Minor Assignment 02

ENGINEER 1D04
Dr. William Farmer and Dr. Spencer Smith
McMaster University, Winter 2012

Revised: June 20, 2013

Please use AutoMarker (`automarker.mcmaster.ca`) and Avenue to acquire, test, package and submit your assignment. The procedure for submitting assignments is summarized on Avenue, with additional details provided by AutoMarker. **Please frequently back up your work by creating a submission package in AutoMarker**. This will provide a chance to recover your work in the event of an equipment failure.

Background

The formula for computing the future value of an investment with *compound interest* is

$$p\left(1+\frac{r}{n}\right)^{nt} \tag{1}$$

where

- p is the initial principal.
- r is the interest rate.
- n is the number of times the interest is compounded per year.
- t is the time expressed as a number of years.

The formula for computing the future value of an investment with *continuous compound interest* is:

$$\lim_{n\to\infty} p\left(1+\frac{r}{n}\right)^{nt} = pe^{rt} \tag{2}$$

(Notice the similarity between this limit and the limit given in Assignment 1.)

Overview

Banks sometimes offer continuous compound interest, which is, roughly speaking, compounding interest infinitely often. The purpose of this assignment is to write a Python program that computes the future values of investments

with compound interest and with continuous compound interest and then compares the difference between them.

Design, implement, and test a program that satisfies the requirements below.

****Important**:** This assignment will be run through an automated testing program to be graded. The program will make the assumption that any output containing an equals sign (=) is an answer to be marked. **Do not** print output containing the equals sign, except where specified in the requirements below. Outputs must also be printed on separate lines. Additionally, input and output statements must be **exactly** in the order specified. Failure to precisely follow the requirements below will result in a significant loss of marks.

Requirements

1. The program asks the user to enter the following amounts in the following **order**:
    a. The initial principal (p).
    b. The interest rate (r).
    c. The number of times the interest is compounded per year (n).
    d. The time expressed as a number of years (t).
2. Using these values, the program computes:
    a. A_1 = the future value of p with compound interest, given in Equation (1), calculated using either the **pow** function from the **math** library or **.
    b. A_2 = the future value of p with compound interest, given in Equation (1), calculated using a **for** statement. ($p * \prod_{i=1}^{nt}(1 + \frac{r}{n})$)
    c. B = the future value of p with continuous compound interest using the formula given in Equation (2).
3. The program prints the following values with appropriate labels:
    a. A_1.
    b. Value of A_2 for **each** compounding period (A_2 everytime for-loop is executed)
    c. B.
    d. $100 \cdot \frac{|A_1 - A_2|}{A_1}$, which is the *relative difference* between A_1 and A_2 expressed as a percentage.

e. $100 \cdot \frac{|A_1 - B|}{B}$, which is the *relative difference* between A_1 and B expressed as a percentage.

Do not print $ or % with your outputs.

4. The program prints to Python Shell the value of each term in the sequence in the format "*label* = `value`," where *label* is an appropriate description of `value` and "=" is the equals sign.

5. The program is written in Python in a module, NOT in the Python Shell. To create a new module in IDLE, go to File → New Window. You must save this file with a .py extension. For more information on submitting your program, click the "AutoMarker Instructions" button above.

6. Your name, MacID, student number, and the date are given in comments at the top of your Python (.py) file before your program.

7. Your answers to the design questions and the test plan (see below) are given in comments at the bottom of your Python (.py) file after your program.

8. Your program MUST have valid Python syntax and it must run without errors. Ensure that your program runs properly by running it before you submit.

9. You must sign out with a TA or IAI after you have submitted your lab at the submission station. Failure to do so could result in a zero.

   item You must sign out after you've submitted your lab at the submission station. Failure to do so will result in a zero.

Design and Implementation Instructions

1. Use the `futval.py` program from Chapter 2 as a **starting point**.

```
# futval.py
# A program to compute the value of an investment
# carried 10 years into the future


print "This program calculates the future value"
print "of a 10 year investment"

principal= input("Enter the initial principal: ")
apr= input("Enter the annual interest rate: ")

for i in range(10):
    principal= principal*(1+apr)
    print "The value in 10 years is:", principal
```

2. Use the `exp` function from the `math` library to compute B.

Design Question

Does your program enforce n and t to be integers? If so, why is this necessary? If not, is this a problem?

Testing Plan

Produce a test plan in the following form:

```
Test i
Input: [p, r, n, t]
Expected Output: [A1, A2, B, d1, d2]
```

where `i` is the test number, `p`, `r`, `n`, `t` are the four inputs for the program and `A1`, `A2`, `B`, `d1` and `d2` are the outputs the program is expected to produce. The values of `d1` and `d2` are the relative differences calculated.

Example:

```
Test 1
Input: [100, 0.5, 4, 5]
Expected Output: [1054.51, 1054.51, 1218.25, 0.0, 13.44]

Test 2
...
```

Please include at least two tests in your test plan. Use your test cases to build your confidence in the correctness of your code before doing a test run with AutoMarker.