

Remember to indent the bodies of your IF-THEN and looping structures.

For the problems below that require you to implement the bisection method you should use a value of 10^{-10} for each of the tolerances. Note, in order to assign a value of 10^{-10} in MATLAB, you should do

```
tol = 1.0e-10
```

If you enter

```
tol = 10.0e-10
```

then your tolerance will be 10^{-9} . Also, you should allow a maximum of 50 iterations.

- 1) (3 pts) Write a function to implement the secant method. Use the approach from Problem 2 of HW 18.
- 2) (8 pts) Repeat Problem 3 of HW 18. Use the secant method to find the roots of the three problems below using the indicated initial guess:
 - a) $x - 2^{-x} = 0$, $x_1 = 0, x_2 = 2$.
 - b) $\cos t - t = 0$, $t_1 = 0, t_2 = 1$.
 - c) $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$, $x_1 = 1, x_2 = 3$.

You should only have 1 driver script that runs all 3 cases.

For each case, you should print out the value of the root, the value of the function at the root, the number of iterations, the status variable and create plots of the convergence histories. How do the iteration counts for the secant method compare to those from the bisection and Newton's methods?

- 3) (2 pts) Polynomial functions have special properties that can be exploited in order to find their roots (both real and complex). MATLAB has a built-in function called `roots` for computing the roots of polynomials. To use this function, you need to create a vector of polynomial coefficients in descending order. Use 0 for missing coefficients. For example, to compute the roots of $3x^2 + 1 = 0$ you would do

```
p = [3 0 1];
r = roots(p)
```

Use the `roots` function to compute the roots of the polynomials below

- a) $x^6 - 3x^2 + 1 = 0$.
- b) $7x^7 + 6x^6 + 5x^5 + 4x^4 + 3x^3 + 2x^2 + x + 1 = 0$.
- c) $ix^3 - ix^2 + 1 = 0$.

Note that i is the imaginary unit, $i = \sqrt{-1}$. Why are the roots for part c not complex conjugates?