Alternative Bandwidth Optimization Methods for GWR

Tyler Hoffman

GIS563 Project 2 Presentation

Problem structure: The AICc curve

$$\mathrm{AICc} = -2\underbrace{\boxed{\log f(y|\beta,\sigma^2,X)}}_{\text{log-likelihood}} + \underbrace{\boxed{\frac{2n(k+1)}{n-k-2}}}_{\text{GWR correction}}$$

Problem structure: **The GWR correction**

The GWR correction is a *monotonically increasing* function of k:

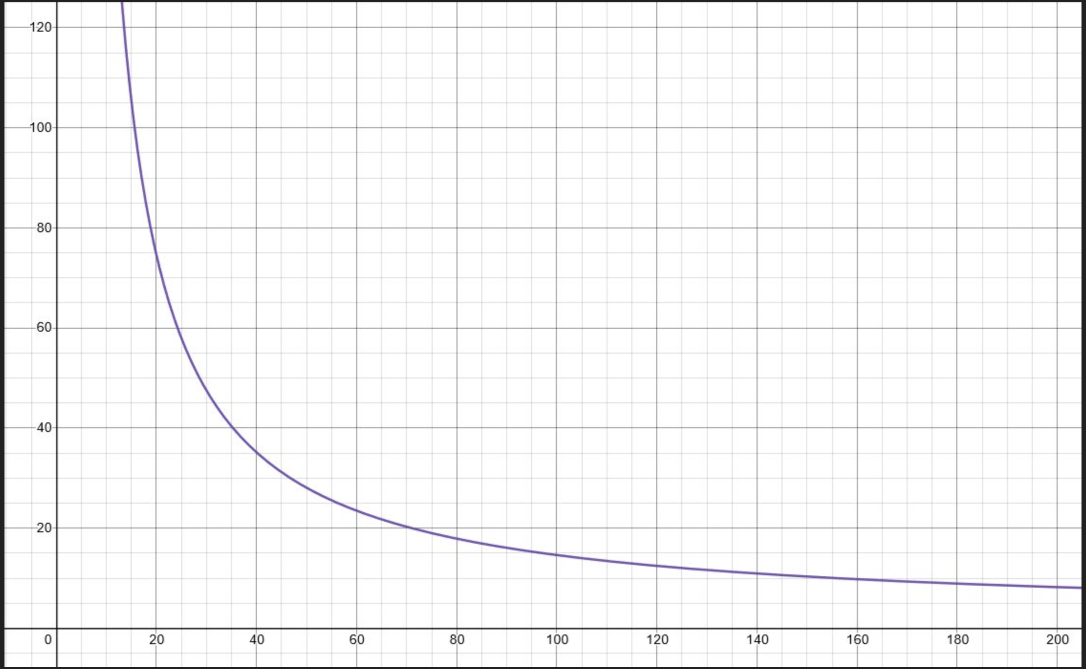
$$\text{as } k \rightarrow n, \frac{2n(k+1)}{n-k-2} \rightarrow \infty$$

k (in the interval [p, n]) is a *decreasing* function of bandwidth:

$$\text{as bw} \rightarrow \infty, k \downarrow p$$

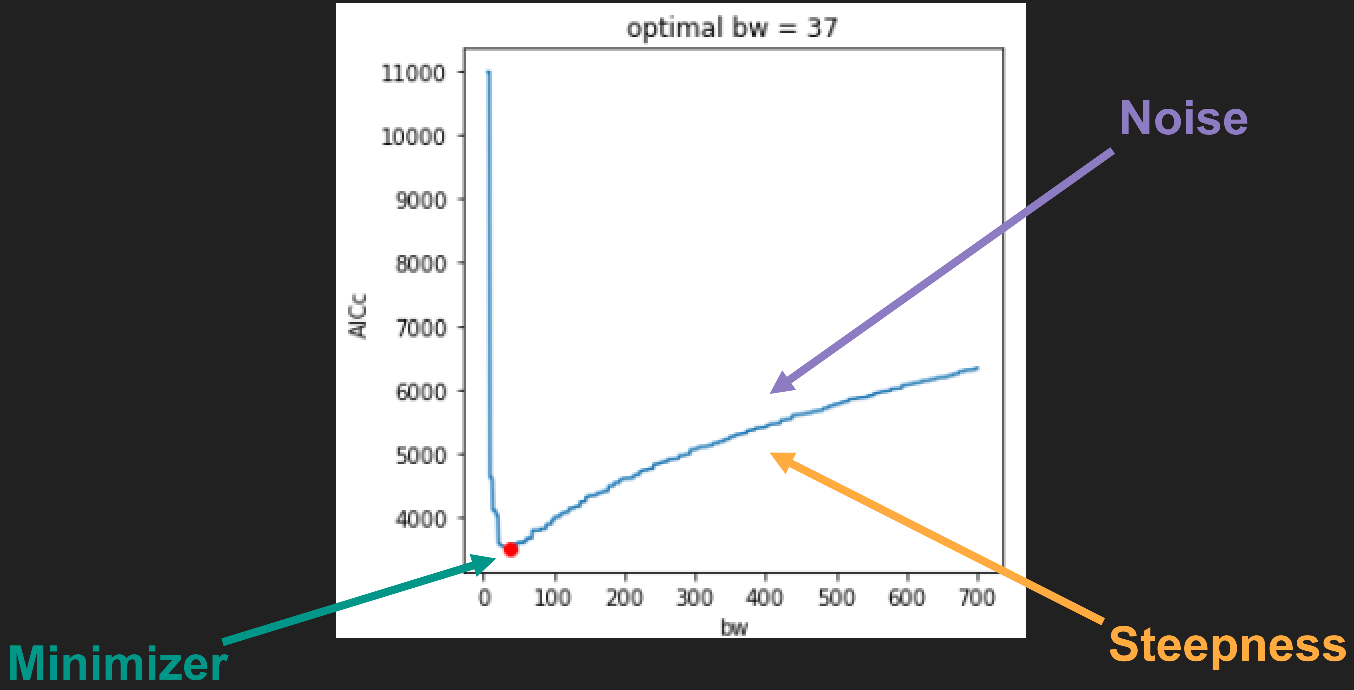
Therefore, the correction is a *decreasing* function of the bandwidth.

Problem structure: The GWR correction



The correction is a *decreasing* function of the bandwidth.

Problem structure: Example



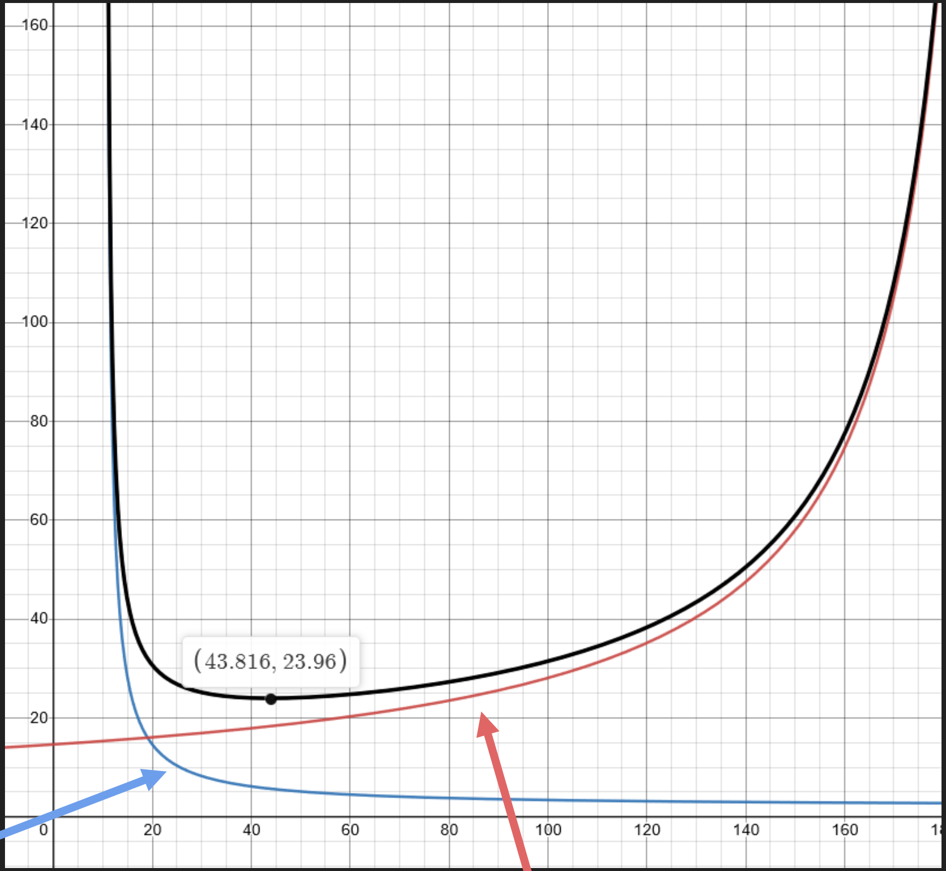
Component 1: **Minima**

- Minimizers of an AICc curve indicate appropriate bandwidths for analysis.
- The absolute minimizer is the best bandwidth for analysis.
- If the absolute minimizer is on the boundary, the process is local (left bound) or global (right bound).
- **Multiple minimizers are evidence of the MAUP, bandwidths where model fit improves but doesn't outpace the complexity.**

Component 2: **Steepness**

- Since the correction is fixed and monotonic, steepness of the curve indicates process variation.

- If the curve is **very flat** then the bandwidths could equally work within that set. Low **process variation** within that region slowly pushes the AICc to global.



Correction

log-likelihood

Component 3: **Noise**

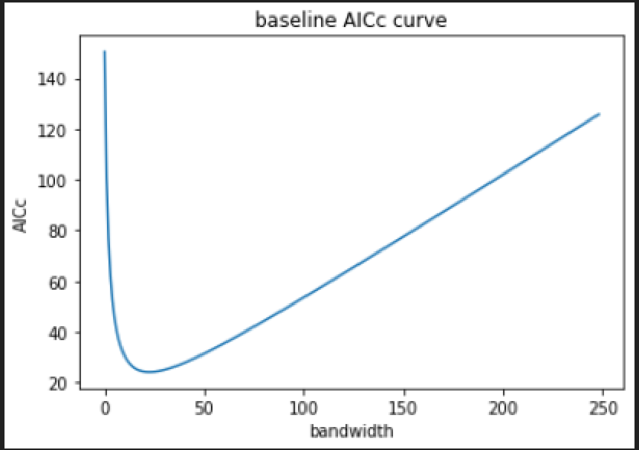
- Noise in the curve is natural for real datasets due to the calibration process.

- At what point are we looking at **noise, a local min, or the absolute min**?

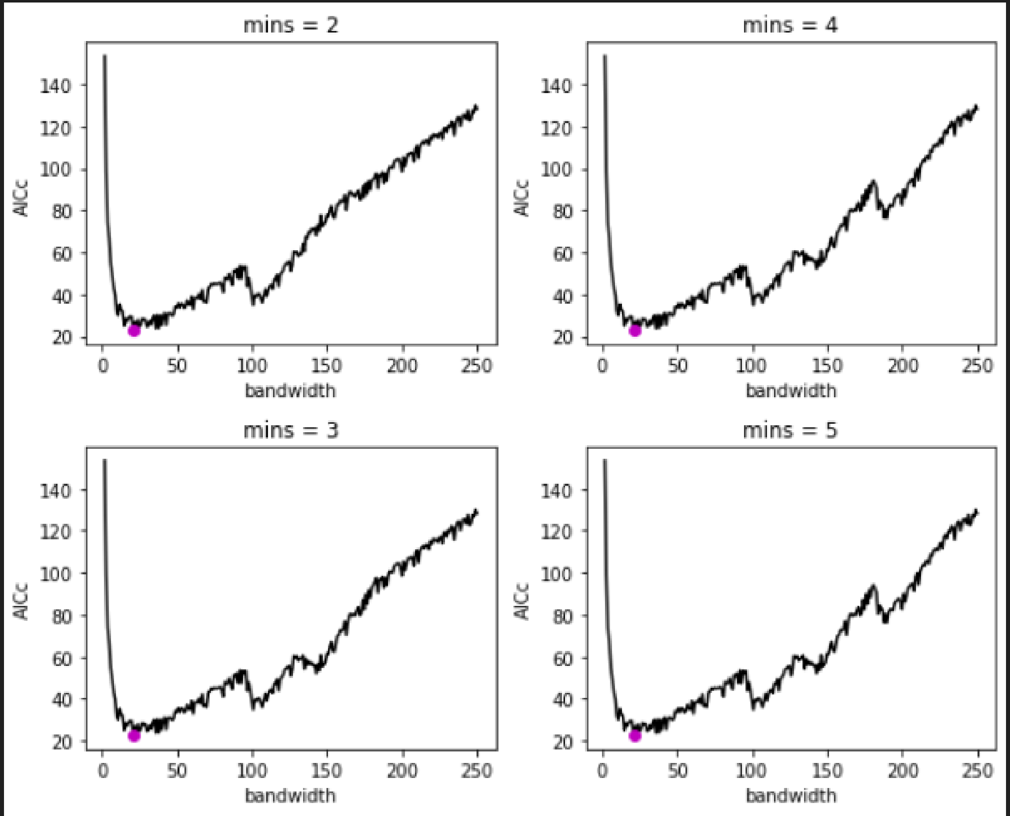
Numerical Experiments: Setup

- Tested grid search, golden section search, Brent's method, and particle swarm optimization (PSO)

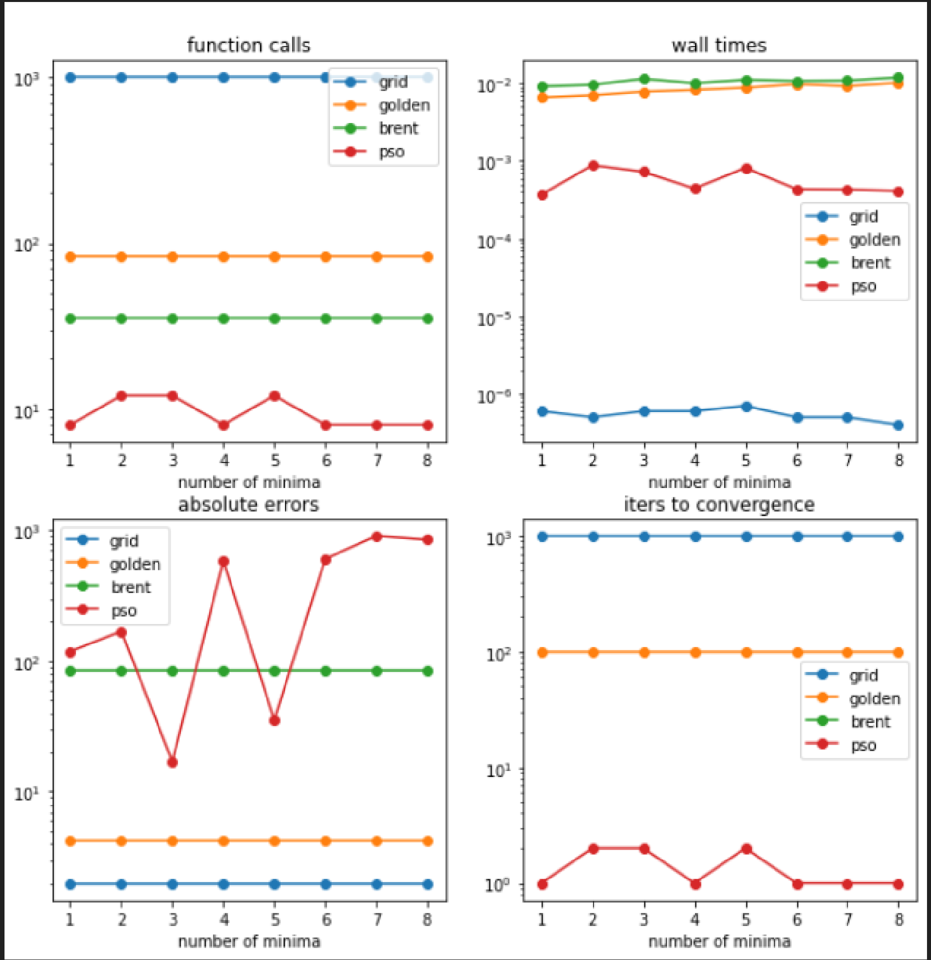
- Created a variety of sample AICc curves and added deterministic "noise" using hyperchaotic dynamics (the Generalized Henon map)



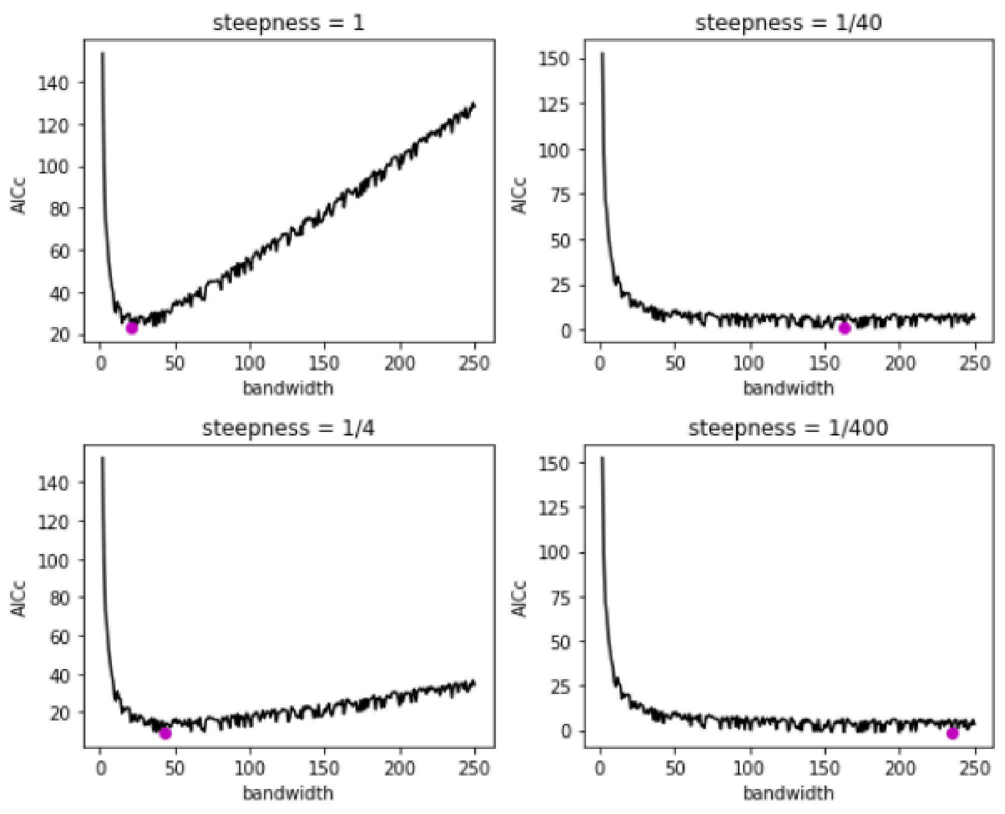
Objects of study: Minima



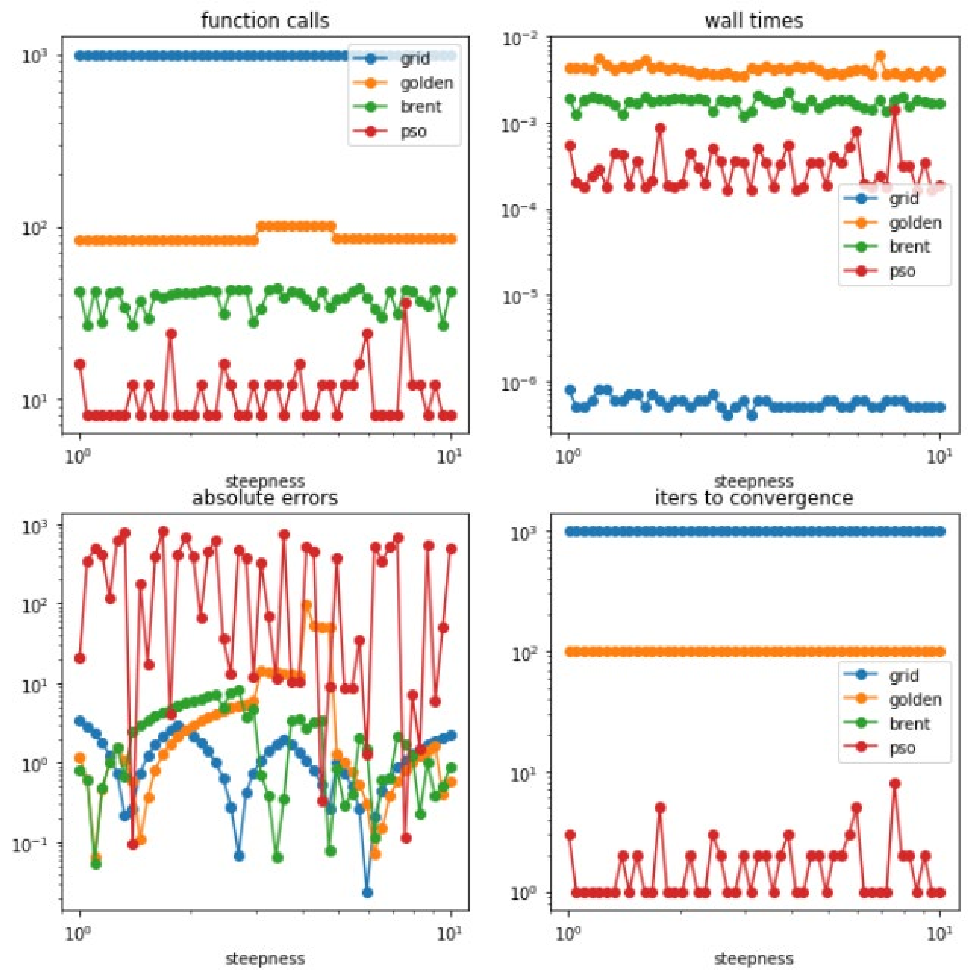
Numerical Results: Minima



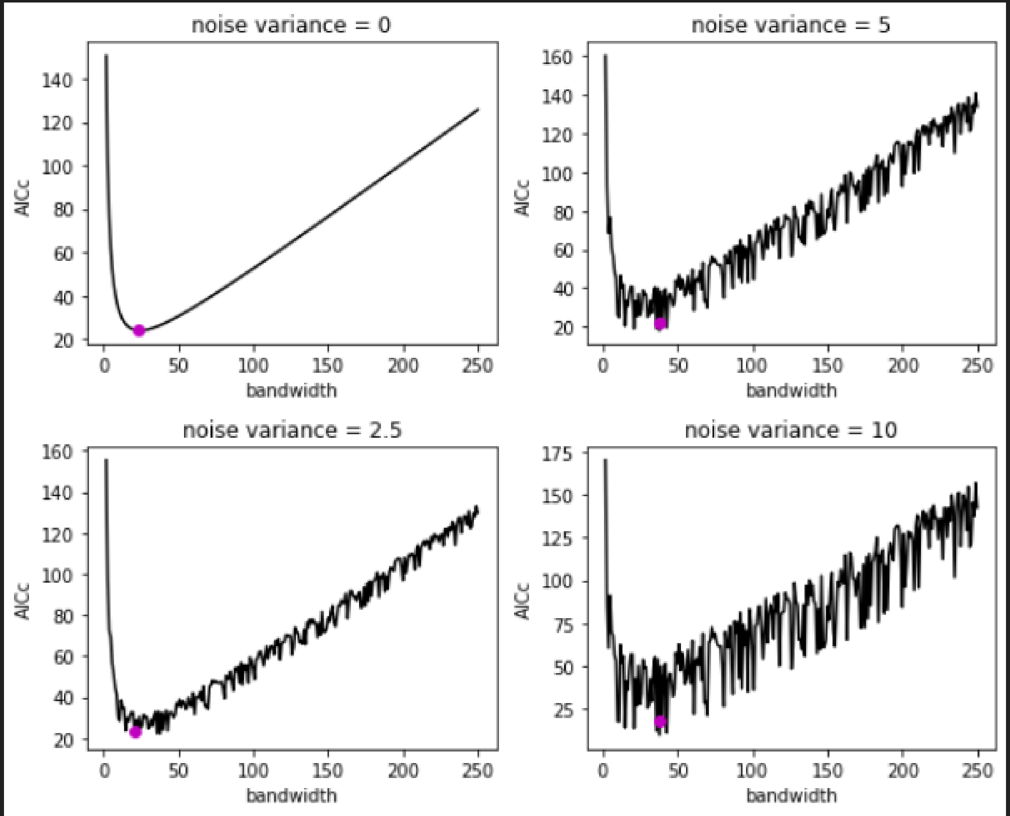
Objects of study: **Steepness**



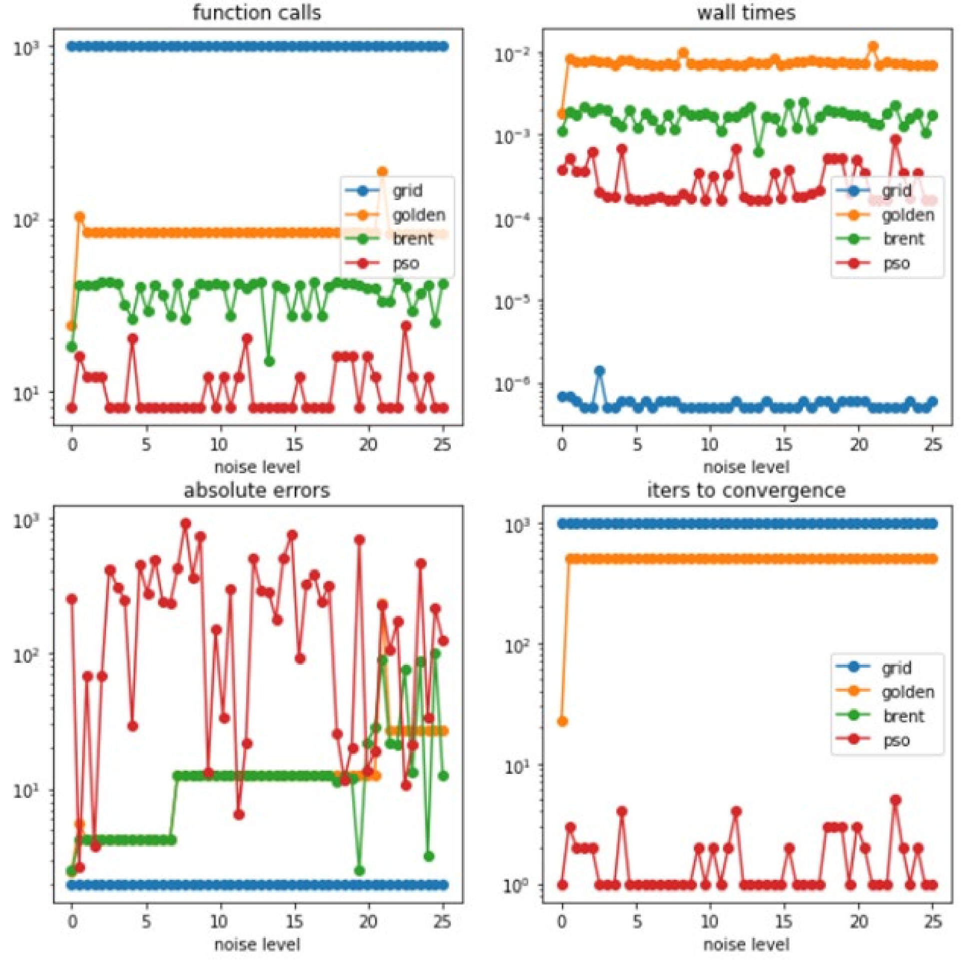
Numerical Results: **Steepness**



Objects of study: **Noise**



Numerical Results: **Noise**



Discussion and conclusions

- PSO is not very robust to multiple local minima and has a hard time distinguishing them from noise
- PSO is very efficient in terms of function calls but needs tweaking for accuracy
- We do see some accuracy/complexity tradeoffs in golden section search and Brent’s method

Next steps

- Add several PSO implementations and hyperparameter tuning / smoothing
- Flesh out these lines of inquiry with more exhaustive experiments