Introduction to Scientific Computing with Fortran 90
Homework I.1

Due: W 1/20

1. The goal of this problem is to write a code which computes the real roots of a general quadratic \( ax^2 + bx + c \) by implementing the quadratic formula

\[
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.
\]

A first draft of the code is found in `quadratic_formula.f90`.

a. Find the compile time errors in the code `quadratic_formula.f90`.
b. Run the code for a quadratic such as \( x^2 + x - 2 = (x - 1) \ast (x + 2) = 0 \). As you can see, if does not get the correct answers. Find the logical error in the code and fix it.
c. Add read statements to the code to enter the coefficients of the quadratic \( ax^2 + bx + c \) from the screen.
d. Try your program on \( x^2 + 4 \). What happens and why? Add a test (a conditional statement) to avoid this problem; add an appropriate print statement.
e. Make sure that your code is working properly by trying it on quadratic expressions with a single root, two real roots and two imaginary roots.
f. Make sure that your final code has comment statements in appropriate places, that any new variables you defined are named in a meaningful manner and that you don’t use mixed arithmetic.

2. In calculus we learned to determine the path of a projectile that is fired with an initial velocity of \( v_0 \) ft/sec and at an angle with the horizon of \( \theta \). In particular, using the earth’s gravitational field of 32 ft/sec\(^2\) we found that after \( t \) seconds the position of the projectile is given by

\[
x = v_0 t \cos \theta \quad y = v_0 t \sin \theta - 16t^2
\]

a. We first want to write a code which reads in \( v_0 \), \( \theta \), and \( t \) from the screen, writes them out and then computes and outputs the \((x, y)\) position of the projectile at the given time. The source code `projectile.f90` is a start on this code. Complete the code and output your results for the choice \( v_0 = 240 \text{ ft/sec}, \theta = 45^\circ \) and \( t = 2 \) seconds.
b. Make a copy of your file and rename it, say `projectile2.f90`. Instead of reading in the time, we now want to output the position for a range of times. This means we need to add a do loop. Delete your statements for reading in and printing the time. Add a loop which prints out the time and position for 10 equally spaced times starting at \( t = .2 \) and ending at \( t = 2 \). Hint: \( t \) is not used in your do loop counter, rather you just want to use do \( i = 1, 10 \) and increment \( t \) inside of the loop. Run your code for the values given in (a).

Submit the final version of each code which incorporates all of your modifications. Both codes should be emailed to both Michelle and Professor Peterson.