Program #1 Simultaneous Root Finding **11.25%**  
Due: 09/21/2016  
Assignment:

In this program, you will use the **modified secant method** to simultaneously identify roots for several equations.

a(w, x, y, z) = 9\*w + 3\*x - 17.2  
b(w, x, y, z) = log(y) + x3 - 4  
c(w, x, y, z) = z1.6 - 4.72\*w6.2 - 0.8  
d(w, x, y, z) = w + cos(x) + y/7.4 + 0.5\*z - 11

One simultaneous zero is (w = 1.54197, x = 1.10742, y = 14.0398, z = 14.2276).

**Simultaneous Root Finding**

Finding derivatives when you have a function of more than one variable, such as a(w, x, y, z), is complicated by the fact that you must numerically compute **partial derivatives**. For example, **∂a/∂y** evaluated at (x, y, z) can be approximated by **forward difference** as:

(a(x, **y + dy**, z) - a(x, y, z))/dy where **dy** represents a small change in y (small step size).

The basic task is to transform equations **6.23a and 6.23b on page 170** into a matrix equation (Ax = b). Then it is just a matter of solving the matrix equation for the vector x to obtain the variable values for the next root finding iteration (that is, the vector x contains the variables with subscript i + 1). Use the **modified secant method** to perform your root finding, adapted for multiple variables and multiple equations. Terminate your search using the usual error approximations, but modified for multiple variables (**do something that makes sense**). Remember to plug your solution back into the original equations to make sure that you get simultaneous zeroes.

The method signature for your modified secant method is as follows (use **Secant.cpp**):

double\* multipleEquationModifiedSecant(MultiVarAndEqnsFunction\* f, int num\_eqns, double\* xy\_array, double\* delta\_xy\_array, double tol, int max)

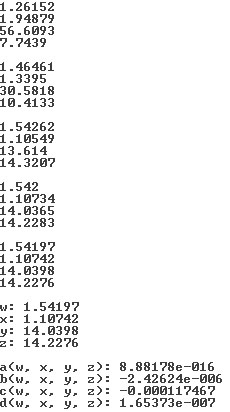
where f represents a pointer to the struct that will be used to evaluate the function, num\_eqns is the number of simultaneous equations, **xy\_array** is an array of initial values for each variable, **delta\_xy\_array** is an array of small increment sizes for each variable, **tol** is the required tolerance for the computation, **max** is the maximum number of iterations in case the modified secant method is oscillating or diverging.

Do not assume four variables and four equations. **Make your solution as general as possible**, allowing for any number of variables and the same number of equations.

**Solving Ax = b**

To find the variable values for the next iteration, it is necessary to solve the matrix equation Ax = b for x. One way to do this is to invert A (x = A-1b). Instead, use the **Gauss-Jordan** method to solve the matrix equation.

**Example**



**Use your program to solve Problem 8.42 on page 220**. Note that this problem has two equations and two unknowns.