# In-class assignment # 3

Brian O'Shea,
PHY-905-003, Computational Astrophysics and Astrostatistics
Spring 2017

**Instructions:** We're going to use the functions you wrote for the Bisection and Newton's method, as well as some of the additional root finding methods in the SciPy `optimize` package (specifically, the methods for scalar functions).

Your goal is straightforward: try to come up with multiple functions f(x) that have one or more roots in an interval $[a, b]$ and which break one or more of the following methods: the Bisection Method, Newton's Method, the Secant Method (which is Newton's method with a numerically-computed derivative), and Brent's Method. In this case, "break" simply means to force the method to not find an existing root in that interval.

Experiment with several functions, intervals $[a, b]$, and all of the methods described above. What are the characteristics of functions or chosen intervals $[a, b]$ that tend to break root-finding algorithms? Are some algorithms more robust than others to choice of function and/or interval? Are any of these algorithms effectively unbreakable? Make some notes in `ANSWERS.md`, and we'll also discuss it in class.

**What to turn in:** Turn in `ANSWERS.md`, any source code you wrote, any plots you created (and the scripts you used to create them). **Do not** turn in object files or executables!