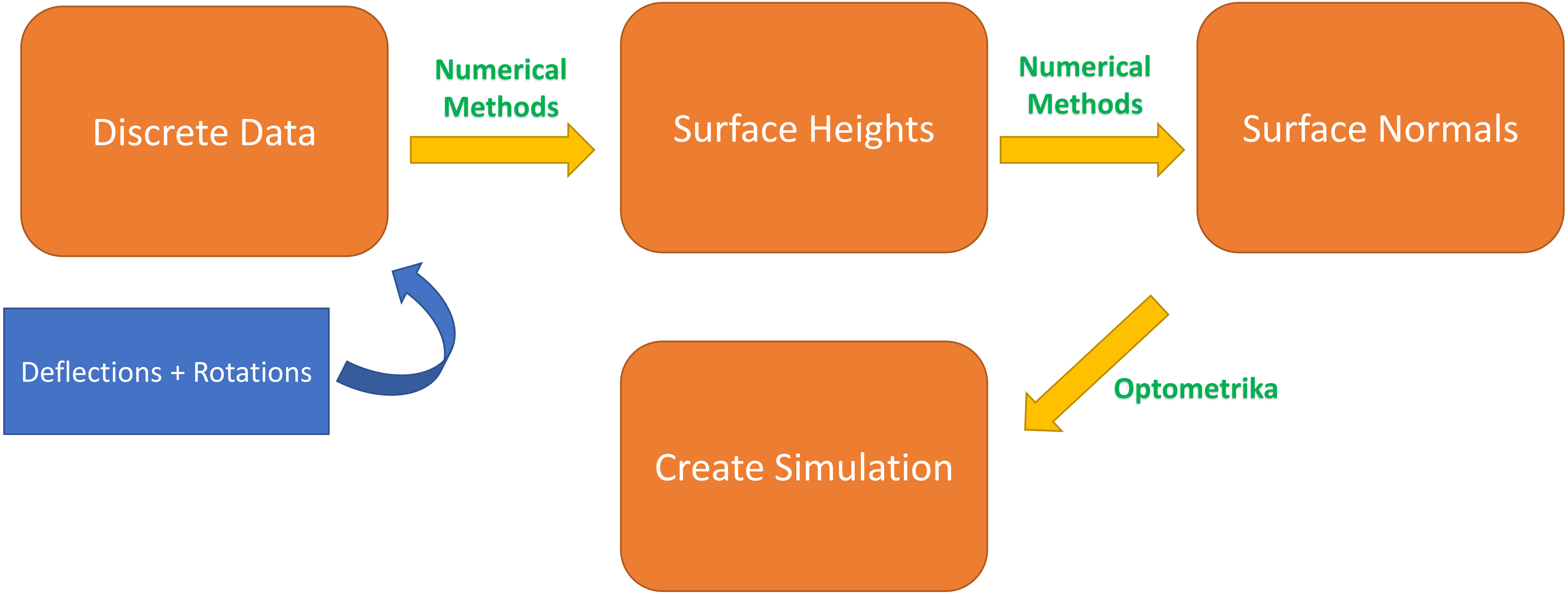


Image of screen produced by the brute force algorithm

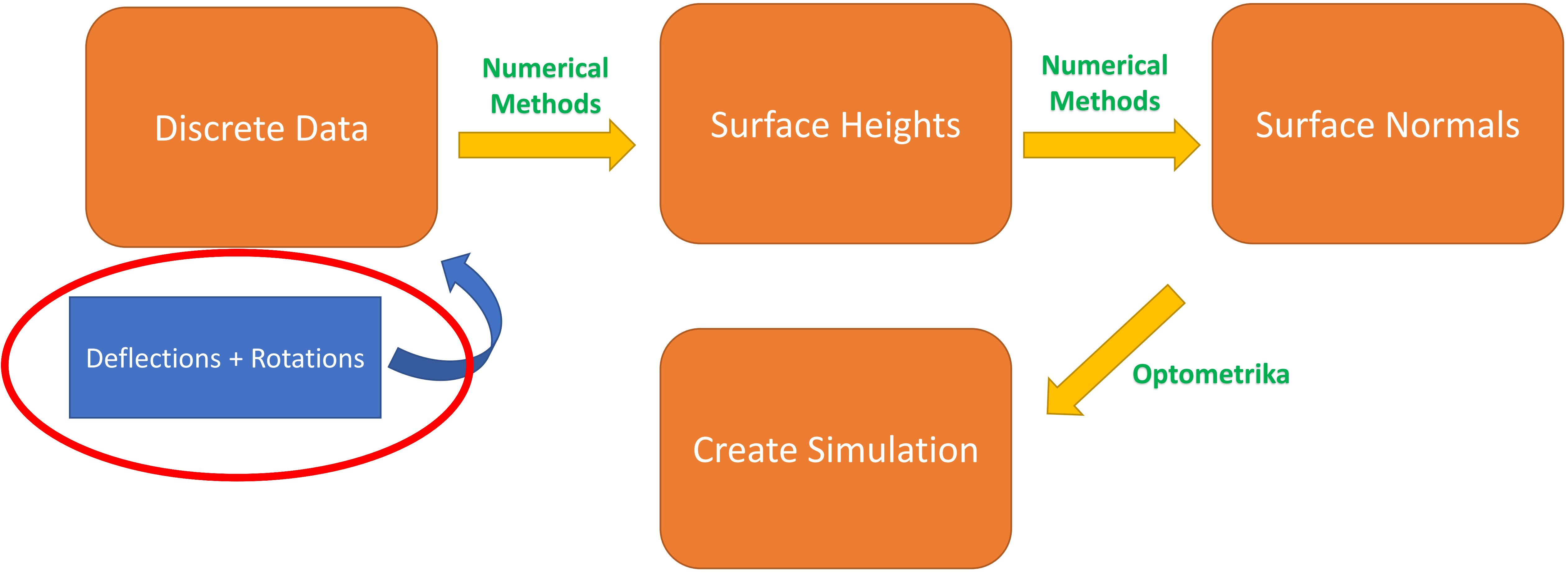


Standard of Comparison

Layout


Discrete Data
Numerical Methods
Surface Heights
Numerical Methods
Surface Normals
Deflections + Rotations
Optometrika
Create Simulation

Layout

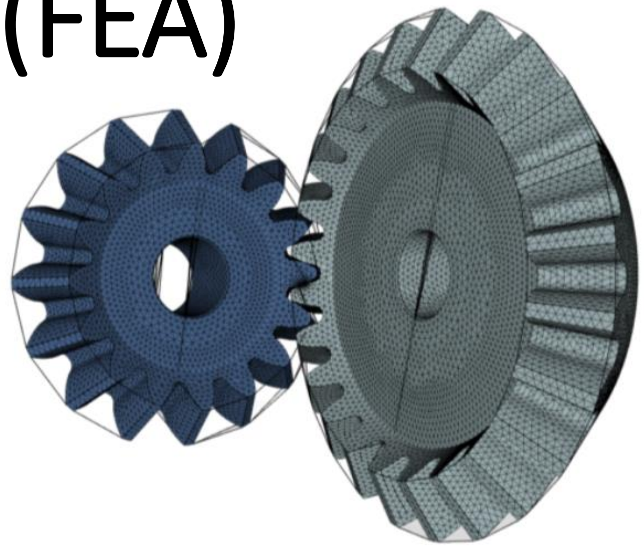

Discrete Data
Numerical Methods
Surface Heights
Numerical Methods
Surface Normals
Deflections + Rotations
Optometrika
Create Simulation

Finite Element Analysis (FEA)

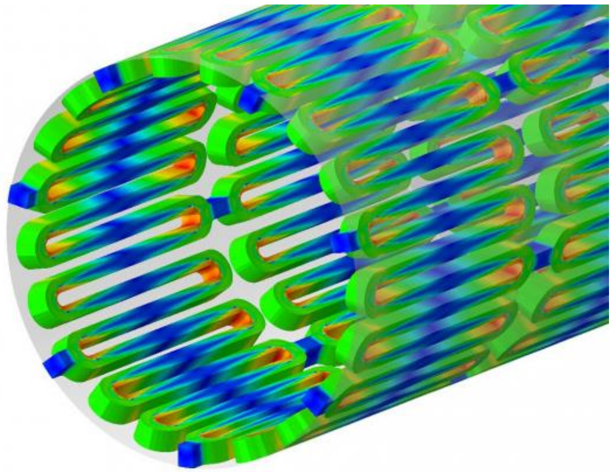
- A method for predicting how an object reacts to environmental forces
  - Vibrations, mechanical stress, motion, heat flow, fluid flow, etc
- The object is broken down into a large number of smaller “elements” (e.g., little cubes)

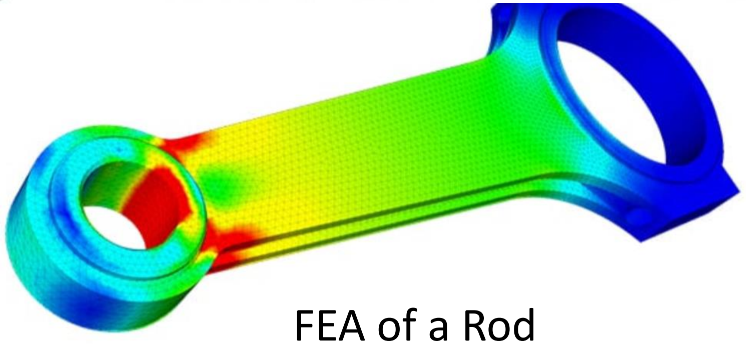
➔Equations predict the behavior of each element

➔These behaviors are added up to obtain the behavior of the aggregate

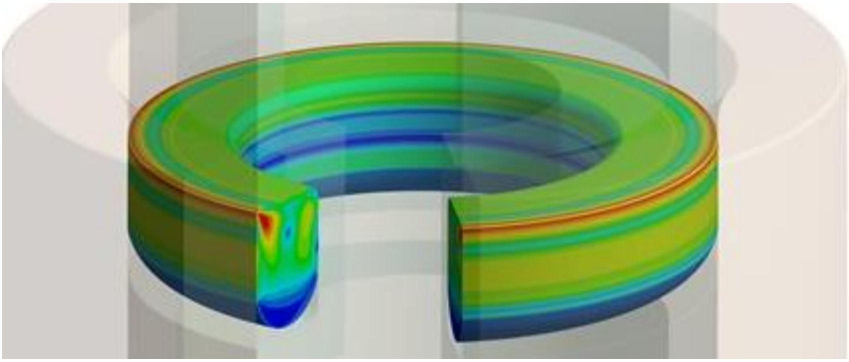


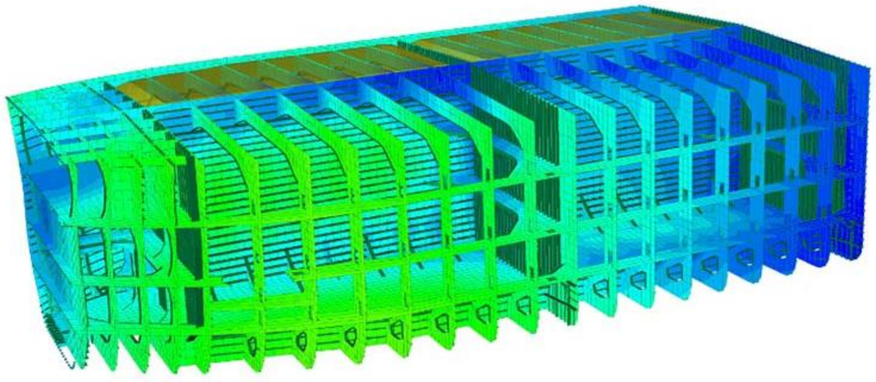
Meshing of Gears





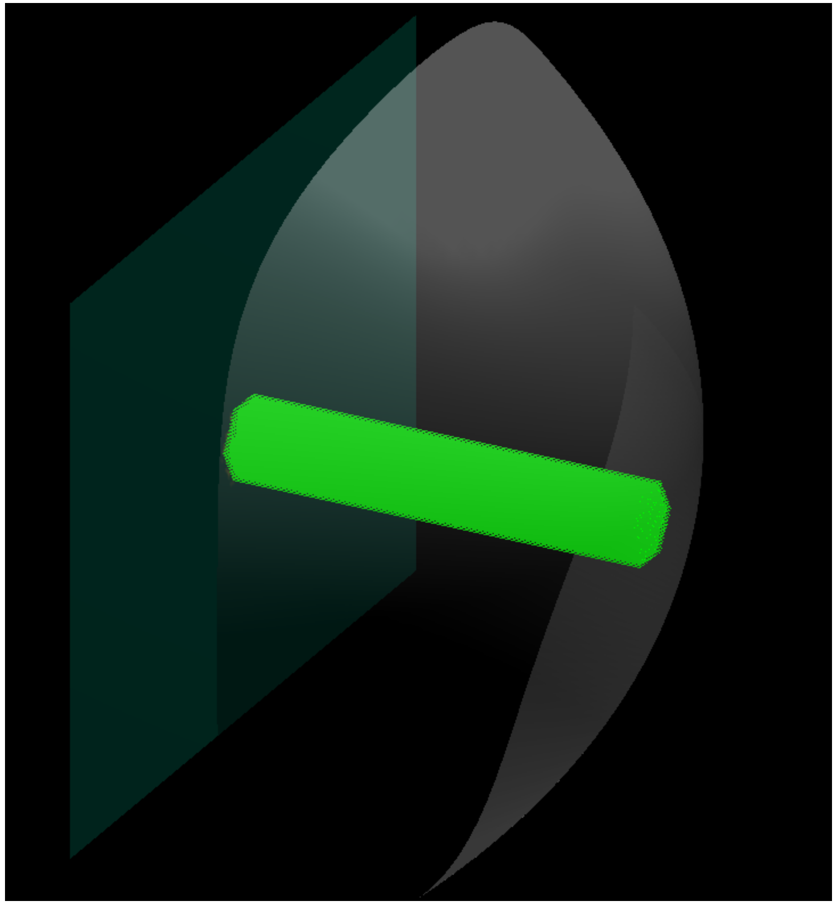
FEA of a Rod





Simulating a Deformed Geometry

- Add random numbers within a tenth of a micron to surface heights and within a hundredth of a micron to left/right translations



Rays appear to be focusing on the screen

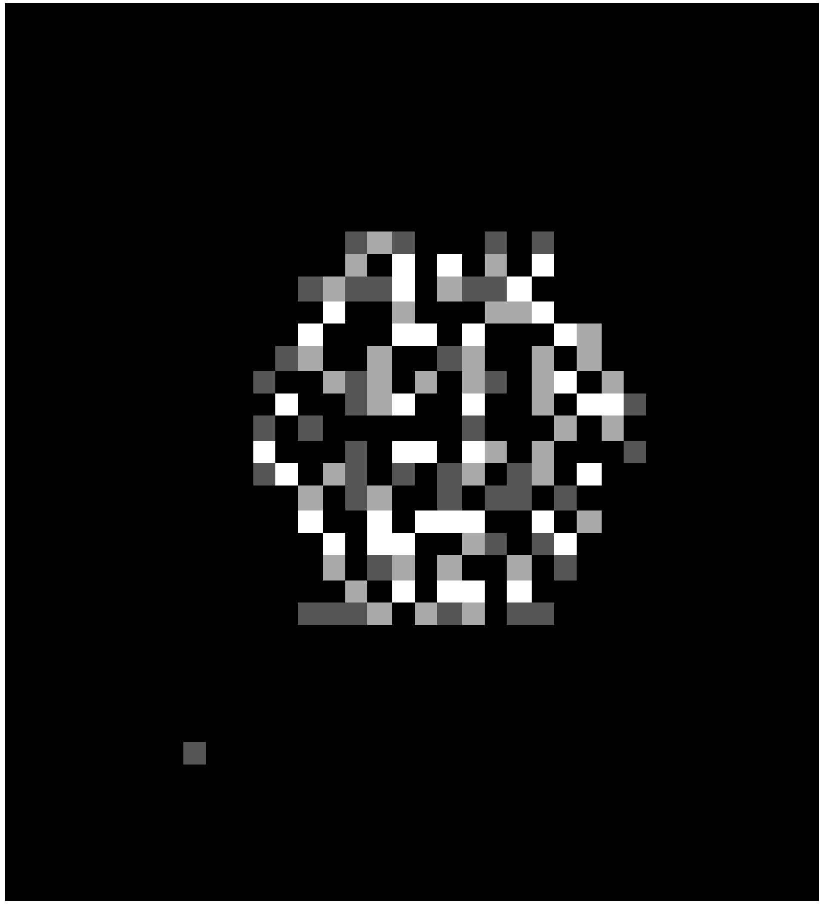
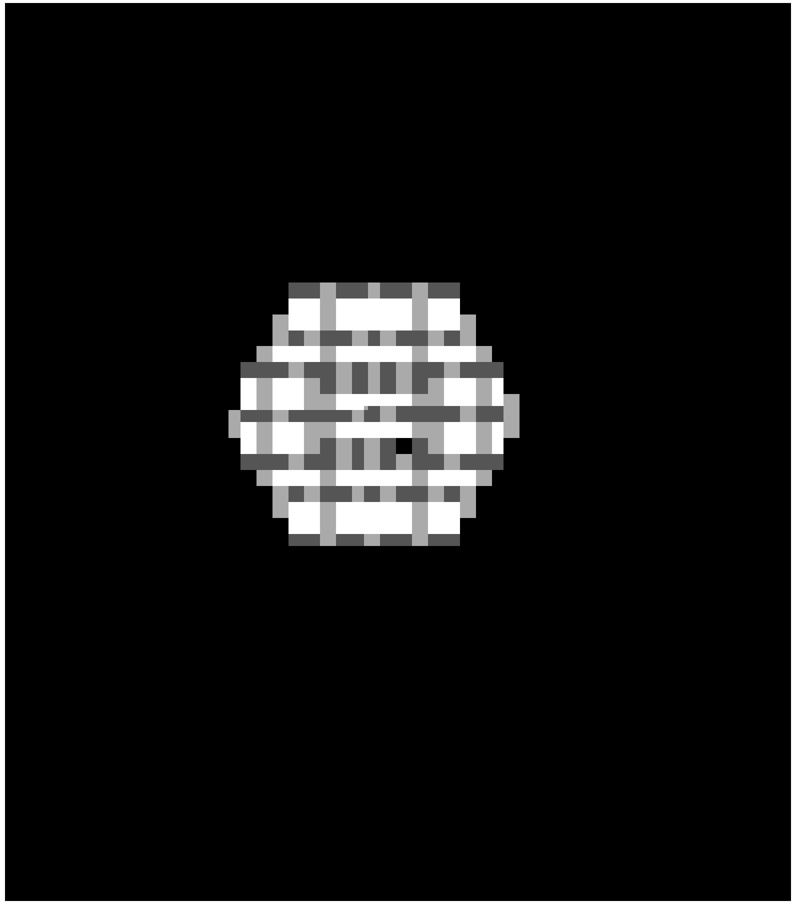


Image of screen



Standard of Comparison

Next Steps

- Use the FEA output of a real mirror model as our discrete data set
  - A deformed geometry consisting of deflections + rotations
- Apply our telescope design program to this data
- Simulate a real mirror and add the deflections on this perfect surface to the FEA output

Sources for Images Taken From Internet

- Slide 2: https://www.mathworks.com/matlabcentral/fileexchange/45355-optometrika
- Slide 6: https://en.wikipedia.org/wiki/Normal_(geometry)
- Slide 9: https://www.mathworks.com/help/matlab/ref/triangulation.facenormal.html#d120e1040181
- Slide 9: https://fsu.digital.flvc.org/islandora/object/fsu:182663/datastream/PDF/view
- Slide 12: https://web.ma.utexas.edu/users/m408m/Display14-4-2.shtml
- Slide 16: https://www.simscale.com/blog/2016/10/what-is-finite-element-method/
- Slide 16: https://interestingengineering.com/what-is-finite-element-analysis-and-how-does-it-work
- Slide 16: https://www.prepol.com/services/finite-element-analysis-fea
- Slide 16: https://www.dnvgl.com/services/finite-element-analysis-of-ship-structures-nauticus-hull-fe-analysis-for-ships-4574