

Project 3: Root Finder (Newton's Method) Due 10/31/24
– Stephen Lepp

Write a “safe Newton’s Method” in python. The method should have the signature

```
safenewton(fn, start1, start2, err)
```

which is the same as our secant routine.

The function will use Newton’s method beginning at the point start1, with the provision that if a Newton’s step fails it will instead use bisection.

General outline is as follows. Take a Newton’s step at midpoint to generate an estimate of the root called guess. If guess is between start1 and start2, then evaluate function at guess and replace either start1 or start2 with guess. If guess is outside range of (start1, start2) then use $(start1 + start2)/2$ as guess instead of Newton’s step. Repeat until two successive guesses are within err.

note: (start1,start2) means between but not including start1 and start2.

- 1) Your program should work for any start1, start2 that contain a simple root (i.e. $f(start1)$ is opposite sign to $f(start2)$).
- 2) start1 does not need to be smaller than start2.
- 3) If $f(start1)$ is the same sign as $f(start2)$ print an error message and exit.

Is this routine sure to converge?

Can you find a function for which this is worse than ‘Bisection’? worse than ‘Secant’?

What modifications would you make if the user supplied another function with the derivative?