For the following two problems, write and debug MATLAB codes and make sure they run with the test autograder from the course web page. Test them thoroughly on test cases of your own design including simple roots, multiple roots with sign change, and closely separated roots. When you are convinced they work, submit your codes together with

- brief discussion of any design decisions
- brief comparison of test results with theory

**Part 1** Implement a MATLAB function `findbracket.m` of the form

```matlab
function [a, b] = findbracket(f, x0)
% f: function handle f(x) to find a zero for
% x0: starting point / center of interval containing a zero
to try to find an initial interval [a, b] containing the input x0 and bracketing a zero of f(x), i.e. with sgn f(a) ≠ sgn f(b).
Your function should begin with a = b = x0 and a step size δ = 2−k chosen so that
\[ f(x_0 - \delta) < f\left(x_0 - \frac{\delta}{2}\right) = x_0 \]
(i.e. the unit of least precision for x0). While sgn f(a) = sgn f(b), decrease a by δ, increase b by δ, evaluate f(a) and f(b). and double δ.

**Part 2** Implement a MATLAB function `schroderbisection.m` of the form

```matlab
function [r, h] = schroderbisection(a, b, f, p, t)
% a: Beginning of interval [a, b]
% b: End of interval [a, b]
% f: function handle [ f, fp, fpp ] = f(x, p) to find a zero for
% should return f, f’ and f’’ at x
% p: parameters to pass through to f
% t: User-provided tolerance for interval width
which combines the fast convergence of the Schröder iteration for multiple roots

\[ g(x) = x - \frac{f(x)}{f'(x)} \frac{1}{1 - \frac{f(x)f''(x)}{f'(x)^2}} = x - \frac{f(x)f'(x)}{f'(x)^2 - f(x)f''(x)} \]
with the bracketing guarantee of bisection. At each step j = 1 to n, carefully choose p as in geometric mean bisection (watch out for zeroes!). Define

\[ \epsilon = \min(||f(b) - f(a)||/8, ||f''(p)|| |b - a|^2) \]
Apply the Schröder iteration function g(x) to two equations f±(x) = f(x)±ε = 0, yielding six candidates q± = g±(p), a± = g±(a), b± = g±(b). Replace [a, b] by the smallest interval with
endpoints chosen from \( a, a_+, a_-, p, q_+, q_-, b_-, b_+, b \) which keeps the root bracketed. Repeat until a \( f \) value exactly vanishes, \( b - a \leq t \), or \( b \) and \( a \) are adjacent floating point numbers, whichever comes first. Return the final approximation to the root \( r = (a + b)/2 \) and a \( 6 \times n \) history matrix \( \mathbf{h}[1:3, 1:n] \) with column \( \mathbf{h}[1:3, j] = (a, b, f(p)) \) recorded at step \( j \).

**Code Submission:** Upload the MATLAB files `findbracket.m` and `schroderbisection.m` and any supporting files to bCourses for your GSI to grade.