Astro 503 Homework #9

Due Tuesday, March 21

1. **Calculate your mortgage:** Mortgages and other *annuities* are structured so that you contribute a fixed payment P at each time interval (monthly, let's say), with a total amount A of principal. At the first payment the principal is $A_1 = A$; subsequently the amount of principal owed goes down (we hope) as

$$P = xA_i + (A_i - A_{i+1}). \tag{1}$$

   The first term is the interest at rate x per month, the second term is the contribution to reduction of principal. The payment P is chosen so that after the Nth payment, the principal is $A_{N+1} = 0$ and the loan is paid off.

   (a) Derive an equation relating x, P, A, and N.

   (b) Now write code to implement one of the two-point root-finding routines which we discussed in class (bisection, secant, false-position, secant with Aitken's extrapolation) and another code that implements Newton-Raphson. You may adapt the Numerical Recipes routines into good C++ if you like.

   (c) Then apply the code to the annuity formula to ask the following question: If I want to pay off a \$300,000 mortgage with monthly payments of ≤ 2000 over 30 years, what is the highest interest rate I can afford?

   Choose some sensible bracketing range for the root by hand. Then run the 2-point code and your Newton-Raphson code, and report how many iterations it takes each of them to determine x to an absolute accuracy of 10^{-5} or better.