Project 1: Roots of a quadratic equation

A time honoured programming exercise. This is not computational physics but the idea is for you to get gain more experience with PYTHON by designing, writing and testing a simple program.

Due Dates

Week 3, Monday at 4:59 Your submission should be a single zip file containing all the .py files containing your code as well as a Single typed PDF document with your test results and answers to the problems/questions. Word documents will not be accepted.
Your zip file should be in the form <lastname>_project_<N>.zip where you replace ¡lastname¿ with your last name and ¡N¿ with the project number.
Upload your zip file via the link provided on the project webpage found at http://urminsky.ca.

Project Briefing

In physics we often need to solve quadratic equations of the form

$$ax^2 + bx + c = 0$$

As an example, we can consider the simple kinematics problem of a ball thrown vertically upwards at time $t = 0$ with an initial velocity v_0. If we ignore air resistance (well deal with that later in the course) then after time t, the ball has reached a height h, given by the standard kinematics formula

$$h = v_0 t - \frac{1}{2} g t^2$$

where $g = 9.81$ m/s^2, is the acceleration due to gravity. If we want to find the time at which the ball reaches a certain height, we need to solve the quadratic for t (i.e., we need to find the roots of the equation). There is a standard formula for the roots of a quadratic, but rather than having to remember it and implement it anew each time, it is useful to have a program that does the job.

Your task is to design, write and test a Python script that prints the roots of the quadratic equation $ax^2 + bx + c = 0$ using the standard formula. (Your first job is to look this up, if you dont have it memorized!)

The source code for the function should be saved in a file named quadratic.py (This is one of the files you should upload, when finished.)

When executed, the user should be prompted to input values for the coefficients a, b, and c.
Your code should

- not use any pre-existing python functions (eg. np.roots() from the numpy library).
- Allow for special cases, such as $a = 0$ and/or b=0.
- print the values of the roots to the screen

Documentation: Make sure that your code is well commented so that it can be easily be understood by almost anyone, including yourself, if you look at it again at the end of the course (when youve forgotten how you coded it).

Testing: Test your code for several sets of coefficients. Does it give the correct results in all cases? Does it handle special cases properly (e.g., a = 0)? What about cases where the roots are complex numbers?

Using the program: Use your program to find the elapsed time in the following situations.

a. A ball is shot vertically upwards from ground level at $t = 0$ with an initial velocity $v_0 = 30$ m/s.

- What are the value(s) of t when the balls height is $h = 25$ m?
- What is the total time of flight from launch to landing?

b. The ball is dropped at time t = 0 from a height h = 100 m.

- how long before it hits the ground?

For each case, what is the physical meaning (if any) of both roots?

There can be issues with numerical accuracy of one of the roots when $b2 >> 4ac$. (Which root is problematic depends on the sign of b.) There is an alternate way to write the root so that it is not prone to such numerical inaccuracy and then a general way to implement the calculation of the solution to the quadratic equation so that the most appropriate form is used for each of the roots. Can you find it? (Try http://mathworld.wolfram.com/). Make a new version of your function that calculates the roots in this way. (Dont overwrite the original function save the source code in a different file quadratic2.py which should be upload in the submitted zip file.

Experiment with both functions, fixing a and c, but varying b.

- Do both functions produce the same results?
- How large does the value of the ratio $b^2/4ac$ have to be before the two functions give significantly different answers? (Make up a sensible quantitative definition of significantly different.)