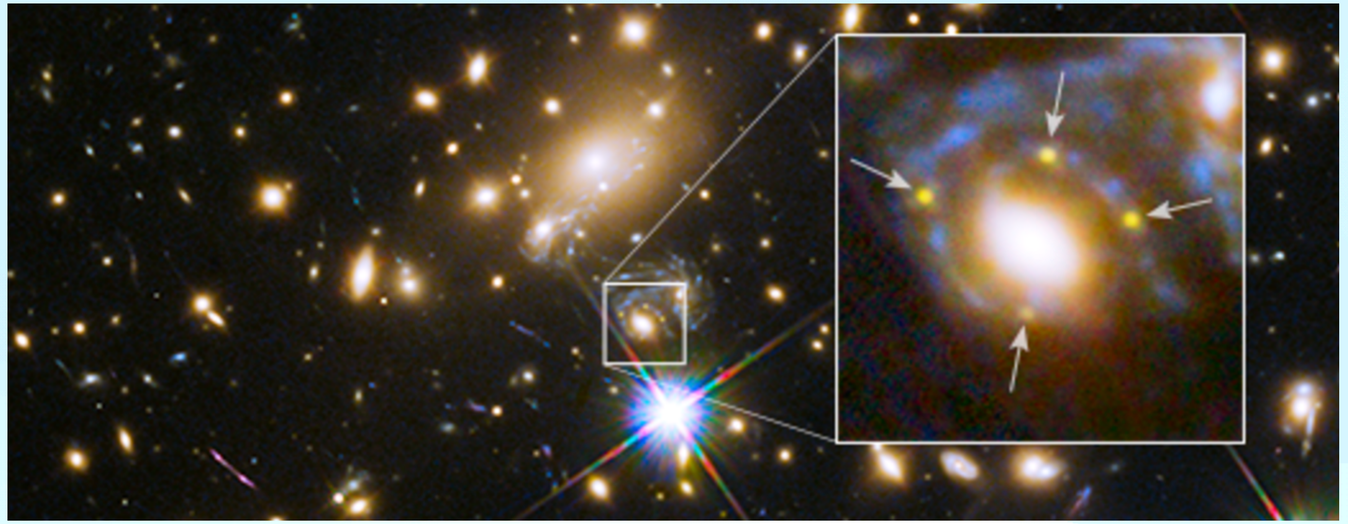
Optimization for Gravitational Point Lenses

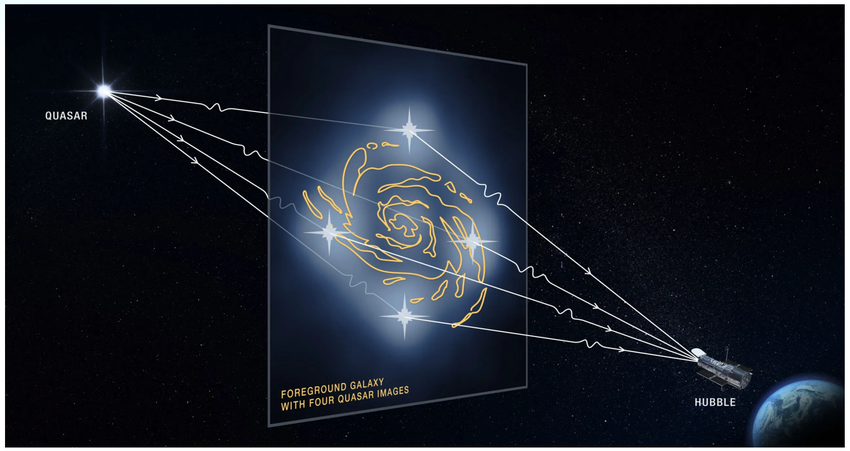
Sophia Miskiewicz

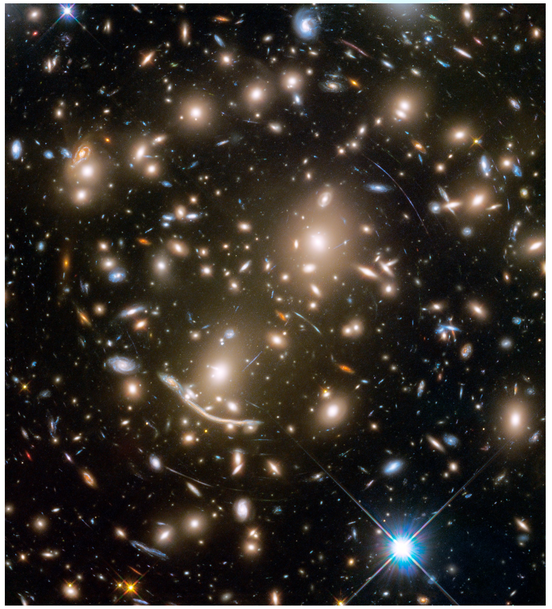
Computational Methods for Astrophysics

Brief Overview on Gravitational Lensing



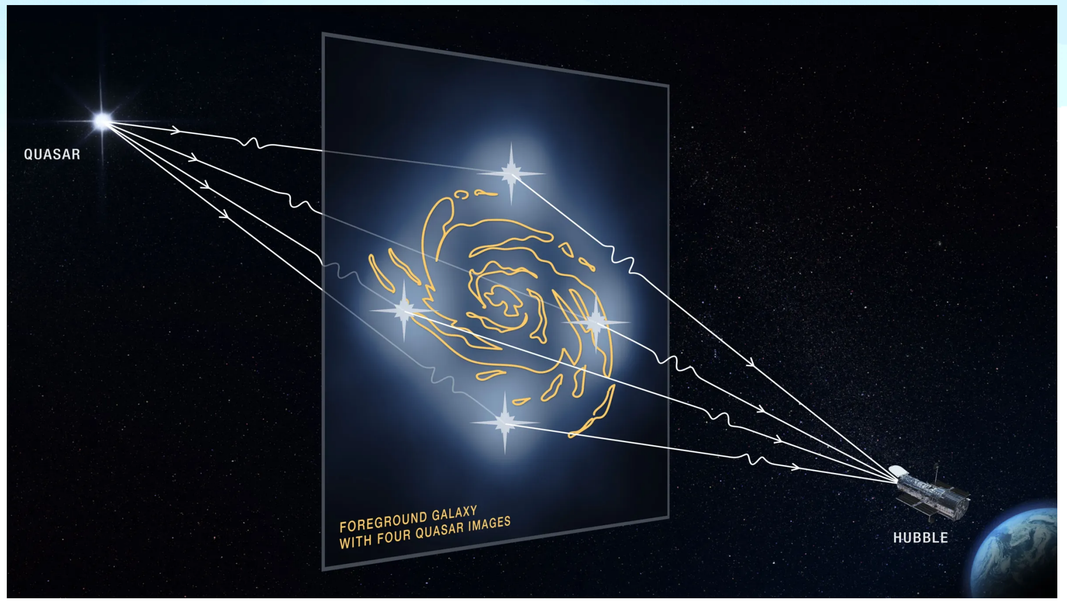
- Gravitational lensing is a phenomenon that occurs due to the presence of a massive object warping space-time around itself.
- This causes light emitted from background objects to bend around the “lens” and produce multiple images, distorted from the original source objects and varying in magnification.





Lensing simulations & parameters

- After collecting image data from a gravitationally lensed system, we often want to reconstruct the parameters of the lens to gain information about and model the system.
- These parameters include the approximate Einstein radius b, the axis ratio q, the angle θ, the location of the lens and the source, as well as the external shear
- We'll call this process "lens fitting"

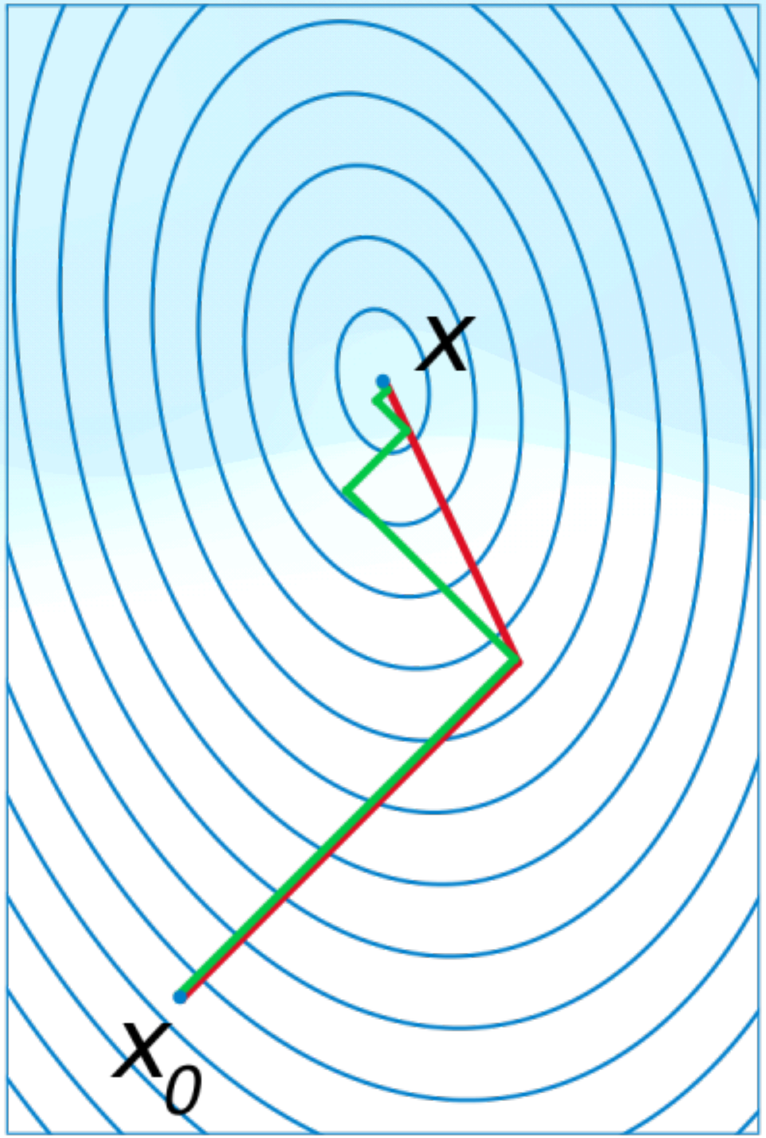


The Problem Statement

- My advisor, Quinn Minor, developed QLens, a software package designed for modeling gravitational lensing systems, and is in the process of creating a python wrapper for it.
- QLens already has multiple methods of optimization in C for fitting lens systems, but my goal here is to see if I can write a script in python to appropriately estimate the lensing parameters and re-create a lensed image.
- For my project, I'll only look at point lenses, so I will not directly be taking into account the shape or mass of the source or the lens.

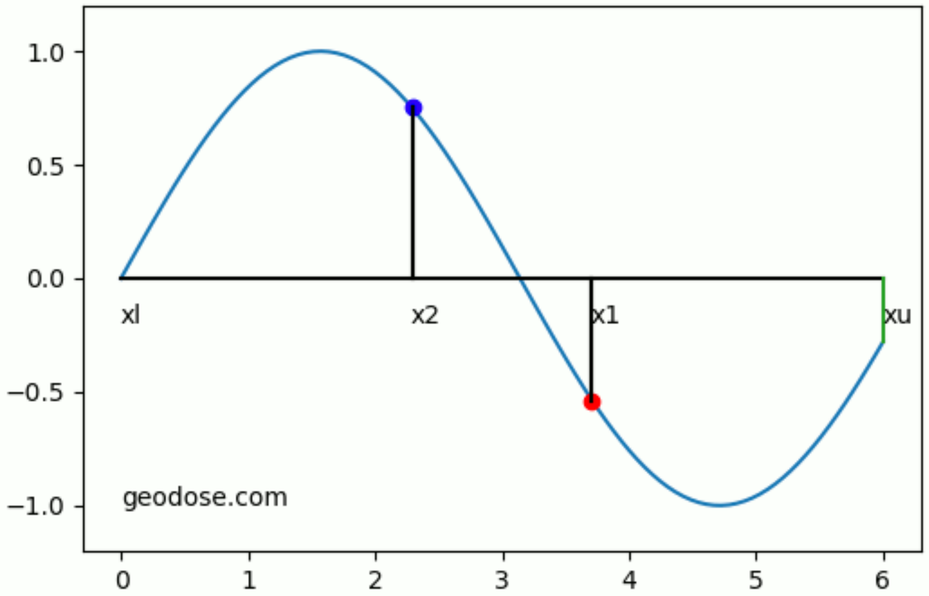
Numerical Approach: Powell's Method

I chose to use Powell's method, which is a method that allows for multi-dimensional optimization of a function. In this case, I was maximizing the Log-Likelihood function for the lensing parameters.



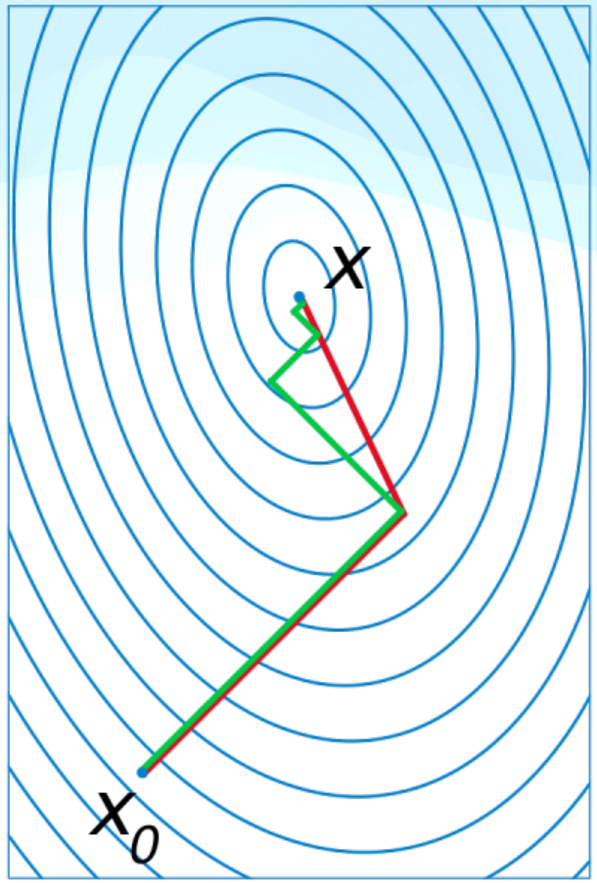
Implementation of Powell's Method

- 1. **The Golden Ratio Search:** first I needed to choose an actual minimization scheme to include in Powell's method. I chose the Golden Ratio Search, which we looked at in class in order to find the extremum value of a function.
- 2. **Initial Guessing:** Make initial guesses for the parameter values & for the search range of these parameters. In this case, I made an initial guess for 7 lensing parameters.



Implementation of Powell's Method

- **3. Establishing initial search directions:** At first, we will be optimizing only one parameter at a time. For this, an identity matrix was initialized for the number of parameters.
- **4. Run the Golden Ratio Search**
- **5. Choose new conjugate directions or search in the same direction**: based on whether the new Log-Likelihood function fulfills certain criteria, we will either continue optimizing in the same direction or choose a new set of conjugate directions in which to move.



Let's take a look at some code!

- Once the virtual environment is ready, anyone should be able to initialize the virtual environment for the QLens python wrapper and run the testalpha_new.py script.
- https://github.com/sophiamiskiewicz/qlens-beta/tree/development
- **Results**: as you can see, the optimization is able to make some reasonable guesses, but still isn't perfect yet. If I work on this further, I think the next steps would include finding a better bracketing procedure rather than just guessing and determining an appropriate stopping condition for Powell's method besides a specified number of iterations.

Thanks for listening!

References

- https://science.nasa.gov/mission/hubble/science/universe-uncovered/hubbles-gravitational-lenses/
- Introduction to Gravitational Lensing and QLens, Quinn E. Minor
- Images:
  - https://miniopt.wordpress.com/2015/01/21/numerical-optimization-8-powells-method/
  - https://hubblesite.org/contents/news-releases/2015/news-2015-08.html
  - https://science.nasa.gov/mission/hubble/science/universe-uncovered/hubbles-gravitational-lenses/
  - https://www.geodose.com/2021/06/golden-section-search-python-application-example.html