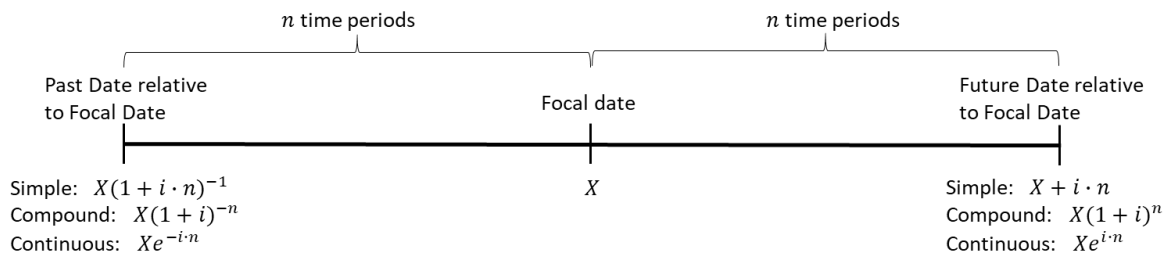


# Financial Mathematics

## A Concise Guide

2<sup>nd</sup> Edition



by **Stephen Fratini**

## Table of Contents

|   |    |
|---|----|
| List of Figures .....                                     | 6  |
| List of Tables .....                                      | 7  |
| Preface to Second Edition .....                           | 8  |
| Acknowledgements for First Edition .....                  | 9  |
| 1 Introduction.....                                       | 11 |
| 1.1 Purpose .....   | 11 |
| 1.2 Intended Usage.....                                   | 11 |
| 1.3 Prerequisites .....                                   | 11 |
| 1.4 Conventions .....                                     | 11 |
| 1.5 Outline .....   | 12 |
| 2 Background.....   | 13 |
| 2.1 Progressions.....                                     | 13 |
| 2.1.1 Arithmetic Progressions .....                       | 13 |
| 2.1.2 Geometric Progressions .....                        | 13 |
| 2.1.3 Recursive Progressions .....                        | 14 |
| 2.2 Supporting Programs .....                             | 15 |
| 2.2.1 Arithmetic Progression Calculator .....             | 15 |
| 2.2.2 Geometric Progression Calculator .....              | 15 |
| 2.3 Exercises .....                                       | 16 |
| 3 Interest .....  | 17 |
| 3.1 Simple Interest and Comparison to a Ballon Loan ..... | 17 |
| 3.2 Compound Interest.....                                | 17 |
| 3.3 Continuous Interest .....                             | 18 |
| 3.4 Effective Rate of Interest .....                      | 19 |
| 3.5 Present Value.....                                    | 21 |
| 3.6 Discounting .....                                     | 21 |
| 3.7 Time Value of Money .....                             | 23 |
| 3.8 Interest, Inflation and Taxes.....                    | 25 |
| 3.9 Supporting Programs.....                              | 26 |
| 3.9.1 Interest Compounding Calculator.....                | 26 |
| 3.10 Exercises .....                                      | 27 |
| 4 Annuities.....  | 28 |

|         |   |    |
|---------|---|----|
| 4.1     | Terminology .....   | 28 |
| 4.2     | Accumulated Value of Ordinary, Simple, Certain Annuities .....          | 28 |
| 4.2.1   | Concept and Formulas .....  | 28 |
| 4.2.2   | Examples .....  | 29 |
| 4.2.3   | Blocking .....  | 30 |
| 4.2.3.1 | Example .....   | 30 |
| 4.2.3.2 | Generalization .....  | 31 |
| 4.3     | Discounted (Present) Value of Ordinary, Simple, Certain Annuities ..... | 31 |
| 4.3.1   | Concept and Formulas .....  | 31 |
| 4.3.2   | Examples .....  | 32 |
| 4.4     | Annuities Due .....   | 33 |
| 4.5     | Deferred Annuities .....  | 33 |
| 4.6     | Forborne Annuities .....  | 35 |
| 4.7     | General Annuities .....   | 35 |
| 4.7.1   | Concept and Interest Conversion Formulas .....                          | 35 |
| 4.7.2   | Example – Accumulated Value of a General Annuity .....                  | 36 |
| 4.7.3   | Example – Discounted (Present) Value of a General Annuity .....         | 36 |
| 4.8     | Perpetuities .....  | 37 |
| 4.9     | Continuous Annuities .....  | 37 |
| 4.9.1   | Discounted (Present) Value .....  | 37 |
| 4.9.2   | Accumulated Value .....   | 39 |
| 4.10    | Supporting Programs .....   | 39 |
| 4.10.1  | Block Annuity Calculator .....  | 39 |
| 4.10.2  | Annuity Payout Calculator .....   | 40 |
| 4.10.3  | Deferred Annuity Calculator .....                                       | 41 |
| 4.11    | Exercises .....   | 42 |
| 5       | Repayment of Loans .....  | 44 |
| 5.1     | Amortization of Loans .....   | 44 |
| 5.1.1   | General Formula and Equivalence to Annuities .....                      | 44 |
| 5.1.2   | Breakdown of Interest versus Principal Repayment .....                  | 45 |
| 5.1.3   | Rounding Mortgage Payments .....  | 46 |
| 5.2     | Balloon Loans .....   | 47 |
| 5.2.1   | Concept and Formulas .....  | 47 |

|       |  |    |
|-------|--|----|
| 5.2.2 | Example – Balloon Loan as a General Annuity .....                | 48 |
| 5.3   | Computation of Remaining Balance on a Loan .....                 | 48 |
| 5.4   | Refinancing a Loan.....  | 49 |
| 5.5   | Supporting Programs .....  | 51 |
| 5.5.1 | Annuity Payout Calculator .....                                  | 51 |
| 5.5.2 | Amortization Calculator .....                                    | 51 |
| 5.5.3 | Balloon Loan Calculator .....                                    | 51 |
| 5.5.4 | Loan Refinance Calculator .....                                  | 51 |
| 5.5.5 | Loan Refinance - Breakeven Interest Rate Analyzer .....          | 52 |
| 5.6   | Exercises .....  | 54 |
| 6     | Bonds.....   | 56 |
| 6.1   | Terminology.....   | 56 |
| 6.2   | Similarity between Bonds and Balloon Loans .....                 | 57 |
| 6.2.1 | School Bond Example.....   | 58 |
| 6.2.2 | Determine Purchase Price of Bond to Generate Various Yields..... | 58 |
| 6.2.3 | Bond Prices versus Yield .....                                   | 59 |
| 6.3   | Schedule for a Bond.....   | 59 |
| 6.3.1 | Schedule for a Bond Bought at a Premium.....                     | 60 |
| 6.3.2 | Schedule for a Bond Bought at a Discount .....                   | 61 |
| 6.4   | Correction if Bond Not Purchased on a Coupon Date.....           | 62 |
| 6.4.1 | Dirty Price of a Bond .....                                      | 62 |
| 6.4.2 | Dirty Bond Price Example.....                                    | 63 |
| 6.4.3 | Clean Price of a Bond .....                                      | 63 |
| 6.5   | Computing the Yield given the Bond Price .....                   | 66 |
| 6.6   | Supporting Programs .....  | 67 |
| 6.6.1 | Balloon Loan Calculator .....                                    | 67 |
| 6.6.2 | Bond Schedule Creator .....                                      | 68 |
| 6.7   | Exercises .....  | 69 |
| 7     | Capital Budgeting .....  | 71 |
| 7.1   | Net Present Value (NPV).....                                     | 71 |
| 7.1.1 | Concept with an Example .....                                    | 71 |
| 7.1.2 | Negative Cash Flows .....  | 74 |
| 7.1.3 | Multiple Investments.....  | 75 |

|       |   |     |
|-------|---|-----|
| 7.1.4 | Range of Cash Flow Estimates .....                      | 76  |
| 7.1.5 | Initial Investment is Loaned .....                      | 77  |
| 7.1.6 | Comparison of Projects .....                            | 78  |
| 7.2   | Internal Rate of Return (IRR) .....                     | 79  |
| 7.2.1 | Acme Furniture Example Revisited .....                  | 80  |
| 7.2.2 | Multiple IRR Values for a Given Cash Flow .....         | 80  |
| 7.3   | Modified IRR .....                                      | 81  |
| 7.4   | Payback and Discounted Payback Period .....             | 83  |
| 7.5   | Profitability Index .....                               | 84  |
| 7.6   | Computational Support via Microsoft Excel .....         | 85  |
| 7.7   | Exercises .....   | 86  |
| 8     | Python Programs .....                                   | 87  |
| 8.1   | Arithmetic Progression .....                            | 87  |
| 8.2   | Geometric Progression .....                             | 91  |
| 8.3   | Interest Compounding Calculator .....                   | 94  |
| 8.4   | Block Annuity Calculator .....                          | 96  |
| 8.5   | Annuity Payout Calculator .....                         | 99  |
| 8.6   | Deferred Annuity Calculator .....                       | 102 |
| 8.7   | Amortization Calculator .....                           | 105 |
| 8.8   | Balloon Loan Calculator .....                           | 110 |
| 8.9   | Loan Refinance Calculator .....                         | 113 |
| 8.10  | Loan Refinance – Breakeven Interest Rate Analyzer ..... | 117 |
| 8.11  | Bond Schedule Creator .....                             | 121 |
|       | Acronyms and Symbols .....                              | 125 |
|       | References .....  | 126 |
|       | Index of Terms .....                                    | 127 |

## List of Figures

|   |    |
|---|----|
| Figure 1. Effective Rate of Discount for a Given Time Period .....                            | 22 |
| Figure 2. Accumulating with a Discount Rate .....   | 23 |
| Figure 3. Equivalent Values to X in the Past and Future .....                                 | 24 |
| Figure 4. Debt and Payments Relative to a Focal Date .....                                    | 25 |
| Figure 5. Payment Schedule for Accumulated Value of Ordinary, Simple, Certain Annuities ..... | 28 |
| Figure 6. Payment Schedule for an Annuity Due .....   | 33 |
| Figure 7. Deferred Annuity .....  | 34 |
| Figure 8. Multiple Payments per Compounding Period .....                                      | 37 |
| Figure 9. Payouts timeline for a bond .....   | 58 |
| Figure 10. Bond Purchases between Coupon Dates.....   | 63 |
| Figure 11. Graph of Dirty Bond Prices over Time .....   | 65 |
| Figure 12. Graph of Clean Bond Prices over Time .....   | 66 |
| Figure 13. Graph of NPV as a Function of the Rate of Return .....                             | 81 |

## List of Tables

|   |    |
|---|----|
| Table 1. Computation of Final Principal using Compound Interest .....               | 18 |
| Table 2. Accumulated Value of an Ordinary, Simple, Certain Annuity .....            | 29 |
| Table 3. Interest-Principal Repayment Breakdown Example .....                       | 46 |
| Table 4. Interest Rate Analysis for Refinancing .....                               | 50 |
| Table 5. Bond Purchase Price versus Yield .....                                     | 59 |
| Table 6. Schedule for Bond Purchased at a Premium .....                             | 61 |
| Table 7. Schedule for Bond Purchased at a Discount .....                            | 62 |
| Table 8. Dirty Price Changes over Time for a Bond .....                             | 64 |
| Table 9. Clean Price Changes over Time for a Bond .....                             | 65 |
| Table 10. Balloon Loan – Bond Comparison .....                                      | 67 |
| Table 11. Answer to Exercise 6 .....  | 69 |
| Table 12. Initial Investment and Cash Flow in Half-year Increments .....            | 72 |
| Table 13. Investment Analysis for Acme Furniture .....                              | 73 |
| Table 14. Initial Investment, and Cash Flow in Half-year Increments - Revised ..... | 73 |
| Table 15. Investment Analysis for Acme Furniture - Revised .....                    | 74 |
| Table 16. Project Quarterly Cash Flows for a Resort Business .....                  | 74 |
| Table 17. Investment Analysis for Resort Investment .....                           | 75 |
| Table 18. Operating and Investment Cash Flows .....                                 | 75 |
| Table 19. NPV Analysis in the Case of Multiple Investments .....                    | 76 |
| Table 20. Optimistic-Pessimistic Cash Flows for New Restaurant .....                | 76 |
| Table 21. NPV for Optimistic and Pessimistic Scenarios .....                        | 77 |
| Table 22. Acme Furniture - Cash Flow Estimates with Loan Repayments .....           | 78 |
| Table 23. Investment Analysis for Acme Furniture with Loan Repayments .....         | 78 |
| Table 24. Cash Flows for Projects A and B based on Same Initial Investment .....    | 79 |
| Table 25. NPV Comparison for Projects A and B .....                                 | 79 |
| Table 26. Cash Flow with Two IRRs .....   | 80 |
| Table 27. Input for MIRR Example .....  | 83 |
| Table 28. DPP for Project A .....   | 84 |
| Table 29. DPP for Project B .....   | 84 |

## Preface to Second Edition

This book is targeted at people who are interested in learning about the mathematics behind some basic financial concepts, such as interest rates, annuities, loans, bonds and capital budgeting, as well as trained mathematicians looking to apply their skills to finance. The target audience includes individual investors, small business owners, and students of business and finance.

Derivations for most of the stated formulas are provided in the text. The assumed mathematical background is minimal (basic algebra and just a touch of the limit and integral concepts from calculus). For more extensive coverage of basic financial mathematics, the reader is referred to the textbooks by Brown and Kopp [2], Broverman [3] and Alhabeeb [4].

**[Author's Remark:** In my opinion, the financial concepts are more complex than the mathematics used in this book.] By the way, the convention of using “author’s remark” is used elsewhere in the book when I state a personal opinion or add a note that is not in the flow of the discussion.

Many of the financial processes discussed in this book are supported by Python programs which are listed at the end of the book. The author has also made freely available at the following Github site:

[https://github.com/sfratini33/art-of-managing-things-external/tree/master/fin\\_math/code](https://github.com/sfratini33/art-of-managing-things-external/tree/master/fin_math/code)

Further, some of the computations concerning capital budgeting are supported by Microsoft Excel financial functions such as NPV, IRR and MIRR. The associated Excel spreadsheets (supporting examples in this document) are also made available at the above mentioned Github site.

While this document does provide some background and description for each Python program, it is not intended to be a Python tutorial. However, the reader should be able to use the Python programs without understanding Python. The programs were developed on the open source Anaconda software environment using the Jupyter Notebook (which is a browser-based development platform for Python), see <https://www.anaconda.com>. There are several good online tutorials on Python (see <https://www.w3schools.com/python/> or <https://www.tutorialspoint.com/python/index.htm>).

The second edition includes some minor corrections to the first edition. Also, the Python programs are now included in the body of the book (see Section 8).



## Acknowledgements for First Edition

The author wishes to thank Jenny Huang and James Swigart for their helpful review and comments on this book.

Stephen Fratini  
Sole Proprietor of The Art of Managing Things  
Eatontown, New Jersey (USA)  
Email: [sfratini@artofmanagingthings.com](mailto:sfratini@artofmanagingthings.com)  
LinkedIn: [www.linkedin.com/in/stephenfratini](https://www.linkedin.com/in/stephenfratini)

Copyright © 2024 by The Art of Managing Things

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the expressed written permission of the author except for the use of brief quotations in a book review.

**Other books by the author:**

- *The Art of Managing Things (2<sup>nd</sup> edition)*, self-published on Amazon, <https://www.amazon.com/Art-Managing-Things-Stephen-Fratini-ebook/dp/B07N4H4YWH/>, January 2019.
- *Mathematical Thinking: Exercises for the Mind (2<sup>nd</sup> Edition)*, self-published on Amazon, <https://www.amazon.com/dp/B0CL34FRP1>, October 2023.
- *Math in Art, and Art in Math*, self-published on Amazon, <https://www.amazon.com/dp/B091D1F8MB>, March 2021.
- *Algebra through Discovery and Experimentation*, self-published on Amazon, <https://www.amazon.com/dp/B09B5L9WL5>, July 2021.
- *The Struggle Against Chaos*, self-published on Amazon, <https://www.amazon.com/dp/B09BLPQ86Q>, July 2021.
- *Mathematical Vignettes: Number theory, stochastic processes, game theory, cryptography, linear programming and more*, self-published on Amazon, <https://www.amazon.com/Mathematical-Vignettes-stochastic-cryptography-programming-ebook/dp/B0BBP1PBJQ/>, August 2022.
- *Learning Math through Puzzles: Number properties, counting, sequences and series, algebra, functions, and mathematical reasoning*, self-published on Amazon, <https://www.amazon.com/dp/B0BZFRZP5B>, March 2023.
- *Mathematical Vignettes: Volume II: Topics from combinatorial design, magic squares, finite geometry, abstract algebra, error correcting codes, geometric packing problems and much more*, self-published on Amazon, <https://www.amazon.com/dp/B0CM1CLSK8>, October 2023.
- *Shape Up and Solve It! – Learn Geometry Through Puzzles*, self-published on Amazon, <https://www.amazon.com/Shape-Up-Solve-Geometry-Through/dp/B0CRVHHMJB/>, January 2024.

Electronic versions of my books are available (free of charge) at <https://www.artofmanagingthings.com/home/my-books>.

# 1 Introduction

## 1.1 Purpose

The purpose of this book is to provide the mathematical basis for several fundamental financial concepts, i.e., interest compounding, annuities, loans, bonds and capital budgeting. A secondary purpose is to provide the reader with some simple Python programs to assist with the associated financial calculations. [**Author's Remark:** While the Python programs do some checking for data entry errors, they are not production level. Feel free to make modifications to the code and use elsewhere. I only ask that you give attribution to the source (i.e., me) if you make commercial use of my programs.]

## 1.2 Intended Usage

This book can be used in several ways:

- for self-study by those wanting to learn about the mathematics behind some basic financial concepts
- as a supplement to a business or finance course for advance high school or college students
- as a quick reference to basic finance formulas and associated procedures.

## 1.3 Prerequisites

As noted in the preface, the mathematical prerequisites are minimal (basic algebra). The summation of a series of numbers is a common theme in the book (the basics concerning series are covered in Section 2). From a financial perspective, very little background is assumed but the reader is warned that some of the concepts are subtle and will require some reflection.

To use the Python programs, one needs to know how to install Anaconda (or a similar environment for executing Python code). The provided programs are Graphical User Interface (GUI) based, with some of the output going to the GUI and in a few cases, the output goes to a Comma-Separated Values (CSV) file which can be opened with any spreadsheet application such as Microsoft Excel or even a simple text editor.

## 1.4 Conventions

This book uses a period as the decimal separator and a comma as the thousands' separator, e.g., \$24,000.98. This is the convention used in the United States and many other countries. However, there are several other conventions used in other parts of the world. For further information concerning conventions for number representation (and not just for money), see the Wikipedia article entitled "Decimal separator" [1].

Fractions with repeating digits are represented using an overhead bar. For example, we write  $\frac{1}{3}$  as  $. \overline{3}$  with the meaning that there are an infinite number of threes after the decimal point. This also applies for multiple repeated digits, e.g., 12.345781781781 ... would be written 12.345 $\overline{781}$  (the bar just goes over the infinitely repeating digits).

## 1.5 Outline

Section 1 is this introduction.

Section 2 provides some background on various types of progressions and their sums. Basic mathematical concepts such as progressions are a recurring topic in the book.

Section 3 covers topics concerning interest rates, and the past and future value of money. Conceptually, Section 3.7 (concerning the time value of money) is critical to the understanding of much of what follows in the book.

Annuities are discussed in Section 4.

Section 5 covers topics concerning the repayment of loans such as mortgages.

Various types of bonds are discussed in Section 6.

Section 7 covers decision making techniques for capital budgeting.

The code for the Python programs used in this book are listed in Section 8 and also available online at [https://github.com/sfratini33/art-of-managing-things-external/tree/master/fin\\_math/code](https://github.com/sfratini33/art-of-managing-things-external/tree/master/fin_math/code).

## 2 Background

### 2.1 Progressions

Progressions are sequences where each term is based on mathematical operations applied to one or more previous terms. Progressions are essential in finance concerning topics such as interest accumulation.

#### 2.1.1 Arithmetic Progressions

An arithmetic progression is an ordered list (i.e., sequence) of terms where each term is the sum of the previous terms plus some common amount (which can be positive or negative). For example,

- 1, 4, 7, 10, 13, ... is an arithmetic progression with difference 3 between successive terms
- 10, 5, 0, -5, -10, ... is an arithmetic progression with difference -5
- -0.125, 0.125, 0.375, 0.625, 0.875, ... is an arithmetic progression with difference 0.25.

In general, an arithmetic progression with initial term  $x_1$  and difference  $d$  can be represented as

$$x_1, x_1 + d, x_1 + 2d, \dots$$

where the  $n^{th}$  term is  $x_n = x_1 + (n - 1)d$ .

The sum of the first  $n$  terms (call it  $S_n$ ) can be computed by writing the progression forward and backward, and then adding, i.e.,

$$S_n = x_1 + (x_1 + d) + (x_1 + 2d) + \dots + (x_n - 2d) + (x_n - d) + x_n$$

$$S_n = x_n + (x_n - d) + (x_n - 2d) + \dots + (x_1 + 2d) + (x_1 + d) + x_1$$

which implies (after some cancellation of terms)

$$2S_n = n(x_1 + x_n)$$

or

$$S_n = \frac{n}{2}(x_1 + x_n)$$

#### 2.1.2 Geometric Progressions

A geometric progression is an ordered list of terms where each term is a multiple of the preceding term. For example,

- 3, 15, 75, 375, 1875 ... is a geometric progression with a common multiple of 5
- -10, 1.25, -0.15625, 0.01953125, ... is a geometric progression with a common multiple -0.125.

In general, a geometric progression with initial term  $a$  and multiplier  $r$  can be represented as

$$a, ar, ar^2, ar^3, \dots$$

where the  $n^{th}$  term is  $x_n = ar^{n-1}$  where  $n = 1, 2, 3, \dots$

The sum of the first  $n$  terms is

$$S_n = a + ar + ar^2 + \dots + ar^{n-1} \quad (1)$$

Multiplying by  $r$  on both sides of the equation gives

$$rS_n = ar + ar^2 + ar^3 + \dots + ar^n \quad (2)$$

Subtracting equation (2) from (1) yields

$$S_n - rS_n = a - ar^n$$

$$S_n = \frac{a(1 - r^n)}{1 - r}$$

In subsequent parts of this book, we use a slightly modified version of the above formula (gotten by multiplying the top and bottom of the fraction by  $-1$ )

$$S_n = \frac{a(r^n - 1)}{r - 1}$$

As an example, consider the sum of the first 5 terms in the geometric sequence with multiplier 2 and with initial term  $a = 100$ . Using former version of the summation formula for geometric series, we get the following

$$S_5 = \frac{100(1 - 2^5)}{1 - 2} = -100(-31) = 3,100$$

For  $-1 < r < 1$ , the term  $r^n$  approaches zero for  $n$  approaching infinity. Apply this observation to the formula for  $S_n$  gives the following result:

$$a + ar + ar^2 + ar^3 + \dots = \frac{a}{1 - r}, -1 < r < 1$$

For example, the sum of the series with first term 2 and multiplier  $\frac{1}{3}$  is

$$2 + \frac{2}{3} + \frac{2}{3^2} + \frac{2}{3^3} + \dots = \frac{2}{1 - \frac{1}{3}} = 3$$

### 2.1.3 Recursive Progressions

A recursive progression is a progression in which each term is defined using one or more of the previous terms where the first several terms are given. For example,  $a_n = a_{n-1} + a_{n-2}$  with  $a_1 = a_2 = 1$  defines the famous Fibonacci sequence. When expanded, the recursion gives the following progression:

|   |   |   |   |   |   |    |    |    |    |    |     |     |
|---|---|---|---|---|---|----|----|----|----|----|-----|-----|
| 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 | 89 | 144 | ... |
|---|---|---|---|---|---|----|----|----|----|----|-----|-----|

For another example, consider the tetranacci numbers which start with four predetermined terms, with each term afterwards being the sum of the preceding four terms, i.e.,  $a_n = a_{n-1} + a_{n-2} + a_{n-3} + a_{n-4}$  with  $a_1 = a_2 = a_3 = 0$  and  $a_4 = 1$ . When expanded, the recursive formula yields the following:

|        |        |         |         |         |           |       |       |        |        |
|--------|--------|---------|---------|---------|-----------|-------|-------|--------|--------|
| 0      | 0      | 0       | 1       | 1       | 2         | 4     | 8     | 15     | 29     |
| 56     | 108    | 208     | 401     | 773     | 1,490     | 2,872 | 5,536 | 10,671 | 20,569 |
| 39,648 | 76,424 | 147,312 | 283,953 | 547,337 | 1,055,026 | ...   |       |        |        |

The Fibonacci and tetranacci progressions are examples of difference equations.

**[Author's remark:** Recursive progressions are not discussed further in this book. They are mentioned here for completeness with regard to the types of progressions.]

## 2.2 Supporting Programs

### 2.2.1 Arithmetic Progression Calculator

File name: arith\_prog.py (see the preface for location of this file and instructions about running Python programs)

This program generates an arithmetic progression, given a starting value (it can be a decimal, and it can be positive or negative), an increment (it can be a decimal, and it can be positive or negative) and a desired number of terms. If the number of terms is large (fills the output window), then a scroll bar will appear on the right.

There is another button that allows one to request a given term in an arithmetic progression without computing the entire progression up to that term. For example, the program can compute the 101<sup>st</sup> term of the arithmetic progression with initial term 71 and increment 19035, which is 1,903,571.

There is a button to clear the input so that another progression can be computed. There is also a button to quit the program. (For the record, the clear and quit buttons are present in the other programs provided in support of this book.)

For those looking to learn the tkinter module of Python, this program may be helpful as an example. The tkinter module focuses on the user interface.

There are several tkinter tutorials on the Internet, e.g., <https://www.geeksforgeeks.org/python-tkinter-tutorial/> and the YouTube video entitled "Tkinter Course - Create Graphic User Interfaces in Python Tutorial" at <https://youtu.be/YXPyB4XeYLA>.

### 2.2.2 Geometric Progression Calculator

File name: geo\_prog.py

This program generates a geometric progression, given a starting value (it can be a decimal, and it can be positive or negative), an increment (it can be a decimal, and it can be positive or negative) and a desired number of terms. The sum of the progression is also computed and displayed.

The output is listed in the GUI, with one number per line.

There is another button that only generates the  $n^{th}$  term in a given progression.

## 2.3 Exercises

1. Show that the sum of an arithmetic progression can also be represented as  $S_n = \frac{n}{2}(2x_1 + (n - 1)d)$ .
2. Find the 335<sup>th</sup> term in the arithmetic progression 3, 9, 15, 21, ... and also find the sum of the first 335 terms. **Hint:** Use the Python program on arithmetic progressions.
3. If the 57<sup>th</sup> term in an arithmetic progression is 677 and the different between terms is 12, what is the first term. **Hints:** One approach is to use the equation for the  $n^{\text{th}}$  term in an arithmetic progression and solve for  $x_1$ . Another approach is to use the Python program with 677 as the initial term, -12 as the difference and 57 as the desired term (basically working backward with negative steps).
4. Find the 237<sup>th</sup> term in the geometric progression with initial term 2 and multiplier 1.1. **Hint:** Use the Python program on geometric progressions.
5. Find the sum of the first 10 terms in the geometric progression with initial term 25 and multiplier 7. **Hint:** Use the Python program on geometric progressions.
6. If the 8<sup>th</sup> term in a geometric progression is 2,470,629 and the multiplier is 7, what is the first term? **Hint:** Work backwards by dividing by 7 (eight times).
7. The Lucas progression is given by 2, 1, 3, 4, 7, 11, 18, ... Determine a recursive formula that yields the Lucas progression. **Hint:** The initial two terms (2 and 1) are given. The recursive formula should start with the third term.



### 3 Interest

#### 3.1 Simple Interest and Comparison to a Ballon Loan

Simple interest is computed based on the amount of principal and time duration, but without any compounding. Simple interest can be applied to a loan or to money saved in a bank account.

In its most basic form, simple interest entails an interest rate  $r$  (which is typically stated per year), a given amount of principal ( $P$ ) and the term ( $t$ ) when the principal and interest is paid back to the lender. The term is typically stated in years or fractions thereof. Under these assumptions, the amount of simple interest plus the original principal is given by the formula:

$$P + Prt$$

which can be written as

$$P(1 + rt)$$

For example, Mr. Smith purchases a car from a car dealer. The car costs \$28,440. Mr. Smith agrees to make an initial (upfront) payment of \$20,000 and the car dealer agrees to allow Mr. Smith to pay the remaining amount, with 6% annual interest, in 7 years. At the end of 7 years, Mr. Smith must pay the principal (\$8,400) plus simple interest over the 7 years, i.e.,  $\$8,400 * .06 * 7 = \$3,528$ .

In comparison to the above example, let's say that Mr. Smith pays the principal back to the car dealer at the end of 7 years but he must make monthly interest payments in the amount of  $\$8,400 * \left(\frac{.06}{12}\right) = \$42$ . The accumulated amount of interest paid to the car dealer is an arithmetic progression, i.e., 42, 84, 126, ..., 3,528. This is an example of what is called a **balloon loan** that allows for the entire principal to be paid back at the end of the loan's term. To be clear, this is not simple interest. Balloon loans are discussed further in Section 5.2.

#### 3.2 Compound Interest

Compound interest is computed on the sum of the original principal plus accrued interest. Compound interest can be thought of as "interest on interest," and grows faster than simple interest of the same rate.

The term nominal annual interest rate or just **nominal interest rate** as used here is the interest rate "as stated," without adjustment for the effect of compounding. An interest rate is called nominal if the frequency of compounding (e.g., monthly or daily) is not identical to the basic time unit in which the nominal rate is quoted (normally a year). For example, a bank may advertise a nominal yearly interest rate of 6% compounded daily which (as will be shown at the end of Section 3.4) is effectively about 6.183% when daily compounding is applied.

Let  $P$  be the initial principal (e.g., money placed in a bank account),  $r$  be the nominal annual interest rate,  $n$  be the compounding frequency per year, and  $t$  be the length of time the interest is applied (using the same units as  $r$ ). Table 1 shows the computation of principal after the 1<sup>st</sup>, 2<sup>nd</sup> and final compounding intervals. The entries in the right-hand column constitute a geometric progression, with initial term  $P$  and multiplier  $\left(1 + \frac{r}{n}\right)$ .

Some financial books and articles use the notation  $i^{(n)}$  to indicate a nominal annual interest rate that is compounded  $n$  times per year. Thus, our  $r$  is  $i^{(n)}$ . [Author's remark: This notation is confusing. First, it is not a power of  $i$ . Secondly, if the nominal annual interest rate is  $i$ , we would

say that  $i^{(n)} = i$  for all values of  $n$  (also confusing). So, I have avoided using this notation as much as possible and have taken the approach of writing out in prose what is meant, i.e., “nominal annual interest of 6% compounded monthly” rather than  $i^{(12)} = .06$ .]

*Table 1. Computation of Final Principal using Compound Interest*

|   |   |
|---|---|
| Initial principal                           | $P$   |
| Principal after first compounding interval  | $P + \frac{r}{n}P = P\left(1 + \frac{r}{n}\right)$  |
| Principal after second compounding interval | $P\left(1 + \frac{r}{n}\right) + P\left(1 + \frac{r}{n}\right)\left(\frac{r}{n}\right)$ $= P\left(1 + \frac{r}{n}\right)\left(1 + \frac{r}{n}\right)$ $= P\left(1 + \frac{r}{n}\right)^2$ |
| ...   | ...   |
| Principal after $nt$ compounding intervals  | $P\left(1 + \frac{r}{n}\right)^{nt}$  |

Alternately, we can let  $i = \frac{r}{n}$  be the interest rate per compounding period and  $s = nt$  be the number of compounding intervals. These substitutions yield an equivalent formula for the accumulated principal after  $s$  compounding intervals, i.e.,  $P(1 + i)^s$ .

As an example, consider a scenario where Ms. Ortega deposits \$10,000 into a bank account that advertises a 2% nominal annual interest rate which is computed monthly. Assuming no withdrawals have been made, how much is the account worth after 3 years and 6 months? Using the formula for compound interest with  $P = \$10,000$ ,  $r = .02$ ,  $n = 12$  and  $t = 3.5$ , the final principal is  $\$10,000\left(1 + \frac{.02}{12}\right)^{42} = \$10,724.46$ .

### 3.3 Continuous Interest

At first thought, one might think there is no bound to the amount of interest that can be accrued if the compounding is done infinitely often. However, that is not the case. As the number of compounding intervals approach infinity, the formula converges.

**[Author’s remark:** The following derivation entails calculus. If you are not familiar with calculus, don’t worry. The key point here is that if the number of compounding periods is infinite, the formula for accumulated interest is simply  $Pe^{rt}$  where  $e \cong 2.71828$ .]

In terms of calculus, we want to determine

$$\lim_{n \rightarrow \infty} P\left(1 + \frac{r}{n}\right)^{nt}$$

where  $r$  is the nominal annual interest rate (compounded  $n$  times per year where  $n$  approaches infinity) and  $t$  is duration of interest accumulation in years.

To this end, let  $i = r/n$  and substitute into the above expression to get

$$\lim_{n \rightarrow \infty} P\left(1 + \frac{r}{n}\right)^{nt} = \lim_{i \rightarrow 0} P\left(1 + i\right)^{irt} = P \lim_{i \rightarrow 0} \left[\left(1 + i\right)^i\right]^{rt}$$

From calculus, we have that  $\lim_{i \rightarrow \infty} \left(1 + \frac{1}{i}\right)^i = e$  where  $e \cong 2.71828$  (known as Euler's number). Thus, the formula for continuous interest is given by  $Pe^{rt}$ .

Returning to the example of Ms. Ortega's bank account, assume the same interest rate and time period (3.5 years) but with continuous interest. In this case, the account would be worth  $\$10,000e^{(.02)(3.5)} = \$10,725.08$  which is only slightly more than monthly compounding!

### 3.4 Effective Rate of Interest

As defined by Brown and Kopp in their book *Financial Mathematics: Theory and Practice* [1]

The **effective rate of interest** is the amount of money that one unit of capital, invested at the beginning of a period, will earn during the period, where interest is paid at the end of the period.

[Other sources define effective interest rate as pertaining to a period of one year. For example, Wikipedia provides the following definition:

The effective interest rate (EIR), effective annual interest rate, annual equivalent rate (AER) or simply effective rate is the interest rate on a loan or financial product restated from the nominal interest rate and expressed as the equivalent interest rate if compound interest were payable annually in arrears.

This is perhaps a more common definition but less general than the definition from Brown and Kopp.]

For example, if Ms. DeMarco buys a house (as an investment) at \$200,000 and sells the house 18 months later (assuming no capital investment in the house) at \$275,000, then the gain is \$75,000 over 18 months. What is the percentage gain? In other words, what percentage (call it  $x$ ) of 200,000 equals 75,000? This can be converted into a precise equation, i.e.,

$$\frac{x}{100} \cdot 200,000 = 75,000$$

which yields,  $x = .375$  or 37.5% over 1.5 years, which equates to  $\frac{37.5}{1.5} = 25\%$  per year (assuming simple interest).

Take the same problem as above but assume the house was held 30 months before being sold. In this case, the effective interest rate per year is  $\frac{37.5}{2.5} = 15\%$  (assuming simple interest).

Again, using the same base problem, but now assuming the owner incurs \$25,000 in capital improvements and nets \$20,000 from renting the house during the 2.5 years the house is owned by Ms. DeMarco. The net capital gain is now  $75,000 + 20,000 - 25,000 = \$70,000$ . So, the effective interest rate per year is

$$\frac{\left(\frac{(70,000)(100)}{200,000}\right)}{2.5} = \frac{35}{2.5} = 14\%$$

In general, if the value of an investment is  $V(t)$  at time  $t$  and  $V(\hat{t})$  at some later time  $\hat{t}$ , then the effective interest rate from time  $t$  to  $\hat{t}$  is given by the formula

$$\frac{V(\hat{t}) - V(t)}{V(t)}$$

In words, the effective interest rate formula indicates the increase or decrease (as a fraction) of an investment from time  $t$  to a later time  $\hat{t}$ .

The effective interest rate provides a way to compare the growth of simple and compound interest. Assume the starting principal is  $P$  and the interest rate per time period is  $i$ .

In case of compound interest, the effective interest rate in period  $n$  is

$$\frac{P(1+i)^n - P(1+i)^{n-1}}{P(1+i)^{n-1}} = \frac{1+i-1}{1} = i$$

In the case of simple interest, the effective interest rate in period  $n$  is

$$\frac{P(1+ni) - P(1+(n-1)i)}{P(1+(n-1)i)} = \frac{i}{1+(n-1)i}$$

The effective interest rate for compound interest remains constant over time. However, the effective interest rate for simple interest decrease over time (i.e., as  $n$  gets larger) because there is no compounding.

It is common for banks to state the effective yearly interest rate when principal is compounded at a nominal yearly rate of  $r$ . This is referred to as **Annual Percentage Yield (APY)**. The APY (represented as the variable  $y$  below) is derived by inserting the compound interest formula (with  $t = 1$ ) into the effective interest rate formula to get:

$$y = \frac{P(1 + \frac{r}{n})^n - P}{P} = (1 + \frac{r}{n})^n - 1$$

For example, if a bank advertises a nominal yearly interest rate of 6% compounded daily, then the APY is  $(1 + \frac{.06}{365})^{365} - 1 \cong .06183$  or about 6.183%.

It is also possible to solve for the nominal yearly compound interest rate that is equivalent to a given APY, i.e.,

$$r = n((y + 1)^{1/n} - 1)$$

For example, what nominal interest rate (compounded monthly) is equivalent to an APY of 8%?

Using the above equation with  $n = 12$  and  $y = .08$ , we get  $r = 12 \left( (1.08)^{\frac{1}{12}} - 1 \right) = .077206$  or 7.7206%.

A common use of effective interest rate is to normalize other types of interest accumulation with respect to simple interest. It is also possible to relate (via formulas) different types of interest (this is covered in Section 4.7).

### 3.5 Present Value

**Present value** refers to the sum of money which if invested now at a given rate of interest will accumulate to a specified amount at a particular future date.

For example, what is the present value of \$25,000 three years from now assuming 5% interest compounded monthly? In other words, how much money needs to be invested now at 5% interest compounded monthly (with no additional investments or withdraws) to have \$25,000 in three years? In terms of the compound interest formula, the following needs to be solved for  $P$

$$\$25,000 = P(1 + \frac{.05}{12})^{(12)(3)}$$

Solving this equation, we get  $P = \$21,524.41$ . Someone might do such a calculation if he or she wants to buy a \$25,000 car three years from now and wants to know how much to put into a given fixed interest investment (such as a government bond) right now.

Let  $PV$  be the present value and  $FV$  be the future value of  $PV$  over  $s$  compounding periods with interest rate  $i$  per compounding period. In terms of the compound interest formula, we have

$$FV = PV(1 + i)^s$$

Solving for  $FV$  yields the following formula for  $PV$  in the case of compound interest

$$PV = FV(1 + i)^{-s}$$

[The other formulation for compound interest, i.e.,  $P(1 + \frac{r}{n})^{nt}$ , could also have been used in the above calculation.]

In the cases of simple interest and continuous interest, similar formulas can be derived, i.e.,

- $PV = FV(1 + rt)^{-1}$  (present value for simple interest)
- $PV = FV(e^{-rt})$  (present value for continuous interest).

In both of the above formulas,  $r$  is the nominal annual interest rate.

### 3.6 Discounting

**Discount** is a concept similar to interest, i.e.,

- Discount is paid/due at the beginning of a payment period and is calculated on the balance at the end of the period.
- Interest is paid/due at the end of a payment period and calculated on the balance at the beginning of the period.

There are some cases where the anticipated amount of money owed at the end of a loan is deducted at the start of the loan, i.e., the “interest” is paid up front. This is known as “discounting a loan” and the rate of discount is represented by the symbol  $d$ . (Note that the term “discount rate” also refers to the interest rate that the United States Federal Reserve charges banks for short term loans. However, that is not the topic of this section.)

For example, a bank makes a short-term (9 months) \$10,000 loan to Ms. Peeples at a 7% annual discount rate. The bank deducts  $\frac{9}{12}(7) = 5.25\%$  (\$525) at the beginning of the loan. Ms. Peeples

receives  $\$10,000 - \$525 = \$9,475$  up front, and needs to repay  $\$10,000$  at the end of 9 months.

The effective interest rate per 9 months (as seen by the bank) is  $\frac{525}{9,475} \cong .05541$  or 5.541%.

The effective rate of discount (denoted as  $d$ ) from time  $t$  to some later time  $\hat{t}$  is the ratio of the change in value of an investment (known as the amount of discount) to the value at the end of period (at time  $\hat{t}$ ), i.e.,

$$d = \frac{V(\hat{t}) - V(t)}{V(\hat{t})}$$

where  $V(s)$  is the value of an investment at time  $s$ . This is almost the same as the effective interest rate formula except that the denominator is the value at the end of the time period rather than at the beginning. The situation is depicted in Figure 1.

In the previous example,  $d = \frac{10,000 - 9,475}{10,000} = \frac{525}{10,000} = .0525$  or 5.25%.

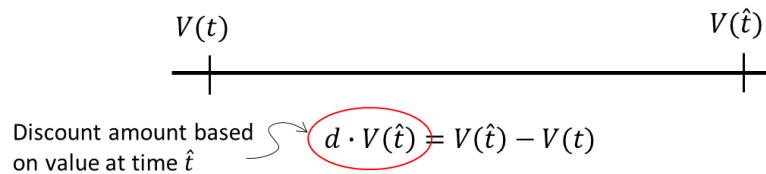


Figure 1. Effective Rate of Discount for a Given Time Period

In the case of compound interest, the effective rate of discount is constant for the repayment periods as can be seen from the calculation below. (Note that  $i$  is the interest rate per compounding period.)

$$d = \frac{V(t_n) - V(t_{n-1})}{V(t_n)} = \frac{P(1+i)^n - P(1+i)^{n-1}}{P(1+i)^n} = \frac{1+i-1}{1+i} = \frac{i}{1+i}$$

Solving for  $i$  in the above equation, we get  $i = \frac{d}{1-d}$ .

In the case of simple interest, the effective rate of discount depends on both the duration of the loan ( $t$ ) and the annual interest rate ( $r$ ). Since there is no compounding, there is only one repayment period which starts at time 0 and ends at time  $t$ . The formula for the effective rate of discount in this case is derived as follows:

$$d = \frac{V(t) - V(0)}{V(t)} = \frac{P(1+rt) - P}{P(1+rt)} = \frac{rt}{1+rt}$$

What is the effective rate of discount for a simple interest discount loan of  $\$10,000$  for 18 months at 10% interest? We are given  $r = .1$  and  $t = 1.5$ . Thus,  $d = \frac{.15}{1+.15} = .130435$  or 13.0435%. Note that 13.0435% of the final value (expected payment) of the loan (i.e.,  $\$11,500$ ) is the stated amount of the loan, i.e.,  $\$10,000$ . In summary, an 18-month loan of  $\$11,500$  at a discount rate of 13.0435% would result in the borrower getting  $\$10,000$  at the beginning of the loan with an obligation to pay the bank  $\$11,500$  in 18 months.

While uncommon, it is possible to accumulate a sum of money with a discount rate. Consider an initial investment of  $x_0$  that is compounded at a discount rate of  $d$  for  $n$  periods:

- At the beginning (time 0),  $x_0$  is calculated as the value of the investment at time 1 (call it  $x_1$ ) minus the discount, i.e.,  $dx_1$  (note that juxtaposition is meant to imply multiplication). So,  $x_0 = x_1 - dx_1$  which implies  $x_1 = x_0(1 - d)^{-1}$ .
- Similarly,  $x_1 = x_2 - dx_2$  which implies  $x_2 = x_1(1 - d)^{-1} = x_0(1 - d)^{-2}$ .
- Continuing in this manner,  $x_n = x_0(1 - d)^{-n}$ .

Growth of the initial amount ( $x_0$ ) is depicted in Figure 2.

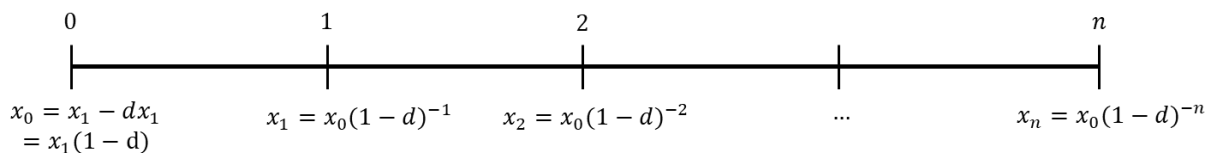


Figure 2. Accumulating with a Discount Rate

For example, what amount must be invested, at a discount rate of .45% per month, to have \$50,000 at the end of 10 years under a monthly compound discount approach? We are given  $d = .0045$  and  $n = 10 \times 12 = 120$  and are looking for  $x_0$  such that  $x_{120} = 50,000$ . Using the formula for compound discounting, we have

$$x_{120} = \$50,000 = x_0(1 - .0045)^{-120}$$

$$x_0 = \$50,000(1 - .0045)^{120} \cong \$29,101.93.$$

As another example, consider a discounted 5-year loan at a discount rate of .4% per month where the loanee initially receives \$100,000 from a bank. What is the actual size of the loan, i.e., the amount of money to be paid back to the loaner at the end of 5-years? We are given  $d = .004$ ,  $n = 5 \times 12 = 60$  and  $x_0 = 100,000$  and are being asked to find  $x_{60}$ . Using the formula for compound discounting, we have

$$x_{60} = (100,000)(1 - .004)^{-60} \cong \$127,186.11$$

In other words, a discounted loan in the amount of \$127,186.11 is requested. The discount amount  $\$27,186.11 = \$127,186.11 - \$100,000$  is deducted from the loan by the bank and the loanee receives \$100,000. At the end of 5 years, the loanee must pay back \$127,186.11 to the bank.

The equivalent interest rate is  $i = \frac{d}{1-d} = \frac{.004}{1-.004} = .004016064$  (monthly). Compounding at  $i$  for 60 months with an initial value of \$100,000 yields a future value (in 60 months) of

$$\$100,000(1 + .004016064)^{60} = \$127,186.11$$

### 3.7 Time Value of Money

Regarding financial decisions, the value of an amount of money (relative to a given time frame) needs to be considered. For example, receiving \$100 from an investment today is not the same as receiving \$100 ten years from now assuming that the \$100 could be making interest over the ten year span.

Figure 3 shows the value of  $X$  at a focal date (could be the present or some other date of importance) and equivalent values in the past and future based on simple, compound, and continuous interest. For example, if the amount  $X(1+i)^{-n}$  is compounded for  $n$  time periods, then we will have  $X(1+i)^{-n}(1+i)^n = X$ . If the amount  $Xe^{-in}$  is continuously compounding for  $n$  time periods, then it would grow to  $(Xe^{-in})e^{in} = X$  at time  $n$ . We can also work in the opposite direction. For example, how much is needed at the focal date to have  $X + in$  at  $n$  time periods in the future. From the diagram, we see the required amount is  $X$ .

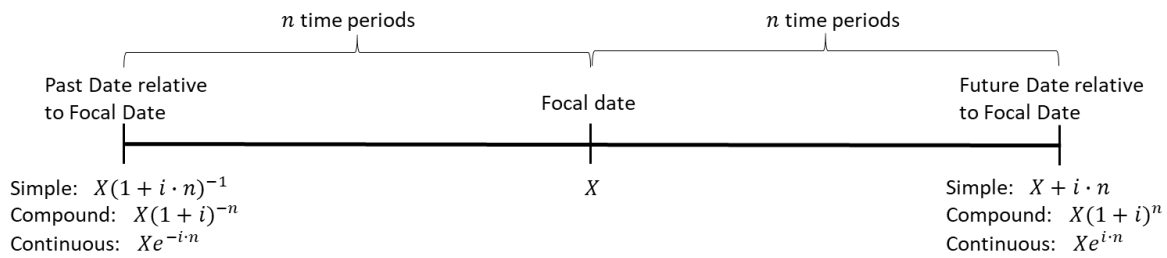


Figure 3. Equivalent Values to  $X$  in the Past and Future

Consider a personal loan of \$5000 between friends that is due at the end of a three year period. Determine an equivalent amount of debt due in 1 year and in 5 years if they agree to annual nominal interest rate of 6% compounded monthly.

Select the focal date to be when the debt is due, i.e., in three years. The equivalent value of the debt two years before the focal date (i.e., one year from the present) is

$5,000(1 + \frac{.06}{12})^{-24} = \$4,435.93$ . In this case, the debt is paid early. The equivalent value of the debt two years after the focal date is  $5,000(1 + \frac{.06}{12})^{24} = \$5,635.80$ . In this case, the debt is later than the agreed repayment time (which we assumed to be the focal date in the calculations).

As a second example, assume Mr. Jasper owes Ms. Dubois \$1,000 in 6 months and another \$1,500 in 2 years from now. What single payment, in the present, will liquidate these debts if both parties agree money is worth 3% (compounded monthly). Same question if the debt is paid in a year.

For the first question, we compute the equivalent value of the two future payments if made at the present time, i.e.,

$$X = 1,000 \left(1 + \frac{.03}{12}\right)^{-6} + 1,500 \left(1 + \frac{.03}{12}\right)^{-24} = \$2,397.88$$

For the second question, notice that the first expected payment is being paid late and will thus be greater than \$1000.

$$X = 1,000 \left(1 + \frac{.03}{12}\right)^6 + 1,500 \left(1 + \frac{.03}{12}\right)^{-12} = \$2,470.82$$

A useful technique in financial mathematics is to replace one set of payments/debts with an equivalent set. Two sets of payments/debts are equivalent if the sum of the dated values of one set is equal to that of the other set on any date (known as a **focal date** or the comparison date or the valuation date). This is under the assumption of a common interest rate scheme.



As an example of this principle, consider a \$100,000 business loan that is to be paid in 3 equal payments at the end of 4, 7 and 11 months, with a final payment of \$20,000 in 13 months. Determine the value of the 3 equal payments. The nominal annual interest rate is 4% (compounded monthly).

To solve the problem, first select a focal date. Any date will do. Let's select 7 months from the start of the loan to be the focal date. At the focal date, the equivalent value of the payments must equal the equivalent value of the debt. If we let  $x$  be amount of each unknown payment, then we have the following equation:

$$\begin{aligned} \text{value of payments relative to focal date} &= \text{value of debt relative to focal date} \\ x \left(1 + \frac{.04}{12}\right)^3 + x + x \left(1 + \frac{.04}{12}\right)^{-4} + 20,000 \left(1 + \frac{.04}{12}\right)^{-6} &= 100,000 \left(1 + \frac{.04}{12}\right)^7 \\ 2.99681x + 19,604.62549 &= 98,023.12747 \\ x &= \$27,613.42 \end{aligned}$$

The relationship between the debt and payments relative to the focal date is depicted in Figure 4.

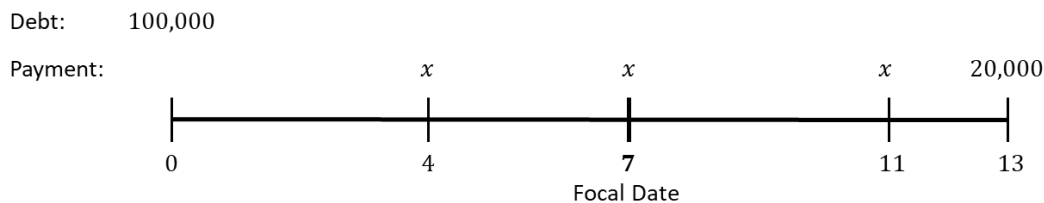


Figure 4. Debt and Payments Relative to a Focal Date

If a different focal date is used, the same answer should be obtained. For example, the following equation uses the end of the loan (13 months) as the focal date.

$$\begin{aligned} x \left(1 + \frac{.04}{12}\right)^9 + x \left(1 + \frac{.04}{12}\right)^6 + x \left(1 + \frac{.04}{12}\right)^2 + 20,000 &= 100,000 \left(1 + \frac{.04}{12}\right)^{13} \\ 3.05725x + 20,000 &= 10,4421.0681 \\ x &= \$27,613.42 \end{aligned}$$

### 3.8 Interest, Inflation and Taxes

At the beginning of a given period of time, assume that you have \$1 and the inflation rate is 3% for that period. An item that costs \$1 at the beginning of the period will cost \$1.03 at the end of the period. Your \$1 at the end of the period can only purchase  $\frac{1}{1.03} \cong .97$  of what it could have at the beginning of the period. So, the effective simple interest rate is -3%. Another way to view this is ask "what is the future value of \$1 under -3% simple interest for one year?" The answer is

$$FV = P(1 + rt) = 1(1 - .03) = .97$$

noting that  $P = 1$ ,  $r = -.03$  and  $t = 1$  in the above equation.

However, if you invested your \$1 at 5% interest at the beginning of the period, you would have \$1.05 which could be used to purchase  $\frac{1.05}{1.03} \cong 1.0194$  of what you could have at the beginning of the period with \$1. So, the real rate of return (for the noted investment) is 1.94% in this scenario.

Look at it this way:

Let's say that at the beginning of the year, a candy bar costs \$1. Also, at the beginning of the year, you invest \$1 at 5% simple interest. Over the course of the year, inflation is 3%. At the end of the year, you have \$1.05 and you want to know how many of the candy bars you can now buy (call this quantity  $x$ ). Due to inflation, the candy bar now cost \$1.03. So, we have the task of solving the equation  $1.05 = 1.03 x$ . This gives the result  $x = \frac{1.05}{1.03} = 1.0194$  candy bars.

In general, if the interest rate for a given period is  $i$  and the inflation rate is  $r_{inf}$ , the **real rate of return** (i.e., interest adjusted for inflation) is

$$\frac{1 + i}{1 + r_{inf}} - 1$$

The above is known as the Fisher equation [5].

If there is a tax rate of  $i_t$  at the end of the year or given period, the effective interest becomes  $i_t \cdot i$ . This gives us the modified real rate of return (with taxes considered):

$$\frac{1 + (1 - i_t) \cdot i}{1 + r_{inf}} - 1$$

Returning to the previous problem with .05 simple interest and a .03 rate of inflation, and assuming a tax rate of 20%, we get the following real rate of return

$$\frac{1 + (1 - .2)(.05)}{1 + .03} - 1 = \frac{1.04}{1.03} - 1 \cong 1.0097 \text{ or } .97\%$$

## 3.9 Supporting Programs

### 3.9.1 Interest Compounding Calculator

File name: interest.py

This calculator determines compound interest and continuous interest. The expected input is

- the original principal
- number of compounding periods per year (this is left blank if continuous interest desired)
- an indication of whether continuous interest is desired (enter True or False)
- time that interest accrues in years or fraction of years (e.g., 1.75 years is allowed).

The output is the final principal (which includes the original principal plus accrued interest).

**[Author's Remark:** While I did not plan it, the interest compounding calculator also works for a negative number of years. For example, let's say you want to know how much to invest now at 5% nominal annual interest (compounded monthly) so that in 10 years you have \$1,000. In the calculator, enter 1000 as the principal, 5 for the interest rate, 12 for the number of compounding periods per year and -10 for the number of years. This will give an answer of 607.16.]

### 3.10 Exercises

1. Determine the value of a simple interest investment of \$10,000 at 3% annual interest rate over a period of 2 years and 7 months. What would the value of the investment be if daily compounding were used instead? **Hint:** For the second part of the problem, use the interest compounding calculator.
2. Country A makes a 100 year loan to Country B from 1 billion dollars at an annual interest rate of 6%. Compute the amount of interest paid back to Country A under the assumption of yearly compounded interest, daily compounded interest and continuous compounding. **Hint:** Use the interest compounding calculator for each scenario.
3. Mr. Wimple buys a diamond for investment purposes at \$55,000, holds the diamond for 5 years and 6 months and then sells the diamond for \$100,000. What is the effective interest rate per year (assuming simple interest)? What is the effective interest rate if monthly compounding is assumed? **Hint:** For the second question, use the compound interest formula with  $P = 55,000$ ,  $n = 12$  and  $t = 5.5$ , set the result equal to 100,000 solve for  $r$ , i.e., solve  $55,000(1 + \frac{r}{12})^{12t} = 100,000$  for  $r$  by taking the natural log on both sides and then apply some basic algebra. The answer is 10.9191% which, as expected, is smaller than the solution to the first question, i.e., 14.876%.
4. Given a nominal yearly interest rate of  $r$ , compute the effective annual interest rate under the assumption of continuous compounding between times  $t$  and  $\hat{t}$ . **Hint:** From the formula for effective interest rate, we have

$$\frac{V(\hat{t}) - V(t)}{V(t)} = \frac{Pe^{r(\hat{t}-t)} - P}{P} = e^{r(\hat{t}-t)} - 1$$

which is the effect interest rate for the entire time period  $\hat{t} - t$ . To get the effective annual interest rate, just divide by  $\hat{t} - t$ .

5. What nominal annual interest rate  $r$  is required to quadruple an initial amount in 20 years, with monthly compounding? Same question with continuous compounding? **Hint and Answer:** For the first question, solve the equation  $4P = P(1 + \frac{r}{12})^{240}$  to get  $r = 6.9515\%$ . For the second question, solve the equation  $4P = Pe^{20r}$  to get  $r = 6.9315\%$ . In both cases, you need to take the natural log (i.e., log base  $e$ ) on both sides and then do some algebra.
6. A rich uncle leaves \$5 million to 3 heirs (all nieces). Each niece is to receive an equal amount of money when she turns 21 years old, which will be 12 months, 27 months, and 43 months from the time the will is executed. In the meantime, the money is invested at a nominal annual interest rate of 6% (compounded monthly). How much money is each niece to receive? **Hint:** Use the time that the will is executed as the focal date and equate present value of the investment with that of the payouts to get the following equation (where  $x$  is the amount to be paid to each heir):  $x(1 + .005)^{-12} + x(1 + .005)^{-27} + x(1 + .005)^{-43} = 5,000,000$ . Solve for  $x$ .
7. Given a debt of  $Q$  at time 0 which is resolved by  $n$  equal payments of amount  $x$  with periodic interest rate  $i$ , determine  $x$ . **Hint:** Using the concept of equivalent payments/debts with the focal date being time 0, we get the equation  $Q = x(1 + i)^{-1} + x(1 + i)^{-2} + \dots + x(1 + i)^{-n}$ . Use the formula for the sum of a geometric series and then solve for  $x$ . We will return to this problem in Section 4.3.1.

## 4 Annuities

### 4.1 Terminology

An **annuity** is a series of payments, usually (but not necessarily) made at equal intervals of time and in equal amounts. The time between successive payments of an annuity is called the payment interval (or period). The time from the beginning of the first payment interval to the end of the last payment interval is referred to as the **term** of an annuity.

When the dates of the first and the last payments of an annuity are fixed, the annuity is called an **annuity certain**. When the term of the annuity depends on some uncertain event (e.g., death of the annuity's holder), the annuity is called a **contingent annuity**.

When payment is made at the end of each payment interval, the annuity is called an **ordinary annuity** (or immediate annuity). When payment is made at the beginning of each payment interval, the annuity is called an **annuity-due**. A **deferred annuity** is an ordinary annuity whose first payment is due at a later date.

When the beginning and end of the payment interval, and the interest compounding period coincide, the annuity is called a **simple annuity** (e.g., the mortgages defined in Section 5.1 are simple annuities); otherwise, it is a **general annuity**.

The **accumulated (or future) value of an annuity** is the value of the recurring payments to an annuity at a certain date in the future, assuming a particular rate of return. The **discounted (or present) value of an annuity** is the current value of future payments of an annuity, given a specified rate of return, or discount rate. The higher the discount rate, the lower the present value of the annuity.

### 4.2 Accumulated Value of Ordinary, Simple, Certain Annuities

#### 4.2.1 Concept and Formulas

Consider an annuity where an amount  $R$  is added (by the entity buying the annuity) at the end of each period, interest is compounded at the end of each period, the interest rate is  $i$  (per period) and there are  $n$  periods. The payment schedule for such an annuity is shown in Figure 5.

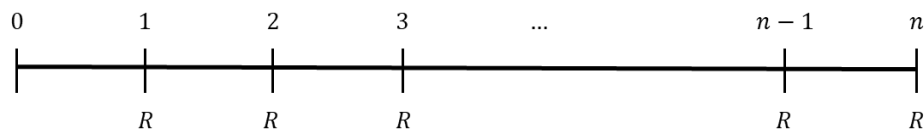


Figure 5. Payment Schedule for Accumulated Value of Ordinary, Simple, Certain Annuities

No interest is earned on the contribution at the end of period  $n$  since that is the end of the annuity's term. The contribution at the end of period  $n - 1$  earns interest  $iR$  and so its accumulated value at the end of the annuity's term is  $R + iR = R(1 + i)$ . The accumulated value for the contributions is summarized in Table 2.

Table 2. Accumulated Value of an Ordinary, Simple, Certain Annuity

| End of Period | Accumulated Value of Given Investment |
|---------------|---------------------------------------|
| $n$           | $R$                                   |
| $n - 1$       | $R + iR = R(1 + i)$                   |
| $n - 2$       | $R(1 + i) + iR(1 + i) = R(1 + i)^2$   |
| ...           | ...                                   |
| 1             | $R(1 + i)^{n-1}$                      |

The value of the annuity at term is the sum (call it  $S$ ) of the following geometric series:

$$\begin{aligned}
 S &= R + R(1 + i) + R(1 + i)^2 + \cdots + R(1 + i)^{n-1} \\
 &= R \frac{(1 + i)^n - 1}{(1 + i) - 1} = R \frac{(1 + i)^n - 1}{i}
 \end{aligned}$$

**[Remark concerning notation:** the quantity  $\frac{(1+i)^n-1}{i}$  is often represented by a shorthand notation in financial books and articles. The notation is usually some variation of  $s_{n|i}$ .]

#### 4.2.2 Examples

For example, Mr. Munch agrees to an annuity savings plan from an investment company. Under the plan, Mr. Munch will contribute \$300 at the end of each month for a period of three years. Interest is compounded at the end of each month at a nominal annual rate of 6%. How much is the annuity worth at term?

We have at  $R = 300$ ,  $n = 3(12) = 36$  and  $i = \frac{.06}{12} = .005$ . Using the accumulated value formula for an annuity gives the following result:

$$S = 300 \frac{1.005^{36} - 1}{.005} = \$11,800.83$$

Mr. Munch reconsiders and tells that investment company that he really needs about \$15,000 to buy a car in three years and would like to know how much he needs to save each month to reach that goal under the same conditions as stated above.

In this case, we need so solve for  $R$  given  $S = 15,000$ ,  $n = 36$  and  $i = .005$ . We first solve the accumulation formula for  $R$  to get

$$R = \frac{iS}{(1 + i)^n - 1}$$

We then substitute the specific values for the problem at hand into the above formula and get the following result

$$R = \frac{(.005)(15,000)}{1.005^{36} - 1} \cong \$381.33$$

In yet another variation of the problem, assume Mr. Munch wants to limit his monthly payment to \$300 but is willing to wait longer to reach his goal of \$15,000. In this scenario, we need to solve the equation  $S = R \frac{(1+i)^n-1}{i}$  for  $n$ . With some rearrangement of the equation, we get

$$\frac{iS}{R} + 1 = (1 + i)^n$$

Take the natural log on both sides and the equation, and then isolation  $n$  to get:

$$n = \frac{\ln(\frac{iS}{R} + 1)}{\ln(1 + i)}$$

For the problem at hand, we get  $n = 44.74$  which we need to round-up to the next whole number, i.e., 45 payments of \$300 are required at the given interest rate for the annuity to surpass \$15,000 at term.

### 4.2.3 Blocking

Especially for longer-term annuities, the entity offering the annuity is not likely to provide a fixed interest rate for the entire term of the annuity. The investment entity may vary the rate based on something like the 30-year United States treasury bond interest rate. In such cases, the computation of the annuity's value needs to be blocked into several computations.

#### 4.2.3.1 Example

An example will illustrate the point. Ms. Gold agrees to an annuity savings plan from a bank that stipulates a deposit of \$500 at the end of each month for 10 years. The interest rate is based on the average 30-year treasury bond rate for the past month. For the first 5 years (60 deposits), the nominal annual rate for the annuity was 3% and for the last 5 years, the nominal annual rate was 3.5%. How much was the annuity worth at term, assuming Ms. Gold has contributed the agreed \$500 per month?

We first compute value of the annuity at the end of the initial 5 years (call it  $S_1$ ):

$$S_1 = 500 \frac{(1 + \frac{.03}{12})^{60} - 1}{\frac{.03}{12}} = \$32,323.36$$

Next, compute the value of the contributions to the annuity during the second 5 years (call it  $S_2$ ):

$$S_2 = 500 \frac{(1 + \frac{.035}{12})^{60} - 1}{\frac{.035}{12}} = \$32,733.06$$

We are not done because it is necessary to consider the interest accrued to  $S_1$  during the second 5 years of the annuity, which is  $S_1(1 + \frac{.035}{12})^{60} = \$38,495.27$

So, the total value of the annuity at term is  $\$32,733.06 + \$38,495.27 = \$71,228.33$ , while Ms. Gold contributed a total of  $120 \times \$500 = \$60,000$ .

In a variation of the above problem, assume the term is changed to 15 years with the first 10 years being as before but for the last five years the nominal interest rate is 6% (compounded monthly). Further, Ms. Gold does not make further contributions beyond the first 10 years. In this scenario, we simply need to compound \$71,716.26 for another 5 years, i.e.,

$$\$71,228.33(1 + \frac{.06}{12})^{60} = \$96,076.34$$

#### 4.2.3.2 Generalization

Blocking can be made even more general than what is suggested in the above example. Each block can have a different interest rate per period, a different payment per period and a different number of periods. So, each block is defined by three variables, i.e., interest rate, payment amount and number of periods. If either the interest rate or payment amount changes, we start a new block. So, these two attributes (i.e., interest rate and payment amount) determine the third variable (i.e., number of periods in a block).

The following algorithm can be used to determine the final value of an annuity with  $m$  blocks:

1. For each of the  $m$  blocks, compute the accrued value of the annuity for the interval when the block is active. This is done using the formula below for block  $k$ . The subscript  $k$  is put on  $i, R$  and  $n$  to emphasize these values can change for each block.

$$S_k = R_k \frac{(1 + i_k)^{n_k} - 1}{i_k}$$

2. For each  $S_k$  ( $k = 1, 2, \dots, m - 1$ ), the value continues to accrue during subsequent blocks, based on the interest rate and number of periods for each subsequent block. So, the value at the end of the annuity for block  $k$  is

$$V_k = S_k(1 + i_{k+1})^{n_{k+1}}(1 + i_{k+2})^{n_{k+2}} \dots (1 + i_m)^{n_m}$$

3. Since there is no further accumulation of interest after the last block is finished, we have that  $V_m = S_m$ .
4. The total value of the annuity at term is  $V = V_1 + V_2 + \dots + V_{m-1} + S_m$

### 4.3 Discounted (Present) Value of Ordinary, Simple, Certain Annuities

#### 4.3.1 Concept and Formulas

Consider a scenario in which someone purchases an annuity up front and then gets constant payments in the amount  $R$  at the end of each of  $n$  periods. At the end of period  $n$ , the value of the annuity is 0. The interest rate per period is  $i$ .

In order to cover the first payment (end of period #1), how much needs to be invested at beginning of period 1? If we call this quantity  $x_1$ , then we need to solve  $x_1 + x_1 i = R$  for  $x_1$ . This gives us  $x_1 = R(1 + i)^{-1}$ . So,  $x_1$  invested at the time 0 will be worth  $R$  at the end of the first period. This is basically a present value calculation for a future desired amount.

In general, we need to calculate the present value for each of the  $n$  pays of amount  $R$  and then sum the amounts, i.e.,

$$\begin{aligned} Q &= R(1 + i)^{-1} + R(1 + i)^{-2} + \dots + R(1 + i)^{-n} \\ &= R(1 + i)^{-1} \{1 + (1 + i)^{-1} + [(1 + i)^{-1}]^2 + \dots + [(1 + i)^{-1}]^{n-1}\} \end{aligned}$$

Noting that the above is the sum of a geometric series with  $a = R(1 + i)^{-1}$  and  $r = (1 + i)^{-1}$ , we get the following equation

$$Q = R(1 + i)^{-1} \frac{1 - (1 + i)^{-n}}{1 - (1 + i)^{-1}}$$

Multiplying by  $\frac{1+i}{1+i}$  and some simplification results in the following formula

$$Q = R \frac{1 - (1 + i)^{-n}}{i}$$

So, an initial investment in the amount of  $Q$  with interest rate  $i$  per period (compounded at the end of each period) is equivalent to  $n$  payments in the amount  $R$  at the beginning of each period. (Recall Exercise 7 from Section 3.10.)

Alternately, the above formula can be solved for  $R$  in terms of the other variables to get

$$R = \frac{iQ}{1 - (1 + i)^{-n}}$$

This is useful if one knows the interest rate, desired number of distributions and the initial investment, and wants to determine how much can be distributed at the end of each period.

We can also solve for  $n$  to determine how long an initial investment will last given the distribution amount per period and interest rate, i.e.,

$$n = \frac{\ln R - \ln(R - iQ)}{\ln(1 + i)}$$

where  $\ln(x)$  is the natural log of  $x$  (i.e., log base  $e$ ).

**[Remark concerning notation:** the quantity  $\frac{1 - (1 + i)^{-n}}{i}$  is often represented by a shorthand notation in financial books and articles. The notation is usually some variation of  $a_{n|i}$ .]

#### 4.3.2 Examples

What initial investment is needed to get a monthly payment of \$750 for 20 years given a nominal yearly interest rate of 5% (compounded monthly)?

We are given  $R = \$750$ ,  $n = 12 \times 20 = 240$  and  $i = \frac{.05}{12}$ . Using the formula for  $Q$ , we get

$$Q = 750 \frac{1 - \left(1 + \frac{.05}{12}\right)^{-240}}{\frac{.05}{12}} = \$113,643.98$$

As a second example, consider someone who has \$100,000 to invest in an annuity with a nominal annual interest rate of 4% (compounded monthly) and wants to have the annuity distribution monthly for 15 years. How much can be distributed each month for the given period?

In this scenario, we are given  $Q = \$100,000$ ,  $n = 12 \times 15 = 180$  and  $i = .04/12$ , and we need to determine  $R$ .

$$R = \frac{iQ}{1 - (1 + i)^{-n}} = \frac{(\frac{.04}{12})(100,000)}{1 - (1 + \frac{.04}{12})^{-180}} = \$739.69$$

In yet another example, we want to find the number of quarters (i.e., quarter-years) that an investment in the amount of \$27,976.08 will last, given equal quarterly payments in the amount of \$1000 with a quarterly interest rate of .013.

We are given  $R = \$1,000$ ,  $Q = \$27,976.08$  and  $i = .013$ , and want to find  $n$  (the number of quarters) the investment will last. Using the formula for  $n$  from the previous subsection, we get



$$n = \frac{\ln R - \ln(R - iQ)}{\ln(1 + i)} = \frac{\ln(1,000) - \ln(1,000 - (.013)(27,976.08))}{\ln(1 + .013)} = 35$$

So, the annuity can payout \$1,000 per quarter for 35 quarters or 8 years and 9 months.

(In most cases, the value of  $n$  will not be a whole number. In such cases, the final payment will be less than the previous recurring payments.)

#### 4.4 Annuities Due

In an **annuity-due**, the periodic payments are due at the beginning of each payment interval. The term of an annuity-due starts at the time of the first payment and ends one payment period after the date of the last payment. The payment schedule for an annuity due is shown in Figure 6.

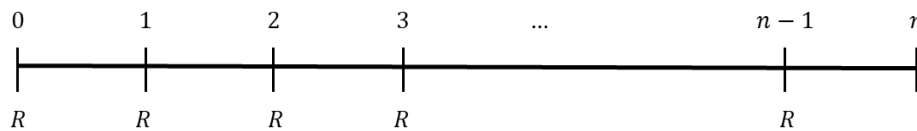


Figure 6. Payment Schedule for an Annuity Due

The accumulated (or future) value of an annuity-due at term is similar to the formula for  $S$  in Section 4.2.1, except that interest is earned on each contribution for one additional period. So, we need to multiply the formula for the accumulated value of an ordinary, simple, certain annuity by  $(1 + i)$  to get the accumulated value for an annuity due:

$$(1 + i)R \frac{(1 + i)^n - 1}{i}$$

which can be written as

$$R \frac{(1 + i)^{n+1} - (1 + i)}{i}$$

The discounted (or present) value of an annuity-due is similar to the formula for  $Q$  in Section 4.3.1, except that all the payments are done at the beginning of each period. So, the discounted value of an annuity-due is

$$\begin{aligned} & R + R(1 + i)^{-1} + R(1 + i)^{-2} + \dots + R(1 + i)^{-(n-1)} \\ &= R[1 + (1 + i)^{-1} + (1 + i)^{-2} + \dots + (1 + i)^{-(n-1)}] \\ &= R \frac{1 - (1 + i)^{-n}}{1 - (1 + i)^{-1}} \end{aligned}$$

Note that the above is just  $(1 + i)$  times the formula for discounted (present) value of an ordinary, simple, certain annuity.

#### 4.5 Deferred Annuities

As noted earlier, a deferred annuity is an ordinary annuity where the first payment is made after a given number of periods  $m$ . Take the example of two parents buying an annuity for their daughter's education. Assume the child is 11 years old when the annuity is purchased and there are no payments back to the annuity owners until the child turns 18. In this case, the annuity is deferred for 7 years (earning interest in the meantime) and the payments are made during the child's time in

college (say 5 years). At the annuity's term (12 years from time of purchase), all the money has been distributed back to the parents who are presumably using the money to pay college bills for their daughter.

In general, assume

- the annuity is purchased at time 0
- there are to be  $n$  payments (to the owner of the annuity), each in the amount of  $R$
- the payments start at the end of period  $m$  (i.e., at time  $m+1$ )
- the interest rate per period is  $i$ .

The scenario is summarized in Figure 7.

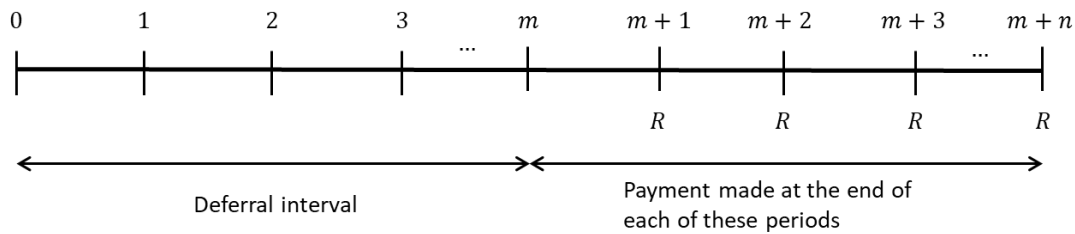


Figure 7. Deferred Annuity

To compute the overall present value (purchase price) of the annuity, we compute the present value of each payment. To cover the first payment (at time  $m+1$ ), how much needs to be invested at time 0? If we call that quantity  $x_1$ , then we need to solve  $x_1(1+i)^{m+1} = R$  for  $x_1$ . This gives us  $x_1 = R(1+i)^{-(m+1)}$ . So,  $x_1$  invested at the time 0 will be worth  $R$  at time  $m+1$ . Using similar reasoning, the present value of the second payment is  $R(1+i)^{-(m+2)}$ , and so on. Thus, the present value for the annuity is the sum

$$\begin{aligned} Q &= R(1+i)^{-(m+1)} + R(1+i)^{-(m+2)} + \dots + R(1+i)^{-(m+n)} \\ &= R(1+i)^{-(m+1)} [1 + (1+i)^{-1} + (1+i)^{-2} + \dots + (1+i)^{-(n-1)}] \\ &= R(1+i)^{-(m+1)} \frac{1 - (1+i)^{-n}}{1 - (1+i)^{-1}} \end{aligned}$$

If the interest rate is assumed to be different in the deferral period (call it  $j$ ) from the rate during the payment periods (call it  $i$ ), then we get a more general equation for  $Q$ . To be clear, it is assumed the interest rate changes from  $i$  to  $j$  **after** the first payment. This is almost the same as the previous equation, with the difference highlighted in bold.

$$Q = R(1+j)^{-(m+1)} \frac{1 - (1+i)^{-n}}{1 - (1+i)^{-1}}$$

Going back to the college expense savings example, we have the following inputs:

- $m+1 = 7 \times 12 = 84$  and  $m = 83$
- $n = 5 \times 12 = 60$
- $i = j = \frac{.05}{12} = .0041\bar{6}$  (assume interest is fixed over the entire term of the annuity)
- assume that \$2,500/month is needed during the child's college years, so  $R = \$2,500$ .

Assuming payments and compounding are done monthly, and using the formula for  $Q$ , we get

$$Q = R(1+i)^{-(m+1)} \frac{1 - (1+i)^{-n}}{1 - (1+i)^{-1}}$$

$$= (2,500)(1.0041\bar{6})^{-84} \frac{1 - (1.0041\bar{6})^{-60}}{1 - (1.0041\bar{6})^{-1}} = \$93,811.97$$

In a variation of the above scenario, assume that during the deferral interval, the interest rate is 6% and 4% just after the payments start. Thus,  $j = \frac{.06}{12} = .005$  and  $i = \frac{.04}{12} = .00\bar{3}$  and using the second formula for  $Q$ , we get that  $Q = \$89,583.59$ .

## 4.6 Forborne Annuities

In a **forborne annuity**, the annuity continues to earn interest after the last payment into the annuity is made. We have already seen a forborne annuity in the variation of the example in Section 4.2.3.1.

There are two steps in calculating the accumulated value of a forborne annuity:

- Calculate the accumulated value at the time of the last payment using the formula from Section 4.2.1, i.e.,  $R \frac{(1+i)^n - 1}{i}$ , where  $i$  is the interest rate per period,  $R$  is the payment amount (in the annuity per period) and  $n$  is the number of payments.
- Calculate the interest earned while the annuity is held after the last payment into the annuity, using the formula for compound interest where the principal is given by  $R \frac{(1+i)^n - 1}{i}$ . Assuming the annuity is held for  $m$  periods after the last payment, the final accumulated value of the annuity is  $R \frac{(1+i)^n - 1}{i} (1+i)^m$ .

## 4.7 General Annuities

### 4.7.1 Concept and Interest Conversion Formulas

In the case of general annuities, the compounding and payment periods do not coincide (unlike the case for simple annuities). This problem can be solved by converting the general annuity into an equivalent ordinary simple annuity where the compounding and payments are on the same schedule.

In general, if compounding is done  $s$  times per year at a rate of  $i$ , and payments are made  $t$  times per year, we want to know the equivalent interest rate to  $i$  (call it  $j$ ) if compounding is done  $t$  times per year. The problem is to solve for  $j$  in the following equation

$$(1+i)^s = (1+j)^t$$

Taking the  $\frac{1}{t}$  power of both sides of the equation and then doing some rearrangement of terms gives us the following formula

$$j = (1+i)^{s/t} - 1$$

If interest is compounded continuously, then we need to solve  $e^{r \cdot 1} = (1+j)^t$  in terms of  $j$ , where  $r$  is the nominal annual interest rate. Taking the  $\frac{1}{t}$  power of both sides of the equation gives

$$j = e^{r/t} - 1$$

Alternately, one can solve for  $r$  to get

$$r = t \ln(1 + j)$$

For example, 6% interest compounded monthly (i.e.,  $t = 12$  and  $j = .005$ ) is equivalent to  $r = 12 \ln(1.005) = 0.05985$  or 5.985% compounded continuously.

#### 4.7.2 Example – Accumulated Value of a General Annuity

Compute the accumulated value of a general annuity with term 11 years, where the interest is compounded 5 times per year at a nominal annual rate of 5%. Payments into the annuity are made 7 times per year in the amount of \$1,000 per payment.

Determine the interest rate per compounding period, i.e.,  $i = \frac{.05}{5} = .01$ .

Determine the equivalent interest rate if compounding is done 7 times per year, i.e.,

$$j = (1 + .01)^{\frac{5}{7}} - 1 = 0.007132697$$

Use the formula for the accumulated value of a simple annuity (from Section 4.2.1) with per period interest rate 0.007132697,  $n = 11 \times 7 = 77$  and  $R = 1000$

$$S = R \frac{(1 + j)^n - 1}{j} = 1000 \frac{1.007132697^{77} - 1}{.007132697} = \$102,138.73$$

This can be verified using the block annuity calculator with an input csv file containing the following line:

$$.007132697, 1000, 77$$

Solve the same problem as above, except that the interest is compounded continuously. In this scenario,  $j = e^{i/t} - 1 = e^{\frac{.05}{7}} - 1 = 0.007168428$  and  $S = \$102,289.23$ .

This can be verified using the block annuity calculator with an input csv file containing the following line:

$$.007168428, 1000, 77$$

#### 4.7.3 Example – Discounted (Present) Value of a General Annuity

Compute the discounted value of a general annuity with term 13 years, where the interest is compounded 12 times per year at a nominal annual rate of 6%. Payments out of the annuity are made 26 times per year in the amount of \$500 per payment.

Determine the interest rate per compounding period, i.e.,  $i = \frac{.06}{12} = .005$ .

Determine the equivalent interest rate if compounding is done 26 times per year, i.e.,

$$j = (1 + .005)^{\frac{12}{26}} - 1 = 0.0023046$$

Use the formula for the discounted value of a simple annuity (from Section 4.3.1) with per period interest rate 0.0023046,  $n = 26 \times 13 = 338$  and  $R = 500$  to get

$$Q = R \frac{1 - (1 + j)^{-n}}{j} = 500 \frac{1 - (1.0023046)^{-338}}{.0023046} = \$117,309.55$$

## 4.8 Perpetuities

A **perpetuity** is an annuity in which regular payments begin on a fixed date and continue indefinitely. It is sometimes referred to as a perpetual annuity.

The formula for a perpetuity is straightforward, i.e., an initial investment of  $S$  at a per compounding period interest rate of  $i$  can payout  $R = iS$  at the end of every period forever. If less than  $R$  is paid out per period, then the principal will increase and one can eventually increase the payout amount (while still preserving the perpetuity). This could, for example, allow one to adapt for inflation.

In practice, such investment instruments are not offered. However, payments from an endowment fund (e.g., for college scholarships) are similar in concept with the exceptions that the interest rate is not guaranteed (based on multiple investments with varying returns), the endowment typically increases based on new contributions from alumni, and only part of the investment returns are typically distributed.

## 4.9 Continuous Annuities

A **continuous annuity** is a type of an annuity where payments are made continuously but interest is compounded per some given period.

### 4.9.1 Discounted (Present) Value

To compute the present value formula for a continuous annuity, we start with a finite number of payments  $m$  within a given compounding period and then take the limit as  $m$  approaches infinity. To that end, start with an annuity that distributes  $n$  payments in the amount of  $R$  ( $m = 1$  at this point). Interest is compounded at the end of each period at the rate  $i$ . Recall from Section 4.3.1 that the current value of such an annuity is

$$Q = R(1 + i)^{-1} + R(1 + i)^{-2} + \cdots + R(1 + i)^{-n}$$

Next, assume that  $m$  payments in the amount of  $\frac{R}{m}$  are made during each compounding period (see Figure 8).

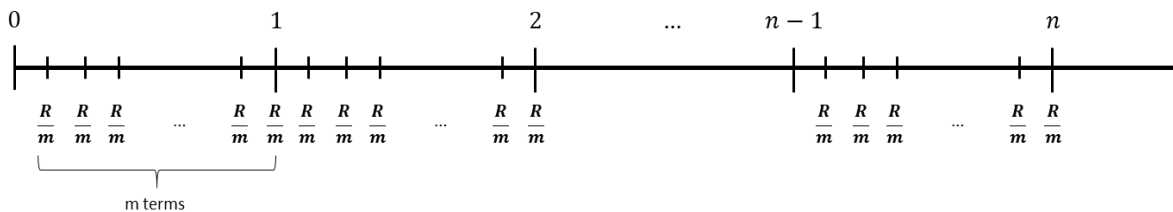


Figure 8. Multiple Payments per Compounding Period

To determine the equivalent interest rate for each of the  $m$  mini-periods, we use the formula  $j = (1 + i)^{s/t} - 1$  (from Section 4.7.1) with  $s = 1$  and  $t = m$  to get  $j = (1 + i)^{\frac{1}{m}} - 1$ . Noting that there are  $mn$  mini-periods (i.e., payments), the present value of the annuity with the  $m$  payments per compounding period is given by

$$\begin{aligned}
Q_m &= \frac{R}{m}(1+j)^{-1} + \frac{R}{m}(1+j)^{-2} + \cdots + \frac{R}{m}(1+j)^{-mn} \\
&= \frac{R}{m} \left[ (1+i)^{-\frac{1}{m}} + (1+i)^{-\frac{2}{m}} + \cdots + (1+i)^{-\frac{mn}{m}} \right] \\
&= \frac{R}{m} \sum_{x=1}^{mn} (1+i)^{-\frac{x}{m}}
\end{aligned}$$

To get the formula for the present value of an annuity with a continuous number of payments in each compound period, we take the limit of the above expression as  $m$  goes to infinity, i.e.,

$$Q_\infty = \lim_{m \rightarrow \infty} Q_m = \lim_{m \rightarrow \infty} \frac{R}{m} \sum_{x=1}^{mn} (1+i)^{-\frac{x}{m}} = R \lim_{m \rightarrow \infty} \sum_{x=1}^{mn} \left(\frac{1}{m}\right) (1+i)^{-\frac{x}{m}} = R \int_0^n (1+i)^{-x} dx$$

The second summation in the above sequence is the right Reimann sum of the function  $(1+i)^{-x}$  from 0 to  $n$ . The typical  $\Delta x$  (used in Riemann sums) is effectively replaced by  $\frac{1}{m}$  which approaches zero as  $m \rightarrow \infty$ .

Solving the definite integral, we get

$$Q_\infty = R \int_0^n (1+i)^{-x} dx = R \left[ \frac{-(1+i)^{-x}}{\ln(1+i)} \right]_{x=0}^{x=n} = R \frac{1 - (1+i)^{-n}}{\ln(1+i)}$$

If  $Q_\infty$ ,  $i$  and  $n$  are known, we can solve for  $R$ , i.e.,

$$R = Q_\infty \frac{\ln(1+i)}{1 - (1+i)^{-n}}$$

As an example, consider two scenarios:

- Scenario A: Ordinary, simple, certain annuity with  $n = 60$  equal payments of \$10,000 and an interest rate of .5% per period
- Scenario B: Annuity with continuous payments over  $n = 60$  periods with an interest rate of .5% per period. The continuous payments equal \$10,000 in each period.

For Scenario A, the present value of the annuity is

$$Q = R \frac{1 - (1+i)^{-n}}{i} = 10,000 \frac{1 - 1.005^{-60}}{.005} = \$517,255.61$$

For Scenario B, the present value of the annuity is

$$Q_\infty = R \frac{1 - (1+i)^{-n}}{\ln(1+i)} = 10,000 \frac{1 - 1.005^{-60}}{\ln(1.005)} = \$518,547.67$$

The present value of the annuity in Scenario B is larger than the present value of the annuity in Scenario A which is to be expected since payments are effectively distributed sooner in Scenario B.

### 4.9.2 Accumulated Value

It is also possible to accumulate value with continuous investments into the annuity. We first consider the case of  $m$  mini-investments into the annuity in the amount of  $\frac{R}{m}$  during each compounding period. This is basically Figure 8 except that money is being added to the annuity rather than removed. If the interest rate for each compounding period is  $i$  then (using the same calculation as in Section 4.9.1) the equivalent interest rate for the mini-period is  $j = (1 + i)^{\frac{1}{m}} - 1$ . The accumulated value of the annuity is given by

$$\begin{aligned} S_m &= \frac{R}{m} + \frac{R}{m}(1 + j) + \frac{R}{m}(1 + j)^2 + \cdots + \frac{R}{m}(1 + j)^{mn-1} \\ &= \frac{R}{m} \left[ 1 + (1 + i)^{\frac{1}{m}} + (1 + i)^{\frac{2}{m}} + \cdots + (1 + i)^{\frac{mn-1}{m}} \right] \\ &= \frac{R}{m} \sum_{x=0}^{mn-1} (1 + i)^{\frac{x}{m}} \end{aligned}$$

To get the accumulated value of an annuity with continuous input investments, we take the limit as  $m$  approaches infinity of  $S_m$

$$S_\infty = \lim_{m \rightarrow \infty} S_m = \lim_{m \rightarrow \infty} \frac{R}{m} \sum_{x=0}^{mn-1} (1 + i)^{\frac{x}{m}} = R \lim_{m \rightarrow \infty} \sum_{x=0}^{mn-1} \left(\frac{1}{m}\right) (1 + i)^{\frac{x}{m}} = R \int_0^n (1 + i)^x dx$$

The second summation is the left Riemann sum of the function  $(1 + i)^x$  from 0 to  $n$ , and thus taking the limit as  $m \rightarrow \infty$  (and  $\frac{1}{m} \rightarrow 0$ ) yields the stated integral. Solving the definite integral, we get

$$S_\infty = R \int_0^n (1 + i)^x dx = R \left[ \frac{(1 + i)^x}{\ln(1 + i)} \right]_{x=0}^{x=n} = R \frac{(1 + i)^n - 1}{\ln(1 + i)}$$

As an example, compare the accumulated value of an ordinary, simple, certain annuity with 120 investments of \$5,000 at an interest rate of .6% per period to a continuous investment annuity with the same parameters.

For the ordinary, simple, certain annuity, the accumulated value is

$$S = R \frac{(1 + i)^n - 1}{i} = 5,000 \frac{1.006^{120} - 1}{.006} = \$875,015.05$$

and for the continuous investment annuity, the accumulated value is

$$S_\infty = R \frac{(1 + i)^n - 1}{\ln(1 + i)} = 5,000 \frac{1.006^{120} - 1}{\ln(1.006)} = \$877,637.48$$

The accumulated value in the continuous case is slightly larger since money is deposited sooner, i.e., during each compounding period rather than at the end of each compounding period.

## 4.10 Supporting Programs

### 4.10.1 Block Annuity Calculator

File name: block\_annuity.py

This program computes the total value of a savings annuity over several blocks, where

- Each block has a fixed interest rate and fixed payment (i.e., investment) per period.
- The interest rate and payment amount may vary among the blocks.
- The blocks can be of different sizes, i.e., different number of periods.

There is no pop-up GUI for this program. The input and output are done via the terminal associated with the Python code. For input to this program, a CSV file needs to be created (put this in the same directory as the program file). The program will ask for the file name (leave off .csv when you enter the file name). Each line of the CSV file represents one block of an annuity. The format for each line is as follows:

interest rate per period, saving amount per period, number of periods

As an example, put the following lines in a CSV file and run the program:

```
0.0042, 500, 10
0.0033, 400, 25
0.0038, 300, 30
0.0045, 700, 15
0.0047, 200, 20
```

The output is as follows (this is displayed on the terminal associated with the Python code):

The value of each block without interest accrued during subsequent blocks:

```
Block 0: $5,095.57
Block 1: $10,406.20
Block 2: $9,513.95
Block 3: $10,837.29
Block 4: $4,183.74
```

The value of each block at the end of the annuity's term, i.e., with the interest accrued during subsequent blocks:

```
Block 0: $7,283.76
Block 1: $13,698.88
Block 2: $11,177.33
Block 3: $11,902.79
Block 4: $4,183.74
```

Total value of the annuity at the end of its term:  
\$48,246.50

#### 4.10.2 Annuity Payout Calculator

File name: simple\_annuity\_payout.py

This program does computations for ordinary, simple, certain annuities. It is assumed that the compounding and the periodic payouts happen on the same schedule, e.g., at the end of each



month. The required input values are the initial investment amount in the annuity, the per period interest rate (**as a decimal**, e.g., .05 rather than 5%), and either the desired number of payouts or the amount of each payout (but not both).

- If, in addition to the required inputs, the number of payouts is provided, the program will compute the amount of each payout.
- If, in addition to the required inputs, the desired amount of each payout is provided, the program will compute the number of payouts until the annuity is depleted.
  - If the desired payout amount is less than the accrued interest per period, the payouts can continue indefinitely. In this case, the program will output “infinite” for the number of payouts.
  - If the desired payout amount is greater than the value of the original investment at the end of the first period, then the payout is not possible and an error message is generated (displayed on the terminal associated with the Python code).

#### 4.10.3 Deferred Annuity Calculator

File name: deferred\_annuity.py

This program does computations concerning deferred annuities. It is assumed that the compounding and the periodic payouts are on the same schedule, e.g., at the end of each quarter. The required input values are the interest rate per period, desired number of payouts and the number of deferral periods.

- If initial investment is given, the amount of each payout will be computed.
- If the desired payout amount is provided, the program will compute the required initial investment.

This program can be used to solve the example presented in Section 4.5. In the program, leave the Initial Investment blank (as this is to be computed), enter .004166666 for the Interest per Period, 60 for the Number of Payouts, 84 for the Number of Deferral Periods and 2500 for the Amount of Each Payout.

### 4.11 Exercises

1. Mr. Periwinkle agrees to purchase an annuity from a bank (this is intended to be a savings plan). He is to deposit \$750 at the end of each month for 5 years, with a guaranteed nominal annual interest rate of 3% (compounded monthly). How much is Mr. Periwinkle's annuity worth at term? **Hint:** Use the block annuity calculator with a single block. The input is a csv file with just one line, i.e., .0025, 750, 60. Note that .0025 is the month rate based on a 3% nominal annual interest rate.
2. Continuing with the previous exercise, assume the term and interest rate is the same but Mr. Periwinkle asks the bank how much he would need to save each month to reach \$50,000 at the end of the term. **Hint and Answer:** Use the formula for  $R$  in terms of  $i$ ,  $S$  and  $n$  in Section 4.2.2. The answer is \$773.44 (rounding up to the next cent). You can check the answer using the block annuity calculator.
3. Joan and John Song are saving for a down payment on a house. An investment company has offered them a savings annuity that pays a guaranteed nominal annual interest rate of 3% (compounded monthly) for the first 30 months, which may change (depending on market conditions) for the remaining 30 months of the annuity. However, it is guaranteed that the interest rate will drop at most .5% during the second 30 months. The couple would like to know how much the annuity will be worth at term if they invest \$1,000 per month, assuming the least optimistic interest rate for the second 30 months of the annuity, i.e., 2.5%. **Hint and Answer:** Use the block annuity calculator with two blocks to determine that the value of the annuity at term is \$64,041.91.
4. In the previous example, assume that the Song's really needed \$75,000. If the term (5 years) and interest rates are the same, how much do they need to save each month to reach their goal? **Hint:** Clearly, there are multiple possible answers here. One approach is to increase the payment amount in the first block and leave the payment amount in the second block at \$1,000/month. Use the block annuity calculator and experiment with various payment amounts for the first 30 months. For example, a monthly payment of \$1,333 for the first 30 months and \$1,000/month in the second 30 months yields a final accrued value of \$75,070.14.
5. Mr. Trubisky has just retired early after a lucrative career as a professional athlete. He has \$10,000,000 in a savings account and would like to invest half of that amount in an annuity. How long will the annuity last at 3% nominal annual interest rate (compounded monthly) if he takes a payment of \$50,000 per month? **Hint:** Use the simple annuity payout calculator.
6. After doing the computation, Mr. Trubisky realizes that his withdraw amount will last about 9.6 years. Given that he is now 35 years old, he wants the payouts to continue for 50 years. With the interest rate conditions being the same, how much can Mr. Trubisky be paid each month if the annuity is to last 50 years? **Hint:** Use the annuity payout calculator.
7. How can the block annuity calculator be used to calculate an annuity due where the payments are made at the beginning of each period and there is one accrual period after the last payment? **Hint:** Use two blocks, with the second block consisting of a \$0 payment but at the same interest rate as the first block.

8. Test out your methodology from Exercise 7 on the following annuity due: \$150 payment (at the beginning of each monthly payment period), for four years and at a nominal annual interest rate of 6% (compounded monthly). Check your answer by using the formula for the accumulated value of an annuity due in Section 4.4.
9. Milton invests \$75,000 in a deferred annuity at a nominal annual interest rate of 4% (compounded monthly) with the first of 90 payments to start in 5 years. Compute the amount of the regular payments. **Hint:** use the deferred annuity calculator.
10. As a variation of the previous exercise, assume Milton wants to get payments of \$1,175.50 per month for 90 payments with the first payment 5 years from now. How much money does he need to invest up front in the deferred annuity? **Hint:** use the deferred annuity calculator.
11. Modify the Python program for the deferred annuity calculator to support the case where the interest rate during the time of deferral is different from the interest rate during the payment periods. **Hint:** You need to add another input for the interest rate during the deferral period, and then use that rate in the calculation of interest during the deferral period.
12. Compute the accumulated value of a general annuity with term 13 years, where the interest is compounded 7 times per year at a nominal annual rate of 7% and payments into the annuity are made 23 times per year in the amount of \$500 per payment. **Hint:** The interest rate per compounding period is  $i = \frac{.07}{7} = .01$ . If compounding is done at the same rate as the payments, the equivalent rate is  $j = (1 + .01)^{\frac{7}{23}} - 1 = .003032952$ . Next, use the formula for the accumulated value of a simple annuity with per period interest rate  $j$ ,  $n = 13 \times 23 = 299$  and  $R = 500$ . Check your answer with the block annuity calculator using a csv file with the following line: .003032952,500,299.
13. Compare the present value of an ordinary, simple, certain annuity with 360 equal payments of \$2,000 at an interest rate of .005 per period with a continuous annuity of the same term, payout, and interest rate. **Hint:** See the example in Section 4.9.1. **Answer:** \$333,583.23 for the ordinary, simple, certain annuity and \$334,416.49 for the continuous annuity.

## 5 Repayment of Loans

### 5.1 Amortization of Loans

#### 5.1.1 General Formula and Equivalence to Annuities

“**Amortization**” refers to the distribution of payments into multiple equal installments where each payment consists of both principal and interest. Amortization is mainly used in loan repayments (e.g., home mortgages and car loans) and in sinking funds (i.e., money set aside over a period of time to fund a future capital expense, or repayment of a long-term debt). Repayments are divided into equal amounts for the duration of the loan. However, the amount of interest and principal pay-back in each installment varies, as demonstrated in the example that follows below. This process is known as Equated Monthly Installment (EMI).

(**Note:** In what follows, we say “payment” although technically it is a “repayment” of money that had been previously loaned, since “payment” is the term more commonly used in the literature.)

Make the following assignments:

$P$  = principal amount borrowed

$A$  = equal periodic amortization payment

$i$  = the interest rate per compounding period

$n$  = total number of payment periods.

It is assumed that the compounding and payment periods coincide.

When doing the calculation, it is important to note that the loanee (person who receives the loan) must pay back the amount of interest due at the end of each payment period.

At the end of the first payment period, the amount still owed to the loaner (“the balance”) is the original principal plus the interest due for the first period minus the amount of principal paid back by the loanee to the loaner, i.e.,

$$B_1 = P + iP - A = (1 + i)P - A$$

At the end of the second payment period, the balance is

$$B_2 = (1 + i)B_1 - A = (1 + i)^2P - A(1 + i) - A$$

At the end of the third payment period, the balance is

$$\begin{aligned} B_3 &= (1 + i)B_2 - A \\ &= (1 + i)^3P - A(1 + i)^2 - A(1 + i) - A \end{aligned}$$

In general, after payment period  $k$ , the balance is

$$B_k = (1 + i)^kP - A[(1 + i)^{k-1} + \dots + (1 + i) + 1]$$

At the end of  $n$  payment periods (for  $n \geq 1$ ), the balance of the mortgage should be zero, i.e.,

$$0 = B_n = (1 + i)^nP - A[(1 + i)^{n-1} + \dots + (1 + i) + 1]$$

which implies

$$(1 + i)^nP = A[(1 + i)^{n-1} + \dots + (1 + i) + 1]$$

Solving for  $A$  and noting that the terms in the square brackets constitute a geometric series whose sum we know how to calculate from Section 2.1.2, we get

$$A = \frac{iP(1+i)^n}{(1+i)^n - 1}$$

which can be rewritten as

$$A = \frac{iP}{1 - (1+i)^{-n}}$$

The above formula should look familiar as it is equivalent to the formula for regular payments from an ordinary, simple, certain annuity (see Section 4.3.1). A mortgage can be thought of as an annuity (purchased by a lending institution such as a bank) from an individual. The discounted (present) value of the annuity ( $Q$ ) is the amount of the mortgage ( $P$ ), and the annuity payments ( $R$ ) are essentially the periodic amortization payments ( $A$ ).

### 5.1.2 Breakdown of Interest versus Principal Repayment

As noted, each payment is a combination principal payback and interest payment where the sum of the two is equal to  $A$  for each payment. However, the ratio of principal payback to interest payment changes over time. At the end of each payment period, the interest owed is  $i$  times the amount of outstanding balance at the beginning of the period. Thus, the interest owed at the end of period  $k$  is given by the following formula:

$$I_k = iB_{k-1} = i(1+i)^{k-1}P - iA[(1+i)^{k-2} + \dots + (1+i) + 1]$$

and of course, the amount of principal repayment at the end of period  $k$  (call it  $R_k$ ) is given by

$$R_k = A - I_k$$

The interest owed at the end of period  $i$  can also be represented recursively, i.e.,

$$I_k = (i+1)I_{k-1} - iA$$

which is convenient for programming purposes.

As an example, consider a 3-year loan on a car for \$10,000 at a 3% annual interest rate where the loanee pays back the loan on a monthly basis. In terms of the above formulas,  $P = 10,000$ ,  $i = \frac{3}{100 \times 12} = .0025$  and  $n = 12 \times 3 = 36$ . Using the formula for equal periodic amortization payment yields  $A = \$290.81$ . A breakdown (interest versus principal) for the first four repayments is given in Table 3. Given the relatively short term of the loan, the principal payments outweigh the interest payments from the start. For home mortgages (e.g., 30 years), the interest payments are greater than principal payback at the beginning of the loan.

Table 3. Interest-Principal Repayment Breakdown Example

| Period | Interest Owed Formula                        | Interest to be paid for given period | Principal to be paid for given period |
|--------|--|--------------------------------------|---------------------------------------|
| 1      | $I_1 = iP$                                   | \$25.00                              | \$265.81                              |
| 2      | $I_2 = i(1+i)P - iA$                         | \$24.34                              | \$266.48                              |
| 3      | $I_3 = i(1+i)^2P - iA(1+i) - iA$             | \$23.67                              | \$267.14                              |
| 4      | $I_4 = i(1+i)^3P - iA(1+i)^2 - iA(1+i) - iA$ | \$23.00                              | \$267.81                              |

### 5.1.3 Rounding Mortgage Payments

It may be desirable to round upward (or downward) the periodic mortgage payments to the next dollar or 10 dollar value, and then adjust the final payment downward (or upward). The algorithm for doing this is as follows:

1. Determine the periodic payment  $A$
2. Round  $A$  up to the next dollar or 10 dollar value. Call this new value  $\hat{A}$ .
3. Determine the penultimate balance assuming the amount  $\hat{A}$  was used for all past payments as well as the penultimate payment and we have

$$B_{n-1} = (1+i)^{n-1}P - \hat{A}[(1+i)^{n-1} + \dots + (1+i) + 1] = (1+i)^{n-1}P - \hat{A} \frac{(1+i)^{n-1} - 1}{i}$$

4. Calculate the adjusted last payment which is

$$B_{n-1} + iB_{n-1} = (1+i)B_{n-1} = (1+i)^n P - \hat{A} \frac{(1+i)^n - (1+i)}{i}$$

Actually, one can go directly to the formula in Step 4 and skip Step 3, or do Step 3 and then just multiply by  $(1+i)$ .

As an example, consider a \$200,000 mortgage at a 6% nominal annual interest rate with payments due each month. The term of the mortgage is short (just 10 years). The 6% nominal annual interest rate means the monthly interest rate is  $\frac{.06}{12} = .005$ .

The monthly payment is  $A = \$2,220.41$  which we round up to  $\hat{A} = \$2,221$ . Using the formula from Step 4 in the above algorithm gives us the amount of the final payment:

$$(1+i)^n P - \hat{A} \frac{(1+i)^n - (1+i)}{i} = (1.005)^{120}(200,000) - (2,221) \frac{(1.005)^{120} - 1.005}{1.005} = \$2,124.32$$

Alternately, we could round the payments to the nearest multiple of \$10 (\$2,220 in this case) and then calculate the final payment, i.e., \$2,287.20 which as one would expect, is slightly higher than the other 119 payments of \$2,220.

We could even round to the nearest multiple of \$100 (i.e., \$2,200) and compute the final payment, i.e., \$5,544.78.

In terms of rounding down, keep in mind that the loanee is expected to pay at least the amount of interest owed each period, and so we cannot go beyond that limit.

In terms of rounding up, it is possible to round up to the point where some of the last several payments are not needed since all the principal is already paid off.

## 5.2 Balloon Loans

### 5.2.1 Concept and Formulas

In Section 3.1, an example of a balloon loan was given. In the example, payment of all principal was deferred until the last payment of the loan. All the intermediate payments only entailed interest payment on the initial principal  $P$ . In the more general case, an amount  $B$  (less than or equal to the original principal) is made at the last payment. The loanee does, in some scenarios, repay some of the principle along the way (i.e.,  $P - B$ ).

In the following calculations, the terms  $A, P, B_n$  and  $i$  are as defined in Section 5.1.1. It is assumed that  $n$  equals payments are made in the amount  $A$  and then a final payment in the amount of  $B$  (at the same time as the  $n^{\text{th}}$  payment).

Relying on the derivation from Section 5.1.1, at the end of  $n$  periods (for  $n \geq 1$ , the total amount owed to the loaner should be  $B$  (the balloon payment). Using this fact, we solve for  $A$  in the equations below:

$$B = B_n = (1 + i)^n P - A[(1 + i)^{n-1} + \dots + (1 + i) + 1]$$

$$B = (1 + i)^n P - A \frac{(1 - (1 + i)^n)}{-i}$$

$$A \frac{((1 + i)^n - 1)}{i} = (1 + i)^n P - B$$

$$A = \frac{iP(1 + i)^n - iB}{(1 + i)^n - 1}$$

If  $A, i, B$  and  $n$  are known, the above equation can be solved for  $P$ :

$$P = \frac{A[(1 + i)^n - 1] + iB}{i(1 + i)^n}$$

which can also be written as

$$P = A \frac{1 - (1 + i)^{-n}}{i} + B(1 + i)^{-n}$$

If  $B = 0$ , the formula for  $A$  is the same as for a non-balloon mortgage. If  $B = P$  (all the principal is paid at the end of the term), then

$$A = \frac{iP(1 + i)^n - iP}{(1 + i)^n - 1} = iP$$

which is to be expected since one is only paying interest during the course of the loan, with the principal repayment deferred until the end of the loan.

There are variants on the assumptions for balloon loans which lead to slightly different formulas for  $A$ . For example, one could assume that  $n$  payments in the amount of  $A$  are made, followed by payment  $n + 1$  in the amount of  $B$ . In this case,  $B = B_n + iB_n = B_n(1 + i)$  which when solved for  $A$  leads to

$$A = \frac{iP(1+i)^n}{(1+i)^n - 1} - \frac{iB}{(1+i)^{n+1} - (1+i)}$$

### 5.2.2 Example – Balloon Loan as a General Annuity

A car loan stipulates payments of \$500 at the end of each quarter for 5 years, with an additional payment of \$7,000 at the end of 5 years. What is the present value of the loan at a 6% nominal annual interest rate (compounded monthly)?

The monthly rate is  $i = \frac{.06}{12} = .005$  which can be converted to an equivalent quarterly rate as follows:  $j = (1+i)^{s/t} - 1 = 1.005^{\frac{12}{4}} - 1 = 0.015075125$

We are essentially being asked to work backwards and find the cost of the car, given payments ( $A = 500$ ), interest rate ( $j = 0.015075125$ ), the balloon amount ( $B = 7,000$ ) and the number of payment ( $n = 20$ ). Using the formula for  $P$  from Section 5.2.1 yields

$$P = \frac{A[(1+j)^n - 1] + jB}{j(1+j)^n} = \frac{500[.015075125^{20} - 1] + (.015075125)(7,000)}{(.015075125)(.015075125^{20})} = \$13,767.57$$

## 5.3 Computation of Remaining Balance on a Loan

In Section 5.1.1, we used a retrospective method to determine the balance on a loan at the end of a given period. In the formula below,  $B_k$  is the accumulated value of the debt at time  $k$  minus the accumulated value of the payments at time  $k$ . This difference is the balance owed at time  $k$ .

$$B_k = (1+i)^k P - A[(1+i)^{k-1} + \dots + (1+i) + 1]$$

applying the formula for the sum of a geometric series to the above, we get the more compact form

$$B_k = (1+i)^k P - A \frac{(1+i)^k - 1}{i}$$

Another approach (the prospective method) is to sum the present (discounted) value of the remaining  $n - k$  payments, i.e.,

$$B_i = A(1+i)^{-1} + A(1+i)^{-2} + \dots + A(1+i)^{-(n-k)}$$

$$B_i = A(1+i)^{-1} [1 + (1+i)^{-1} + \dots + (1+i)^{-(n-k)-1}]$$

$$B_i = A(1+i)^{-1} \frac{1 - (1+i)^{-(n-k)}}{1 - (1+i)^{-1}}$$

$$B_i = A \frac{1 - (1+i)^{-(n-k)}}{i}$$

The first formula works even if the final payment is irregular (e.g., a balloon payment) while the second formula does not work in that case. However, the second formula works even when the original amount of the loan is unknown.

The two formulas are algebraically equivalent, as is shown below:

At the end of the loan (focal date), the dated value of the debt minus the dated value of the payments should equal zero, i.e.,

$$0 = (1+i)^n P - A[(1+i)^{n-1} + \dots + (1+i) + 1]$$



which is equivalent to

$$(1+i)^n P = A \frac{(1+i)^n - 1}{i}$$

The above can be rewritten as

$$P = A \frac{1 - (1+i)^{-n}}{i}$$

Subtracting  $A \frac{1 - (1+i)^{-k}}{i}$  from both sides, we get

$$P - A \frac{1 - (1+i)^{-k}}{i} = A \frac{1 - (1+i)^{-n}}{i} - A \frac{1 - (1+i)^{-k}}{i} = A \frac{(1+i)^{-k} - (1+i)^{-n}}{i}$$

Multiplying by  $(1+i)^k$  yields

$$P(1+i)^k - A \frac{(1+i)^k - 1}{i} = A \frac{1 - (1+i)^{-(n-k)}}{i}$$

Thus, the two formulas for  $B_k$  are algebraically equivalent.

## 5.4 Refinancing a Loan

When interest rates drop, it is not unusual for folks with existing mortgages to consider refinancing. Refinancing entails the termination of an existing mortgage (perhaps with a penalty to pay) and the initiation of a new mortgage (which will also involve various costs, e.g., legal fees, application fee, title search, local taxes, and appraisal fee). A typical goal for the borrower is to lower their monthly payments and to lower the amount of interest paid for the remainder of the mortgage (i.e., from the time of refinancing).

For example, the Butterfields have a \$300,000, 30-year mortgage which is paid back to the bank on a monthly basis at a rate of .005 per month (6% nominal annual rate). They can now (exactly 5 years into their initial mortgage) get a 30-year mortgage at 4% nominal annual rate (.003 per month). Assume the total cost of terminating the existing mortgage and getting a new mortgage is \$6,000. Does it make sense for the Butterfields to refinance?

To answer the question, we first compute the monthly payment for the existing mortgage (which is \$1,798.65) and the balance on the initial mortgage (which is \$279,163.07).

So, the new mortgage would need to be in the amount of \$279,163.07. At the stipulated interest rate and term, the new mortgage would entail monthly payments of \$1,332.76.

- The remaining payments for the initial loan (over the next 25 years), if not refinanced, equal \$539,595.47 of which \$260,432.40 are interest payments.
- The payments in the refinancing scenario, which includes the costs to terminate the initial loan and to initiate the new loan, equal \$485,793.87. The interest paid in this scenario is \$206,630.80.

Assuming the Butterfields are not going to sell their house before they recover the \$6,000 in refinancing costs (which will take 13 months), it makes sense to refinance.

A relevant question is “at what interest rate does it make sense to refinance?” Table 4 shows the total payments (for the new loan) and payments per period (in this case, per month) for various interest rates, starting with the interest rate for the existing mortgage. As one can see, the total

payments (for the new loan) are less than that for the remaining part of the existing loan up to a nominal annual interest rate of up to a little more than 5.86% but less than 5.92%.

The first line of the Table 4 below is the status quo, i.e., keep the existing loan and do not refinance. The amount listed under “total payments” in the first line is actually about left to be paid under the conditions of the existing loan.

Table 4 was generated by the “Loan Refinance - Breakeven Interest Rate Analyzer”, see Section 5.5.5 for a description of this program.

*Table 4. Interest Rate Analysis for Refinancing*

| Interest Rate per Period | Nominal Annual Interest Rate | Total Payments      | Payment per Period |
|--------------------------|------------------------------|---------------------|--------------------|
| <b>0.005000</b>          | <b>6.00%</b>                 | <b>\$539,595.47</b> | <b>\$1,798.65</b>  |
| 0.003333                 | 4.00%                        | \$448,057.64        | \$1,473.53         |
| 0.003383                 | 4.06%                        | \$450,836.89        | \$1,482.79         |
| 0.003433                 | 4.12%                        | \$453,625.56        | \$1,492.09         |
| 0.003483                 | 4.18%                        | \$456,423.60        | \$1,501.41         |
| 0.003533                 | 4.24%                        | \$459,230.99        | \$1,510.77         |
| 0.003583                 | 4.30%                        | \$462,047.70        | \$1,520.16         |
| 0.003633                 | 4.36%                        | \$464,873.71        | \$1,529.58         |
| 0.003683                 | 4.42%                        | \$467,708.98        | \$1,539.03         |
| 0.003733                 | 4.48%                        | \$470,553.47        | \$1,548.51         |
| 0.003783                 | 4.54%                        | \$473,407.17        | \$1,558.02         |
| 0.003833                 | 4.60%                        | \$476,270.04        | \$1,567.57         |
| 0.003883                 | 4.66%                        | \$479,142.04        | \$1,577.14         |
| 0.003933                 | 4.72%                        | \$482,023.16        | \$1,586.74         |
| 0.003983                 | 4.78%                        | \$484,913.35        | \$1,596.38         |
| 0.004033                 | 4.84%                        | \$487,812.58        | \$1,606.04         |
| 0.004083                 | 4.90%                        | \$490,720.82        | \$1,615.74         |
| 0.004133                 | 4.96%                        | \$493,638.05        | \$1,625.46         |
| 0.004183                 | 5.02%                        | \$496,564.21        | \$1,635.21         |
| 0.004233                 | 5.08%                        | \$499,499.30        | \$1,645.00         |
| 0.004283                 | 5.14%                        | \$502,443.26        | \$1,654.81         |
| 0.004333                 | 5.20%                        | \$505,396.07        | \$1,664.65         |
| 0.004383                 | 5.26%                        | \$508,357.69        | \$1,674.53         |
| 0.004433                 | 5.32%                        | \$511,328.09        | \$1,684.43         |
| 0.004483                 | 5.38%                        | \$514,307.24        | \$1,694.36         |
| 0.004533                 | 5.44%                        | \$517,295.10        | \$1,704.32         |
| 0.004583                 | 5.50%                        | \$520,291.63        | \$1,714.31         |
| 0.004633                 | 5.56%                        | \$523,296.80        | \$1,724.32         |
| 0.004683                 | 5.62%                        | \$526,310.59        | \$1,734.37         |
| 0.004733                 | 5.68%                        | \$529,332.94        | \$1,744.44         |
| 0.004783                 | 5.74%                        | \$532,363.83        | \$1,754.55         |

| Interest Rate per Period | Nominal Annual Interest Rate | Total Payments | Payment per Period |
|--------------------------|------------------------------|----------------|--------------------|
| 0.004833                 | 5.80%                        | \$535,403.22   | \$1,764.68         |
| 0.004883                 | 5.86%                        | \$538,451.07   | \$1,774.84         |
| 0.004933                 | 5.92%                        | \$541,507.36   | \$1,785.02         |

## 5.5 Supporting Programs

### 5.5.1 Annuity Payout Calculator

As described in Section 4.10.2

### 5.5.2 Amortization Calculator

File name: amortizations.py

The program offers three different capabilities

- Compute the periodic payments on a mortgage given the loan amount, annual interest rate (but leave off the percentage sign), number of payments per year and the length of the loan in years (doesn't need to be a whole number).
- Compute the maximum affordable loan given the interest rate per year, number of payments per year, length of the loan in years and the maximum affordable payment per period.
- Compute the interest versus principal payback breakdown given the loan amount, interest rate, number of payments per year and the length of the loan in years. The result is sent to a file named breakout.csv (in the same directory as the program). The name of the output file can be changed by editing the Python program on line 106.

### 5.5.3 Balloon Loan Calculator

File name: balloon\_loan.py

This calculator offers two capabilities, i.e.,

- Compute the periodic payment amount for a balloon loan given the loan amount, interest rate per period (**as a decimal**, e.g., .03 for 3%), total number of payments and the amount of the balloon payment.
- Compute the present value (initial amount of loan) of a balloon loan given the interest rate per period, total number of payments, the amount of the balloon payment and the amount of a periodic payment.

### 5.5.4 Loan Refinance Calculator

File name: loan\_refinance.py

This program analyzes the refinancing of a loan (such as a mortgage) given the following input:

- the amount of the initial (existing loan)
- interest rate per period for the initial loan (as a decimal, e.g., .03 for 3%)

- number of periods in the initial loan
- number of periods completed in the initial loan when refinancing
- penalties on termination of initial loan and financing cost for the new loan
- interest rate per period for the refinanced loan
- number of periods for the refinanced loan. There is not an assumption that the number of periods in the refinanced loan equal the number of periods left in the original loan. For example, you can start with a 30 loan, and refinance after 10 years with say a 15 year loan.

It is assumed that the period for the original loan (e.g., a month) is the same as the period for the new loan.

The output gives decision information concerning whether to refinance or not.

For example, take the following as input to the program (exactly as-is, e.g., no commas or percentage signs):

|  |        |
|--|--------|
| Amount of initial loan:                      | 300000 |
| Interest rate per period for initial loan:   | .005   |
| Number of periods for initial loan:          | 360    |
| Number of periods completed in initial loan: | 60     |
| Penalties and refinancing costs:             | 6000   |
| New interest rate per period:                | .0033  |
| Number of periods for new loan:              | 360    |

The result from the program is as follows:

The balance on the initial loan is \$279,163.07  
 The periodic payment for the initial loan is \$1,798.65  
 The periodic payment for the new loan is \$1,326.34  
 The remaining payments for initial loan, if not refinanced, equal \$539,595.47  
 of which \$260,432.40 are interest payments.  
 The payments in the refinancing scenario, which includes the costs to  
 terminate the initial loan and to initiate the new loan, equal \$483,481.54  
 The interest paid in this scenario is \$204,318.47  
 It will take 13 periods to recover the cost of refinancing.

### 5.5.5 Loan Refinance - Breakeven Interest Rate Analyzer

File name: loan\_refinance\_int\_rate\_analyzer.py

This program analyzes the refinancing of a loan (such as a mortgage) given

- the amount of the initial (existing loan)
- interest rate per period for the initial loan (as a decimal, e.g., .03 for 3%)
- number of periods in the initial loan

- number of periods that have transpired in the initial loan when the new loan is to start
- the amount of penalties on termination of the initial loan plus the cost of the new loan
- interest rate per period where the analysis is to start
- number of periods for the new loan
- number of payments per year for the new loan.

It is assumed that the period for the original loan (e.g., a month) is the same as the period for the new loan.

The program prepares a list of scenarios for various interest rates up to the point where the refinancing rate leads to a situation where the remaining payments are more than that of the original loan.

The output of the analysis is sent to a file named `breakeven_int.csv`. An example output (with some reformatting) is shown in Table 4.

## 5.6 Exercises

1. Complete Table 3 and generate the entire repayment schedule. **Hint:** Use the “Compute Interest – Principal breakdown” feature in the amortization calculator.
2. Assume that a potential home buying couple has determined that they can afford a maximum of \$1,500 per month for a home mortgage. Further, they have found the best 15-year mortgage rate to be 3%, the best 20-year mortgage rate to be 3.5% and the best 30-year mortgage rate to be 3.75%. For all three scenarios, assume that payments on the mortgage are made monthly. What is the maximum mortgage (initial principal) that they can afford? **Hint:** Solve the equation  $A = \frac{iP(1+i)^n}{(1+i)^n - 1}$  for  $P$  and use the derived formula to determine the value for  $P$  in each of the three cases. Check your calculations using the “Max affordable loan given interest rate ...” feature in the amortization calculator. **Answer:** In terms of being able to afford the most expensive house, the 30-year mortgage is the best option. It allows the couple to purchase a \$323,893.22 house.
3. For a \$300,000 30-year mortgage at a nominal annual interest rate of 4% with monthly payments back to the lender, determine the amount of the monthly payments. Adjust the monthly payment to the nearest multiple of \$10 and determine the final payment in this case. **Hint:** Use the formula from Step 4 in Section 5.1.3. **Answer:** From the amortization calculator, we determine the monthly payment to be \$1432.25. So, the adjusted monthly payment (to the nearest \$10) is \$1,430 and the final payment (based on the modified monthly payment) is \$2,988.76. If we just wanted to eliminate the cents (i.e., pay \$1432 per month), the final payment would be \$1602.66. If we adjust the monthly payment to the nearest multiple of \$100 (i.e., \$1,400), then the final payment is a surprising \$23,780.24.
4. Continuing with the previous problem, how many payments would it take if the monthly payment is increased to \$1,500? **Hint:** Use the simple annuity payout calculator with \$300,000 as the initial investment (from the point of view of the loaning entity) and \$1,500 as the amount of each payout. Note that you need to enter the monthly interest as a decimal, i.e.,  $.04/12 = .00\overline{3}$ . So, with a modest increase in the monthly payment, one can pay off this mortgage in 330.13 months or about 2.5 years early.
5. For a car loan of \$30,000 at a nominal annual rate of 2% (compounded monthly) for 3 years, and a balloon payment of \$15,000 at the same time as the last monthly payment, how much is each monthly payment? **Hint:** Use the balloon formula, i.e.,  $A = \frac{iP(1+i)^n - iB}{(1+i)^n - 1}$ , and check your answer with the balloon loan calculator. **Answer:** \$454.64.
6. Continuing with the previous exercise, how does the answer change if the loanee makes the balloon payment one period after the last regular payment? **Hint:** Use the alternate formula at the end of Section 5.2.1. **Answer:** \$455.31 which is slightly more than the answer to Exercise 5, as expected since the balloon payment is made a month later.
7. Jedediah (goes by “Jed”) has a 30-year mortgage in the amount of \$5 million at a nominal annual interest rate of 3% (compounded monthly and with monthly payments back to the bank). How much is the balance on the mortgage at the end of 73 months? **Hint:** Use either of the formulas for the remaining balance from Section 5.3. Check your answer with the loan refinance calculator (with  $\frac{.03}{12} = .0025$  for the interest rate per period). Use the amount 0 in the penalties and costs field. The interest rate and number of periods for the refinanced loan are not relevant, but to avoid an error, use .0025 and 360.

8. Continuing with the previous exercise, Jed has the opportunity (73 months into the original mortgage) to refinance at a 1% nominal annual interest rate (compounding and payments are monthly). The new mortgage would be for 30 years. There is no cost to terminate the original mortgage (as written in the mortgage agreement if he uses the same bank to refinance, which he is doing) but the new mortgage costs \$55,000. Should Jed refinance, assuming he intends not to sell the house in the next ten years? **Hint:** *Use the loan refinancing calculator.*
9. Continuing with the previous exercise, what is roughly the highest interest rate (above 1% and less than 3%) at which it still makes sense for Jed to refinance in terms of having a lower total (remaining) payments? **Hint:** *Use the “Loan Refinance - Breakeven Interest Rate Analyzer.”*

## 6 Bonds

### 6.1 Terminology

A **bond** is a fixed income instrument that represents a loan made by an investor to a borrower (usually a corporation or government). When an entity needs to raise money for a long period of time (e.g., a city looking to build a sports stadium), they issue bonds (also known as debentures) which are sold to investors. Bonds are a form of debt.

A bond is essentially a contract between the issuer (borrower) and the investor (lender) that specifies:

- The **face value**, or the denomination, of the bond, which is stated on the front of the bond. This is usually (but not always) the amount of money to be paid to the borrower when the bond matures.
- The **date of maturity** (also known as the redemption date) is when the loan will be repaid. In some cases, it is possible to redeem a bond after the maturity date but typically, no interest is accumulated after the maturity date. For example, United States Series EE savings bonds can be held beyond the maturity but no interest accrues after the maturity.
- The rate of interest that the bond issuer pays the borrower is known as the **coupon rate**. It is critical to keep in mind that the coupon rate is on the face value of the bond and not on the purchase price of the bond.
  - There is also something called a **zero-coupon bond** which does not have a coupon rate and does not pay regular interest to the holder but rather is sold at a discount and renders a profit at maturity when the bond is redeemed for its full face value.
- The price at which the bond issuer originally sells the bonds is known as the **issue price**. Since bonds can be resold, there could be additional purchase prices. It is important to note that the coupon rate does not change when a bond is resold.
- The amount of money that will be paid on the date of maturity is known as the **redemption value**. If the redemption value is the same as the face value, the bond is said to be **redeemed at par**.
- The **yield** of a bond is the rate of return the bond generates. This is not to be confused with a bond's coupon rate which is based on the face value of a bond.
  - An example may help distinguish the two concepts. If a \$100 bond with a 6% coupon rate sells for \$100, then the current yield is also 6%. However, because of market price fluctuation, it may be possible to purchase this bond for the price of \$80. So, if a \$100 (face value) bond is purchased for \$80, the yield is 7.5% since the \$6 annual coupon payment is now 7.5% of the \$80 purchase price. If the \$100 (face value) bond was purchased for \$120, then the yield will be just 5%.

A **callable bond** (also known as a redeemable bond) allows the issuer to redeem the bond before it reaches the stated maturity date. In other words, on the call date, the bond issuer has the option (not the obligation) to buy back the bond from the bond holder at a defined call price. The issuer of a callable bond may choose to call their bonds if market interest rates move lower, and then re-issue another series of bonds at a lower rate. Callable bonds compensate investors for the



possibility of the bond being called by offering a higher interest rate than comparable non-callable bonds.

**Savings bonds** (as offered by the United States Treasury) can be redeemed any time after the maturity date. The investor (bond holder) will receive the face value of the bond (typically the purchase price) plus accrued interest.

**Marketable bonds** (e.g., corporate bonds or government bonds) do not allow for redemption before the maturity date. If the holder of a marketable bond needs to liquidate a bond before the maturity date, the holder must sell the bond on the open bond market at a price that depends on current interest rates and other market factors (e.g., the financial stability of the bond issuer).

## 6.2 Similarity between Bonds and Balloon Loans

Consider a bond that has the following characteristics:

- coupon rate  $j$  (per compounding period)
- face value  $F$
- periodic payment  $A$  (computed by applying the coupon rate against the face value of the bond, i.e.,  $A = j \times F$ )
- a final payout (redemption value) of  $B$
- a purchase price  $P$
- a yield  $y$  (per the same compounding period as the coupon rate)

To determine the price of the bond, we need to determine the present value (at time of purchase) for each interest payout and the final payout.

This is the same problem as computing the principal for a balloon loan given the per period interest rate, payout per period and the balloon payout at the end of the loan. The only difference is that the entity providing the loan is the bond purchaser and the entity taking the loan is the bond issuer. Simply restating the formula for the principal of a balloon loan, we also have the formula for the price of a bond (with the conditions stated above):

$$P = A \frac{1 - (1 + y)^{-n}}{y} + B(1 + y)^{-n}$$

The first part of the right-hand side of the above equation is the present value (at the time of bond purchase) of the individual interest payments and the second term is the present value of the final bond payout.

Remarks:

- The bond purchaser is sold the bond under the guarantee that the yield will be  $y$  and the purchase price  $P$  is computed under this assumption. The coupon rate is only used to compute the regular payout amount  $A$ .
- It is assumed that the yield rate applies on the same schedule as the coupon schedule (e.g., semi-annually or quarterly). If not, then the yield rate needs to be converted to an equivalent rate on the same schedule as the coupon rate using the conversion formula from Section 4.7.1.

The payout timeline for the type of bond described above is shown Figure 9.

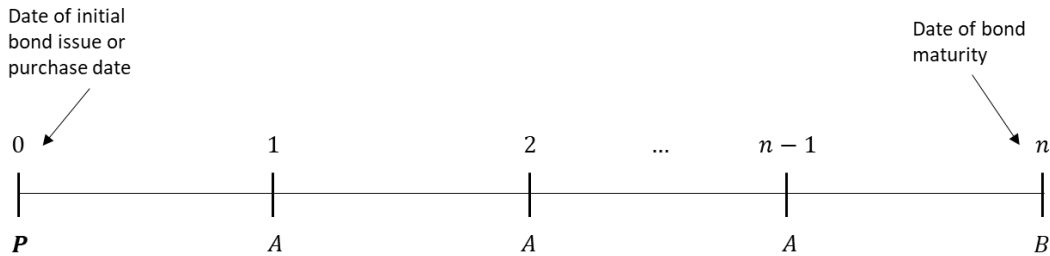


Figure 9. Payouts timeline for a bond

### 6.2.1 School Bond Example

Consider the scenario where a municipality wants to raise \$1,000,000 for a new school building. The head financial planner for the town wants to offer 5-year bonds at a semi-annual yield of 1.5%, with a coupon rate of 2% paid semi-annually and with a redemption value of \$5,000 (which is equal to the face value in this case). How many bonds does the municipality need to sell?

The coupon  $A$  is  $(.02)(5,000) = 100$ . We are given that  $B = 5,000$ ,  $y = 1.5\%$  and  $n = 10$ . Using the balloon loan calculator with these values yields the required issue price of the bond, i.e., \$5,230.55. Dividing the issue price of the bond into \$1,000,000, we get 191.18 and so the town needs to sell at least 192 bonds to fully cover the cost of their planned school building.

### 6.2.2 Determine Purchase Price of Bond to Generate Various Yields

A purchase price other than the bond face value will lead to a different yield from the coupon rate, as we saw in the example in the previous section. As a second example, consider a bond with the following characteristics:

- Redemption value (equal to face value in this example) of the bond is \$1,000
- The coupon rate is compounded semi-annually at a nominal annual rate of 6%
- Length of the bond is 5 years.

Determine the purchase prices that will give yields of 4%, 6% and 8% (compounded semi-annually in each case). In all three cases, the semi-annual coupon payout is  $A = \frac{.06}{2}(1,000) = 30$ ,  $j = .03$  and  $n = 10$ .

In the case that a 6% yield is desired,  $y = j = .03$  and the price should be set at the redemption value of \$1,000. As noted previously, when the redemption value and face value are equal, a bond is said to be purchased at par.

In the case a 4% yield, we have  $y = .02$ . It should be emphasized the  $A$  does not change. Using the balloon loan calculator, the required price to give the desired yield is \$1,089.83. Since the bond is being purchased for more than its redemption value, it is said to be purchased at a **premium**. Such

a bond would be listed as  $\frac{1,089.83}{1,000} \times 100 \cong \$108.99$  (it is typical to list a bond's price per \$100 of face value).

In the case of an 8% yield, we have  $y = .04$ . Using the balloon loan calculator, the required price to produce the desired yield is \$918.89. Since the bond is being purchased for less than its redemption value, it is said to be purchased at a **discount**. The price of this bond would typically be listed as \$91.89 (i.e., the price per \$100 of face value).

### 6.2.3 Bond Prices versus Yield

Extending the previous example to include semi-annual yields between 1% and 5%, in .5% increments, we get the following table relating yields to purchase price:

*Table 5. Bond Purchase Price versus Yield*

| Semi-annual Yield | Bond Price  | Purchased     |
|-------------------|-------------|---------------|
| 1%                | \$ 1,189.43 | at a premium  |
| 1.5%              | \$ 1,138.33 | at a premium  |
| 2%                | \$ 1,089.83 | at a premium  |
| 2.5%              | \$ 1,043.76 | at a premium  |
| $i = y = 3\%$     | \$ 1,000.00 | at par        |
| 3.5%              | \$ 958.42   | at a discount |
| 4%                | \$ 918.89   | at a discount |
| 4.5%              | \$ 881.31   | at a discount |
| 5%                | \$ 845.57   | at a discount |

In general, as the yield increases, the bond price drops and vice versa. For example, assume Abe purchases a bond with a 6% coupon rate (compounded semi-annually), a face value (with equal redemption value) of \$1,000 and a term of 5 years. Abe decides to sell his bond after 2.5 years. Excluding brokerage fees, for how much should Abe expected to sell his bond?

First, note that Abe is basically selling a 2.5 year bond at this point with a redemption value of \$1,000. We need to compare this to other bonds with a 2.5 year term. Assume that such bonds now (at the time Abe wants to sell) have an average coupon rate of 7% (semi-annual compounding). Thus, the price estimate should be based on a bond with  $y = .035$ ,  $A = (.03)(1,000) = 30$  and  $n = 5$ . The key here is the A remains at \$30 (the bond issuer does not change their conditions when the bond gets sold). Using the balloon loan calculator, an estimate for price of the resold bond is \$977.42. This is different from the 3.5% row in Table 5, because the calculations in Table 5 are based on  $n = 10$  periods.

### 6.3 Schedule for a Bond

For accounting and tax purposes, it is useful to have a schedule for the various payments and gains (or losses) related to a bond. Such a schedule is developed in this section.

If we use the annuity notation mentioned in Section 4.3.1 (i.e.,  $a_{n|i} = \frac{1-(1+i)^{-n}}{i}$ ), the price formula for a bond (from Section 6.2) can be written as

$$P = Aa_{n|y} + B(1 - ya_{n|y})$$

and with some rearrangement, we get

$$P - B = (A - yB)a_{n|y}$$

If  $P > B$  (i.e., purchase price exceeds redemption value), the bond is purchased at a premium. From the formula above, the amount of the premium is  $P - B$ . The above formula also provides another way to characterize a premium bond purchase, i.e., when  $A - yB > 0$ . In other words, if the coupon exceeds the periodic yield (i.e., effective interest paid on the redemption value of the bond for a period), then the bond is purchased at a premium. If the redemption value equals the face value, then  $A = jB$  (where  $j$  is the periodic coupon rate). Substituting  $A = jB$  into  $A - yB > 0$ , we get  $(j - y)B > 0$  which implies  $j - y > 0$  (i.e., a bond is purchased at a premium if the coupon rate is greater than the yield).

Similar, if  $P < B$  (equivalently,  $A - yB < 0$  or  $j - y < 0$ ), then the bond is purchased at a discount.

Of course, if a bond is purchased at its redemption value (assuming this is the same as the face value) then  $P = B$ ,  $A = yB$  and  $j = y$ .

We now return to the accounting/tax issue mentioned at the beginning of this subsection. If a bond is purchased at a premium, the redemption amount (not the purchase amount) is returned to the bond holder on the redemption date and in effect, the bond holder will incur a loss. Depending on the applicable tax laws for the bond holder, there are at least two cases:

- This loss can be applied against the bond holder's income at the time of redemption, i.e., for a given tax year.
- Alternately, the bond holder could allocate (i.e., distribute) the loss over each bond interest period (over several tax years with the exception of short term bonds that do not span more than one year).

### 6.3.1 Schedule for a Bond Bought at a Premium

Ms. Erdnussbutter purchased a \$100,000 bond (redemption and face value being the same) with a nominal annual coupon rate of 8% (with semi-annual coupon payments) and at a nominal annual yield of 6% (semi-annual compounding). The length of the bond is 10 years.

The coupon rate is  $j = .08/2 = .04$  per period (i.e., half-year), the yield rate is  $y = .06/2 = .03$  per period, and  $n = 20$ . Since the  $j > y$ , we can conclude the bond has been purchased at a premium. Using the bond schedule creator, we get the schedule shown in Table 6. The purchase price is shown as the book value in row 0, i.e., \$114,877.47.

Depending on her circumstances, Ms. Erdnussbutter can either declare a single loss of \$14,877.47 in the final year of the bond's term, or she could take a series of smaller losses in each of the years she holds the bond. The smaller losses are listed in the "Book Value Adjustment" column in Table 6. Note that the sum of the book value adjustments equals the total amount of loss, i.e., \$14,877.47. In either case, she needs to pay taxes on the coupon payments received each year.

For each period, the book value adjustment (loss in this case) is subtracted from the coupon payment and thus reduces the taxable income. For example, if the bond scheduling technique is used, the taxable amount on the bond from period 1 is \$3,446.32 rather than \$4,000.

The computations are as follows:

- (interest on book value for period  $k$ ) = (periodic yield) **times** (book value from period  $k - 1$ )
- (book value adjustment for period  $k$ ) = (coupon) **minus** (interest on book value for period  $k$ ).

*Table 6. Schedule for Bond Purchased at a Premium*

| Period (k)    | Coupon             | Interest on Book Value | Book Value Adjustment | Book Value   |
|---------------|--------------------|------------------------|-----------------------|--------------|
| 0             | -                  | -                      | -                     | \$114,877.47 |
| 1             | \$4,000.00         | \$3,446.32             | \$553.68              | \$114,323.80 |
| 2             | \$4,000.00         | \$3,429.71             | \$570.29              | \$113,753.51 |
| 3             | \$4,000.00         | \$3,412.61             | \$587.39              | \$113,166.12 |
| 4             | \$4,000.00         | \$3,394.98             | \$605.02              | \$112,561.10 |
| 5             | \$4,000.00         | \$3,376.83             | \$623.17              | \$111,937.94 |
| 6             | \$4,000.00         | \$3,358.14             | \$641.86              | \$111,296.07 |
| 7             | \$4,000.00         | \$3,338.88             | \$661.12              | \$110,634.96 |
| 8             | \$4,000.00         | \$3,319.05             | \$680.95              | \$109,954.00 |
| 9             | \$4,000.00         | \$3,298.62             | \$701.38              | \$109,252.62 |
| 10            | \$4,000.00         | \$3,277.58             | \$722.42              | \$108,530.20 |
| 11            | \$4,000.00         | \$3,255.91             | \$744.09              | \$107,786.11 |
| 12            | \$4,000.00         | \$3,233.58             | \$766.42              | \$107,019.69 |
| 13            | \$4,000.00         | \$3,210.59             | \$789.41              | \$106,230.28 |
| 14            | \$4,000.00         | \$3,186.91             | \$813.09              | \$105,417.19 |
| 15            | \$4,000.00         | \$3,162.52             | \$837.48              | \$104,579.71 |
| 16            | \$4,000.00         | \$3,137.39             | \$862.61              | \$103,717.10 |
| 17            | \$4,000.00         | \$3,111.51             | \$888.49              | \$102,828.61 |
| 18            | \$4,000.00         | \$3,084.86             | \$915.14              | \$101,913.47 |
| 19            | \$4,000.00         | \$3,057.40             | \$942.60              | \$100,970.87 |
| 20            | \$4,000.00         | \$3,029.13             | \$970.87              | \$100,000.00 |
| <b>Totals</b> | <b>\$80,000.00</b> | <b>\$65,122.53</b>     | <b>\$14,877.47</b>    | -            |

### 6.3.2 Schedule for a Bond Bought at a Discount

In this example, Mr. Everest buys a \$50,000 bond. The term of the bond is 5 years, the coupon rate is 4% (with payments semi-annually) and the yield is 5% (with semi-annual compounding).

The coupon rate is  $j = .04/2 = .02$ , the yield rate is  $y = .05/2 = .025$ , and  $n = 10$ . Since the  $j < y$ , we can conclude the bond has been purchased at a discount. Using the bond schedule creator, we get the schedule shown in Table 7. Note that the amounts in parentheses are meant to be negative (this is a convention in accounting).

Mr. Everest can choose to pay tax on the realized gain of \$2,188.02 in the final year of the bond's terms, or distribute the gains over the life of the loan (by the amounts shown in the book value adjustment column). In either case, he needs to pay taxes on the coupon payments received each year. The formulas for the calculations are the same as those used to construct Table 6.

**[Author's remark:** Showing the book value adjustments as negative in Table 7 seems odd to me since the amounts are being listed as profits on one's tax return. I have a similar comment on Table 6. I guess the idea is for the Interest on Book Value plus the Book Value Adjustment to equal the Coupon.]

*Table 7. Schedule for Bond Purchased at a Discount*

| Period        | Coupon             | Interest on Book Value | Book Value Adjustment | Book Value  |
|---------------|--------------------|------------------------|-----------------------|-------------|
| 0             | -                  | -                      | -                     | \$47,811.98 |
| 1             | \$1,000.00         | \$1,195.30             | (\$195.30)            | \$48,007.28 |
| 2             | \$1,000.00         | \$1,200.18             | (\$200.18)            | \$48,207.47 |
| 3             | \$1,000.00         | \$1,205.19             | (\$205.19)            | \$48,412.65 |
| 4             | \$1,000.00         | \$1,210.32             | (\$210.32)            | \$48,622.97 |
| 5             | \$1,000.00         | \$1,215.57             | (\$215.57)            | \$48,838.54 |
| 6             | \$1,000.00         | \$1,220.96             | (\$220.96)            | \$49,059.51 |
| 7             | \$1,000.00         | \$1,226.49             | (\$226.49)            | \$49,285.99 |
| 8             | \$1,000.00         | \$1,232.15             | (\$232.15)            | \$49,518.14 |
| 9             | \$1,000.00         | \$1,237.95             | (\$237.95)            | \$49,756.10 |
| 10            | \$1,000.00         | \$1,243.90             | (\$243.90)            | \$50,000.00 |
| <b>Totals</b> | <b>\$10,000.00</b> | <b>\$12,188.02</b>     | <b>(\$2,188.02)</b>   | -           |

## 6.4 Correction if Bond Not Purchased on a Coupon Date

The discussion in the previous sections has been under the assumption that the bonds were purchased on the date of a coupon payout (with the payout going to the bond seller) or at the time of the initial issue of the bond. However, it is far more likely that a bond is sold between coupon payout dates. In such cases, a correction is needed to the calculations.

### 6.4.1 Dirty Price of a Bond

Figure 10 depicts a situation where a bond is purchased between two coupon payout dates (or possibly between the issue date of the bond and the first coupon payout date). The bond purchase price at time 0 is  $P$ . The coupon payouts are in the amount  $A$ . At maturity, the bond holder receives the final coupon payout plus the redemption value of the bond  $B$ .  $P$  is the price of the bond if it had been purchased at time 0 with the given yield.

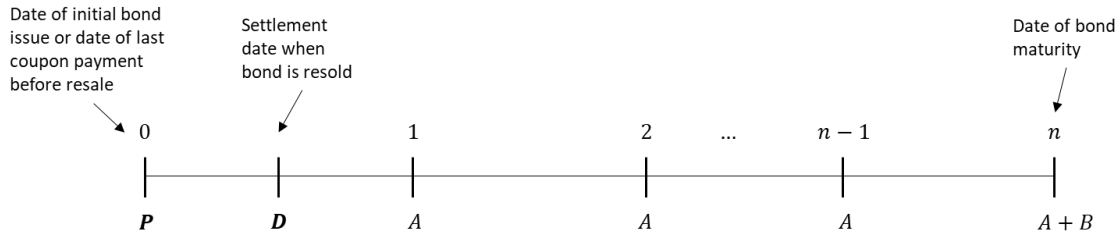


Figure 10. Bond Purchases between Coupon Dates

The price  $P$  of the bond at time 0 can be calculated with the bond price formula, i.e.,  $A \frac{1-(1+y)^{-n}}{y} + B(1+y)^{-n}$ , since this time coincides with the initial bond issue date or a coupon payment date. Using the concepts from Section 3.7, the equivalent value of  $P$  at the time of settlement can be computed (this is shown as  $D$  in the figure and in the calculations that follow). If simple interest (at rate  $y$ ) is assumed over the fraction of the period between time 0 and the settlement date (call this fraction  $k$ ), then  $D = P(1 + ky)$ . [Sometimes, perhaps usually, an alternate formula is used, i.e.,  $D = P(1 + y)^k$ .]  $D$  is known as the full or **dirty price** of a bond.

#### 6.4.2 Dirty Bond Price Example

As an example, consider a 30-year US Treasury bond with an annual yield of 1.61%, an annual coupon rate of 1.25% (interest paid semi-annually) and redemption value (same as face value in this case) of \$10,000. The bond was purchased by a brokerage house. At what price should the bond be sold to achieve the advertised yield if the settlement date is 5 June 2020 (i.e., when the bond is resold to a customer of the brokerage house) and the maturity day is 15 May 2050?

Based on the maturity date and condition of semi-annual coupon payouts, one can deduce that the coupon payouts are on May 15<sup>th</sup> and November 15<sup>th</sup> of each year. There are 21 days between May 15<sup>th</sup> and June 5<sup>th</sup>, and 184 days between May 15<sup>th</sup> and November 15<sup>th</sup>. So,  $k = \frac{21}{184} \cong .11413$ . The other input variables are as follows:

- yield rate per period =  $.0161/2 = .00805$
- total number of payments = 60 (note that no payments are missed as the broker resells the bond before the first coupon payout)
- redemption value = \$10,000
- coupon rate per period =  $.0125/2 = .00625$
- coupon payout per period =  $.00625 \times 10,000 = \$62.50$

Using the bond price formula, we get  $P = \$9,146.12$  (this can be checked using the balloon loan calculator as described previously). Finally, we get  $D = P(1 + ky) \cong \$9,154.52$ . Using the alternate formula, we get  $D = P(1 + y)^k = \$9,154.49$ .

If the bond was sold later in the coupon period (e.g., 10 November 2020) then  $k$  would be  $\frac{179}{184} \cong .972826$  and  $D \cong \$9,217.75$ . Using the alternate formula, we get  $D = P(1 + y)^k = \$9,217.74$ .

#### 6.4.3 Clean Price of a Bond

Before we get to the definition of the clean price for a bond, let's further analyze how the dirty price changes over time. For the example in the Section 6.4.2, Table 8 shows the dirty price for the

bond starting at time 0 and going through the end of period 4 (the other 55 periods are not shown in the table). Each period is divided into 8 equal time intervals. (There is nothing special about 8. The point is to get a few representative points for the graph in Figure 11.) For example, the dirty price of the bond at time  $2\frac{115}{184}$  is \$9,214.59 (shown in bold in the table). The column on the extreme right (Diff) shows the difference between the dirty bond price at the end of one interval and the beginning of the next (just before and after a coupon payout). For example, the difference between the price at the end of period 0 (just before the first coupon payout) and the beginning of period 1 (just after the first coupon payout) is \$9,219.75 - \$9,157.24 = \$62.51 (see the shaded cells table below). The difference should be exactly the coupon payout amount  $A$  but there is a small rounding error in the calculations.

The fact that the dirty price drops by exactly  $A$  at the end of a period can be proven as follows (noting that  $k$  is 1 just before the end of a period and 0 at the beginning of a period):

Take the dirty price of a bond just before the end of a period with  $m$  remaining coupon payments and subtract the dirty price of the bond at the beginning of the next period (at which point there are  $m - 1$  coupon payments remaining) to get the following (with terms in bold cancelling each other):

$$\begin{aligned}
 (1 + 1 \cdot y) \left[ A \frac{1 - (1 + y)^{-m}}{y} + B(1 + y)^{-m} \right] - (1 + 0 \cdot y) \left[ A \frac{1 - (1 + y)^{-(m-1)}}{y} + B(1 + y)^{-(m-1)} \right] \\
 = A \frac{(1 + y) - (1 + y)^{-(m-1)}}{y} + B(1 + y)^{-(m-1)} - A \frac{1 - (1 + y)^{-(m-1)}}{y} - B(1 + y)^{-(m-1)} \\
 = \frac{Ay}{y} + A \frac{1 - (1 + y)^{-(m-1)}}{y} - A \frac{1 - (1 + y)^{-(m-1)}}{y} = A
 \end{aligned}$$

Table 8. Dirty Price Changes over Time for a Bond

| Period | 0        | 23/184   | 46/184   | 69/184   | 92/184   | 115/184         | 138/184  | 161/184  | <1       | Diff  |
|--------|----------|----------|----------|----------|----------|-----------------|----------|----------|----------|-------|
| 0      | 9,146.12 | 9,155.32 | 9,164.53 | 9,173.73 | 9,182.93 | 9,192.14        | 9,201.34 | 9,210.54 | 9,219.75 |       |
| 1      | 9,157.24 | 9,166.45 | 9,175.67 | 9,184.88 | 9,194.10 | 9,203.31        | 9,212.53 | 9,221.74 | 9,230.96 | 62.51 |
| 2      | 9,168.46 | 9,177.69 | 9,186.91 | 9,196.14 | 9,205.36 | <b>9,214.59</b> | 9,223.81 | 9,233.04 | 9,242.27 | 62.50 |
| 3      | 9,179.76 | 9,189.00 | 9,198.23 | 9,207.47 | 9,216.71 | 9,225.95        | 9,235.18 | 9,244.42 | 9,253.66 | 62.51 |
| 4      | 9,191.16 | 9,200.41 | 9,209.66 | 9,218.91 | 9,228.15 | 9,237.40        | 9,246.65 | 9,255.90 | 9,265.15 | 62.50 |

The points in Table 8 are plotted on the graph shown in Figure 11. Each upward line represents one period (one row from the table). The drop-off from one period to another is actually discontinuous, i.e., the downward lines should not be in the figure (limitation of the plotting tool used).



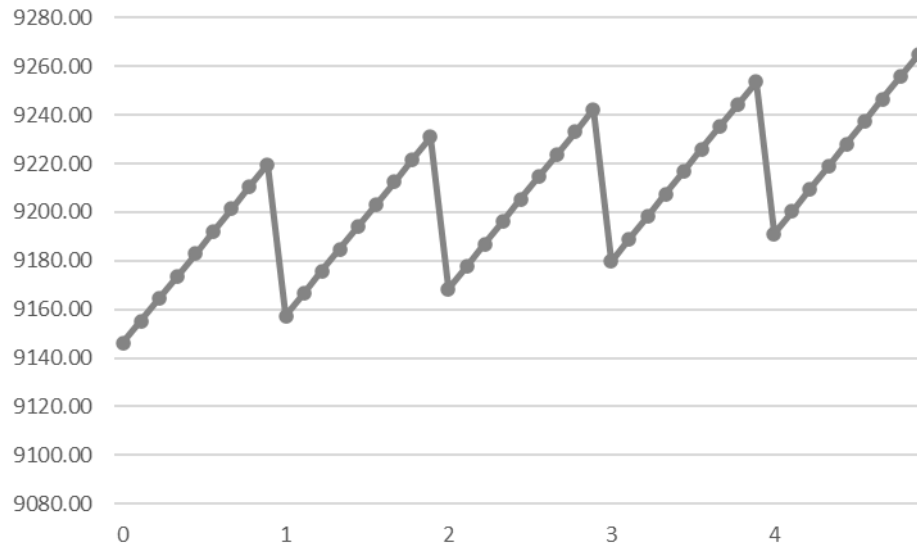


Figure 11. Graph of Dirty Bond Prices over Time

The reason for the increase in the price over a period is that the dirty price reflects the accrued interest on the coupon before it is paid out at the end of the period. Look at it this way, if the price did not reflect the accrued interest on the coupon, everyone would buy bonds just before the coupon dates and immediately get the coupon payment for the previous period.

The **clean price** of a bond (also known as the flat price) considers the accrued interest on the bond since its last coupon payment date. The clean price equals the present value of the bond's future cash flows (i.e., the dirty price) minus the interest earned on the bond between the last payment date and the transaction date. Most financial markets (particularly in the US) quote bonds using their clean prices. Accrued interest is added back to the quoted price to determine the purchase price. So, bonds are typically advertised at their clean price and sold at their dirty price.

In terms of equations, we have the following definitions:

- Accrued interest:  $I_k = kA$  where  $k$  is the fraction of a given coupon period that has transpired at settlement time (i.e., time of purchase).
- Clean price:  $C = D - I_k$

Table 9 shows the clean prices corresponding to the dirty prices in Table 8.

Table 9. Clean Price Changes over Time for a Bond

| Period | 0        | 23/184   | 46/184   | 69/184   | 92/184   | 115/184  | 138/184  | 161/184  | <1       |
|--------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0      | 9,146.12 | 9,147.51 | 9,148.90 | 9,150.29 | 9,151.68 | 9,153.07 | 9,154.46 | 9,155.86 | 9,157.25 |
| 1      | 9,157.24 | 9,158.64 | 9,160.04 | 9,161.45 | 9,162.85 | 9,164.25 | 9,165.65 | 9,167.05 | 9,168.46 |
| 2      | 9,168.46 | 9,169.87 | 9,171.29 | 9,172.70 | 9,174.11 | 9,175.53 | 9,176.94 | 9,178.35 | 9,179.77 |
| 3      | 9,179.76 | 9,181.18 | 9,182.61 | 9,184.03 | 9,185.46 | 9,186.88 | 9,188.31 | 9,189.73 | 9,191.16 |
| 4      | 9,191.16 | 9,192.60 | 9,194.03 | 9,195.47 | 9,196.90 | 9,198.34 | 9,199.78 | 9,201.21 | 9,202.65 |

The points in Table 9 are plotted on the graph shown in Figure 12. This would be a straight line but for some rounding errors in the computations.

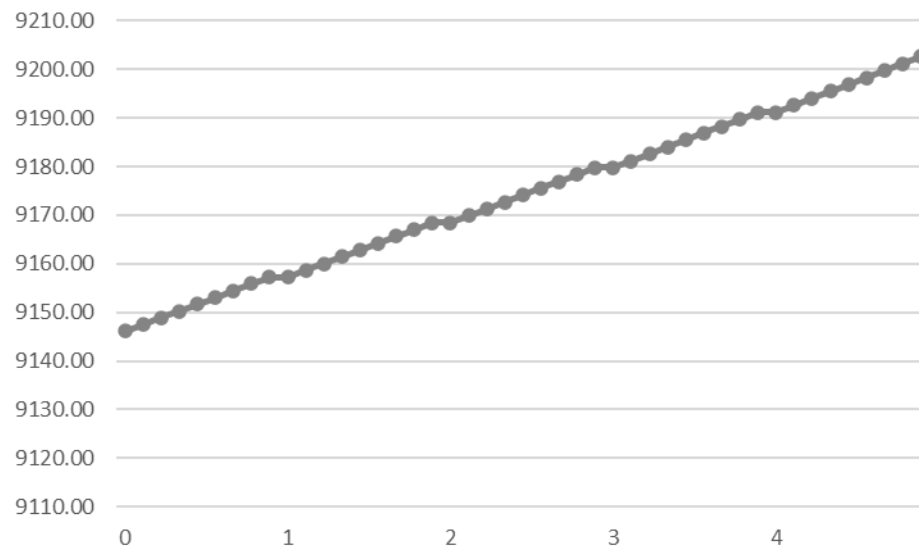


Figure 12. Graph of Clean Bond Prices over Time

## 6.5 Computing the Yield given the Bond Price

So far, all the work in this section assumes that the yield price is given (along with the coupon rate, redemption value, and term of the bond) and the bond price is derived. In fact, it is possible to determine the yield when given the bond price, coupon rate and term of the bond, but there is no closed formula for the yield in terms of the other characteristics of a bond. A numerical approximation method is required to solve the problem.

Fortunately, Microsoft Excel offers a yield function that computes the yield for a bond given:

- Settlement – the date when the security is sold to the buyer (this could be the issue date of the bond or a later date). Need to use the Excel DATE function to enter the date.
- Maturity – the date when the security expires. Again, one needs to use the Excel DATE function to enter the date.
- Coupon rate per year
- Purchase price per \$100 of face value.
- Redemption value per \$100 face value.
- Frequency – the number of coupon payments per year. For annual payments, frequency = 1; for semiannual, frequency = 2; for quarterly, frequency = 4. No other options are supported.
- Basis (optional) - the type of day count basis to use.

Consider the example from Section 6.3.2 concerning Mr. Everett but this time assume we are given the purchase price \$47,811.98 and asked to find the yield. In order to use the Excel function, we need actual dates that span (in this case) 5 years. We can arbitrarily choose January 1, 2000 as the settlement date and January 1, 2005 as the maturity date. The nominal annual coupon rate is 4%.

The purchase price per \$100 of face value is  $47,811.98 \times \frac{100}{50,000} = 95.62396$ . The redemption value is equal to the \$100 per \$100 of face value. The frequency is 2.

If the following is entered in an Excel cell, the result is 0.050000019 (not exact but pretty good):

$$= \text{YIELD}(\text{DATE}(2000,1,1), \text{DATE}(2005,1,1), 0.04, 95.62396, 100, 2)$$

## 6.6 Supporting Programs

### 6.6.1 Balloon Loan Calculator

Usage of the balloon loan calculator was described in Section 5.5.3.

As described in Section 6.2, a bond can be viewed as the balloon loan (see the mappings in Table 10). The differences are as follows:

- The yield of a bond is modified by changing the asking (purchase) price, while the loan interest rate is set by the entity offering the loan.
- The periodic payments on a balloon loan entail payment of interest and principal, while the coupon payments on a bond only entail interest payments.

*Table 10. Balloon Loan – Bond Comparison*

| Balloon Loan Concept   | Bond Concept  |
|--|---|
| Amount of loan ( $P$ )   | Purchase price of bond ( $P$ )  |
| Interest rate per period ( $i$ )   | Yield rate per period ( $y$ )   |
| Number of payments ( $n$ )   | Number of payments ( $n$ )  |
| -  | Coupon rate per period ( $j$ )  |
| Amount of each periodic payment ( $A$ ) – computed using the formula<br>$A = \frac{iP(1+i)^n - iB}{(1+i)^n - 1}$ | Amount of coupon payment – computed using the formula $A = jB$                    |
| Amount of balloon payment ( $B$ )  | Redemption value bond ( $B$ ) – assumed to be equal to the face value of the bond |

When using the balloon loan calculator for bonds, one enters the yield (in the “interest rate per period” field), total number of payments, redemption value (in the “amount of balloon payment” field) and the coupon amount (in the “amount of each periodic payment” field).

### 6.6.2 Bond Schedule Creator

File name: `bond_schedule_creator.py`

This program creates a schedule of payments for a bond.

- The input consists of the redemption value of the bond (assumed to be the same as the face value), the number of payment periods, the coupon rate per period (as a decimal) and the yield rate per period (as a decimal).
- The output shows the time period, coupon interest (which is a constant), interest on book value and book value adjustment for each period. The output is placed in a file named `bond_schedule.csv` (in the same directory where the program has been placed). It is recommended that the file be opened using a spreadsheet application such as Microsoft Excel.

The program works for bonds purchased at a premium, at a discount and at face value.

## 6.7 Exercises

1. Compute the price of a 10-year zero coupon bond that effectively yields 3.5% per year (under the assumption of yearly compounding). The redemption value of the bond is \$15,000. **Answer:** Use the formula for the bond price from Section 6.2 with  $A = 0$  (since there is no coupon) and  $B = 15,000$  to get the result  $P = 15,000(1 + .035)^{-10} = \$10,633.78$ . **Note:** If no compounding is assumed, then we need to compute the present value of 15000 under the assumption of simple interest, i.e.,  $P = \frac{15,000}{1+10(.035)} = \$11,111.11$ .
2. Compute the price of a 5-year bond with semi-annual coupon payments at a nominal annual rate of 5%. The redemption and face value are the same, i.e., \$50,000. The nominal annual yield rate is 4% (compounded semi-annually). **Hints:** Use the balloon loan calculator in the manner described in Sections 6.2 and 6.6.1. The coupon is  $\frac{.05}{2}(50,000) = 1,250$ . **Answer:** \$52,245.65
3. Is the bond from Exercise 2 purchased at a premium, at a discount or at par? **Answer:** The coupon rate is higher than the yield. So, the bond is purchased at a premium.
4. Using the information from Exercise 2, create a table that shows the bond price for the following nominal annual yields (again assuming semi-annual compounding): 2%, 3%, 4%, 5%, 6%, 7% and 8%. **Hints:** This is similar to the example in Section 6.2.3. Don't forget to convert the nominal annual yields to semi-annual yields. Also, keep in mind that the coupon is the same for all cases in this exercise. **Answer:**

Table 11. Answer to Exercise 6

| Semi-annual yield | Bond price  | Purchased     |
|-------------------|-------------|---------------|
| 1                 | \$57,103.48 | at a premium  |
| 1.5               | \$54,611.09 | at a premium  |
| 2                 | \$52,245.65 | at a premium  |
| 2.5               | \$50,000.00 | at par        |
| 3                 | \$47,867.45 | at a discount |
| 3.5               | \$45,841.70 | at a discount |
| 4                 | \$43,916.83 | at a discount |

5. For the bond described in Exercise 2, create bond schedules for two cases, i.e., when the yield is 3% and 7% (semi-annual compounding in both cases). **Hint:** Use the bond schedule creator.

6. Compute the dirty (full) price for a 20-year US Treasury bond that is to have an annual yield of 1% (compounded semi-annually), an annual coupon rate of 1.25% (interest paid semi-annually) and redemption value (same as face value in this case) of \$25,000. The settlement (purchase) date is 15 June 2020 and the redemption (maturity) date is 31 August 2030.  
**Hints and answer:** Clearly, the bond is being resold since the settlement date is far less than 20 prior to the redemption date. *The coupon dates are August 31<sup>st</sup> and March 3<sup>rd</sup> (note the issue with February having only 28 days) and so there are 21 coupon payments remaining, i.e.,  $n = 21$ . Use the balloon calculator to compute the price of the bond at the time of last coupon payment before the settlement date which is \$25,621.50 (this is  $P$  in the formula for the dirty price from Section 6.4.1). Use the online application at <https://www.timeanddate.com/date/duration.html> to count the number of days from 3 March 2020 to 15 June 2020 (105 days), and between 3 March 2020 to 31 August 2020 (182 day) which means that  $k = \frac{105}{182} \cong .5769$ . Applying the dirty price formula, we get  $D = P(1 + ky) \cong 25,621.50(1 + (.5769)(.005)) \cong \$25,695.41$ . If we use the alternate formula for the dirty price, we get  $D = P(1 + y)^k \cong \$25,695.33$ .*
7. Using the simple interest version of the dirty price formula, show that the clean price can be represented by the formula  $C = P + k(yP - iB)$  with  $P, k, y, i$  and  $B$  as defined in Section 6.4.1. Hint: Assume that the face value is redemption value and thus  $A = iB$ .
8. For the bond described in Exercise 6, compute the clean price. **Answer:**  $C = D - kA \cong 25,695.41 - .5769(156.25) \cong \$25,605.27$ .

## 7 Capital Budgeting

This section covers some financial concepts that help in decision making with regard to business investments and the incremental cash flows that results from such investments. The reader is assumed to be familiar with cash flow and how it is different from profit. For a refresher on cash flow and profit, see the YouTube video “What is a cash flow statement?” [8] and the Investopedia article on cash flow [9]. For a more extensive discussion of cash flow analysis and capital budgeting, see Chapters 12 and 13 of The Basics of Finance [6].

**[Author’s Remark:** While I have not provided any Python programs in support of this section, there is an Excel spreadsheet that covers many of the computations related to the examples. It is recommended that you have this spreadsheet in hand when reading this section. For a summary of the various sheets within the spreadsheet, see Section 7.6.]

### 7.1 Net Present Value (NPV)

#### 7.1.1 Concept with an Example

**Net Present Value (NPV)** refers to the present value of all cash flows related to a given project and associated investments. NPV is used as a method for making decisions concerning capital budgeting. If the NPV for a project is negative, then it is recommended not to proceed with the project. If the NPV for a project is positive, then it is recommended to proceed with the project. The NPV can also be used to compare several projects (see the example in Section 7.1.6). As we shall see, NPV is but one measure used to decide whether to go ahead with a project or not.

Calculation of the NPV depends on the stipulation of a given rate of return (known as the cost of capital). The general idea is that a proposed capital investment project should provide a higher rate of return than some other (typically external) investment of similar risk. For example, if a company can make a 5% rate of return with, in their opinion, a moderate risk investment in the bond market, then an internal project, with similar risk, would be expected to make more than 5% per year.

Wikipedia provides the following definition of **Cost of Capital** [7]:

In economics and accounting, the cost of capital is the cost of a company's funds (both debt and equity), or, from an investor's point of view "the required rate of return on a portfolio *[of a]* company's existing securities". It is used to evaluate new projects of a company. It is the minimum return that investors expect for providing capital to the company, thus setting a benchmark that a new project has to meet.

As an example, consider a small business (Acme Furniture) that makes custom furniture and does antique furniture restoration. The business is currently run by Alice and Bob in a converted garage. On a regular basis, Alice and Bob are turning down business because of the lack of resources. To address this issue, they are considering an expansion of the garage, the purchase of additional equipment, the hiring of additional staff, additional parking spaces and a small showroom for customers. The upgrade to their business will require an investment of \$225,000. On the other hand, Alice and Bob could invest the \$225,000 in three other ways, with the following semi-annual rate of returns

- 2% (conservative investment, e.g., bank certificate of deposits)
- 4% (moderate risk, e.g., government and corporate bonds)
- 6% (higher risk, e.g., stocks and higher risk bonds).

Alice and Bob know their business well and are able to make (in their opinion) reasonably accurate projections of cash flows for the business expansion (as a result of the planned expansion). The problem at hand is to compare the projected cash flows against the three alternative investment options.

Table 12 shows Alice and Bob's projected cash flows concerning Acme Furniture after the planned expansion.

- The top row is time in half-year increments, up to the end of year number 5.
- The second row contains the cash inflows to Acme Furniture, e.g., sales of furniture.
- The third row contains the cash outflow, e.g., the initial investment, supplies, wages.
- The bottom row shows net cash flows for each 6-month period. In this example, there is only a net outflow at time 0. However, in general, any of the periods could be a net inflow or outflow of cash.

(Note: In the examples that follow, we usually only show the net cash flow for each period and omit the details concerning cash inflows and outflows.)

*Table 12. Initial Investment and Cash Flow in Half-year Increments*

| Year          | 0         | .5       | 1        | 1.5      | 2        | 2.5     |
|---------------|-----------|----------|----------|----------|----------|---------|
| Cash Inflow   |           | 30,000   | 35,000   | 35,000   | 40,000   | 35,000  |
| Cash Outflow  | (225,000) | (10,000) | (13,000) | (11,000) | (14,000) | (7,000) |
| Net Cash Flow | (225,000) | 20,000   | 22,000   | 24,000   | 26,000   | 28,000  |

| Year          | 3        | 3.5      | 4        | 4.5      | 5        |
|---------------|----------|----------|----------|----------|----------|
| Cash Inflow   | 40,000   | 45,000   | 50,000   | 50,000   | 50,000   |
| Cash Outflow  | (10,000) | (13,000) | (16,000) | (14,000) | (12,000) |
| Net Cash Flow | 30,000   | 32,000   | 34,000   | 36,000   | 38,000   |

Recall from Section 3.7 that we can compare the value of money realized at different times by computing the various values at some common focal date. For the problem at hand, we will use the time of the initial investment (i.e., time 0) as the focal date and compute the value of the various net cash flows for each period based on their value at the focal date. In general, we make the following assignment

- let  $C_0$  be the initial investment
- let  $C_k$  be the net cash flow for period  $k$  which ends at time  $t_k$  (measured in time periods, e.g., half-years, quarters or months)
- $i$  is the interest rate per period, i.e., the cost of capital
- $n$  is the number of cash flows.

With the above assignments, the general formula for net present value is given by

$$NPV = -C_0 + \sum_{k=1}^n C_k(1+i)^{-t_k}$$



The formula allows for the cash flows to be unequally spaced in terms of the number of periods. For example, we could have cash flow estimates at periods number 3, 5, 11 and 13. In this case,  $n = 4$ ,  $t_1 = 3$ ,  $t_2 = 5$ ,  $t_3 = 11$  and  $t_4 = 13$ . However, it is more usual to have a net cash flow estimate for each period, in which case,  $t_k = k$  and the formula becomes

$$NPV = -C_0 + \sum_{k=1}^n C_k(1+i)^{-k}$$

Getting back to the problem at hand, we can use the Microsoft Excel function NPV to compute the NPV for each of the 3 cost of capital scenarios, i.e., 2%, 4% and 6% semi-annual rates. The Excel NPV function is easy to use. The first parameter is the rate of return and the following parameters are the net cash flows for each period (see <https://support.microsoft.com/en-us/office/npv-function-8672cb67-2576-4d07-b67b-ac28acf2a568> for further details). The results are summarized in Table 13 (the parenthesis indicate a negative NPV). For 2% and 4%, Acme Furniture will have a positive NPV and thus it makes sense to go ahead with the expansion in these two cases. However, at 5%, the NPV is negative and it may not be a good idea to go forward with the expansion in this case.

Table 13. Investment Analysis for Acme Furniture

| Semi-annual Cost of Capital | NPV           |
|-----------------------------|---------------|
| 2%                          | \$31,923.43   |
| 4%                          | \$4,789.06    |
| 5%                          | (\$6,915.43)  |
| 6%                          | (\$17,541.15) |

For the sake of argument, let's say that Alice and Bob think that the risk in expanding their business is similar in risk to the 6% cost of capital alternative. **[This is a key point, i.e., the cost of capital needs to be at the same estimated risk as the proposed investment!]** With the given information, they should not go forward with their business expansion. However, it may be that consideration of a longer time frame for the cash flows will change the decision. For example, assume Alice and Bob are able to project the cash flow for a 6<sup>th</sup> year, as shown in Table 14.

Table 14. Initial Investment, and Cash Flow in Half-year Increments - Revised

|           |        |        |        |        |        |        |        |        |        |        |        |        |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0         | .5     | 1      | 1.5    | 2      | 2.5    | 3      | 3.5    | 4      | 4.5    | 5      | 5.5    | 6      |
| (225,000) | 20,000 | 22,000 | 24,000 | 26,000 | 28,000 | 30,000 | 32,000 | 34,000 | 36,000 | 38,000 | 41,000 | 44,000 |

The two extra semi-annual cash flow projections change the investment analysis, as shown in Table 15. With the extra year of cash flow taken into consideration, it now makes sense to go forward with the business expansion, even for the case of a 7% return on the alternative investment. It is not until about 8% where the alternative investment makes more sense.

*Table 15. Investment Analysis for Acme Furniture - Revised*

| Semi-annual Cost of Capital | NPV          |
|-----------------------------|--------------|
| 2%                          | \$98,265.08  |
| 4%                          | \$56,822.80  |
| 6%                          | \$23,463.51  |
| 7%                          | \$9,269.22   |
| 8%                          | (\$3,506.95) |

### 7.1.2 Negative Cash Flows

The previous example showed positive cash flows for each period but this is not always the case, and it is quite possible to have negative net cash flows for some periods. For example, the capital expenditure could be negative in some periods because of start-up issues, seasonal businesses, or expected charges (e.g., insurance payments) at various points in time.

As an example, consider a vacation resort that has a seasonal business, with Fall being a time for repairs and a reduced number of customers. The resort owners are considering a \$500,000 investment to improve the resort. Table 16 shows the estimated project cash flows for each season (quarter) in the 5 years following the potential investment.

*Table 16. Project Quarterly Cash Flows for a Resort Business*

| Year | Winter | Spring | Summer | Fall   |
|------|--------|--------|--------|--------|
| 1    | 40,000 | 41,000 | 42,000 | (3000) |
| 2    | 41,000 | 42,000 | 43,000 | (2000) |
| 3    | 41,000 | 42,000 | 43,000 | (2000) |
| 4    | 43,000 | 44,000 | 45,000 | (1500) |
| 5    | 43,000 | 44,000 | 45,000 | (1500) |

Table 17 shows an analysis of the potential investment at various costs of capital (keeping in mind that the cost of capital refers to alternative investments). The project's cash flows outweigh the alternative investments up to and including a quarterly rate of 2.25%. At 2.5%, it **may** make more sense not to invest in the resort but rather in the alternative investment. On the other hand, the 2.5% alternative investment may be very risky (perhaps junk bonds and small growth stocks) and the resort improvement investment may be seen as much more certain than the 2.5% alternative investment. The point is that the final decision by the resort owners is not black and white, and the NPV analysis is just one tool to help with their decision.

*Table 17. Investment Analysis for Resort Investment*

| Quarterly Cost of Capital | NPV        |
|---------------------------|------------|
| .5%                       | \$97,678   |
| .75%                      | \$82,931   |
| 1%                        | \$68,758   |
| 1.25%                     | \$55,134   |
| 1.5%                      | \$42,037   |
| 1.75%                     | \$29,444   |
| 2%                        | \$17,334   |
| 2.25%                     | \$5,685    |
| 2.5%                      | \$(5,520)  |
| 2.75%                     | \$(16,300) |

### 7.1.3 Multiple Investments

The examples thus far have only had one investment and that occurred at the beginning of the project in question. This does not always need to be the case and in fact, it is quite possible that a project will require several investments at various times. In the case of several investments, one can separate the cash flow into what are called operating cash flow and investment cash flow (these are not to be confused with cash inflow and outflow). These two types of cash flow are added together for each period to get a net cash flow. In general, investment cash flows are negative. Operating cash flows can be positive or negative (as we saw in example from the previous section).

Table 18 shows a project that requires an initial investment of \$70,000 (e.g., purchase of some equipment) and another investment of \$10,000 in 4 year of the project (e.g., maintenance on the equipment or perhaps purchase of another piece of equipment).

*Table 18. Operating and Investment Cash Flows*

| Time in Years        | 0          | 1        | 2         | 3         | 4          | 5         | 6         | 7         |
|----------------------|------------|----------|-----------|-----------|------------|-----------|-----------|-----------|
| Operating Cash Flow  |            | \$ 7,500 | \$ 10,000 | \$ 12,500 | \$ 15,000  | \$ 15,000 | \$ 17,500 | \$ 20,000 |
| Investment Cash Flow | \$(70,000) |          |           |           | \$(10,000) |           |           |           |
| Net Cash Flow        | \$(70,000) | \$ 7,500 | \$ 10,000 | \$ 12,500 | \$ 5,000   | \$ 15,000 | \$ 17,500 | \$ 20,000 |

The operating and investment cash flows are added to get a net cash flow for each period, and we proceed as before to compute the NPV for various rates of return, as shown in Table 19. The investments in the project make financial sense up a little over a 5% rate for the cost of capital.

*Table 19. NPV Analysis in the Case of Multiple Investments*

| Yearly Cost of Capital | NPV       |
|------------------------|-----------|
| 0.02                   | \$9,705   |
| 0.03                   | \$6,258   |
| 0.04                   | \$3,078   |
| 0.05                   | \$143     |
| 0.06                   | (\$2,568) |
| 0.07                   | (\$5,072) |

#### 7.1.4 Range of Cash Flow Estimates

In the previous examples, it was assumed that the decision making parties (regarding capital expenditure) had a good idea of potential cash flows as a result of their investment. Another approach (particularly when the decision makers are not so sure of the potential cash flows) is to stipulate a range of values. For example, consider the case of Mr. Throckmorton who is starting a new restaurant. Unfortunately, Mr. Throckmorton has only a rough idea of cash flow based on market analysis and consequently, he can only stipulate a range of projected cash flows for each period. It is clear, however, that the restaurant will require an initial investment of \$625,000. Table 20 shows the yearly (projected) cash flows for the restaurant under pessimistic and optimistic scenarios.

*Table 20. Optimistic-Pessimistic Cash Flows for New Restaurant*

| Year        | 0         | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8       | 9       | 10      |
|-------------|-----------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|
| Pessimistic | (625,000) | 50,000 | 55,000 | 60,000 | 65,000 | 70,000 | 75,000 | 80,000 | 85,000  | 90,000  | 95,000  |
| Optimistic  | (625,000) | 60,000 | 66,000 | 72,000 | 78,000 | 84,000 | 90,000 | 96,000 | 102,000 | 108,000 | 114,000 |

Table 21 lists the NPV results for the optimistic and pessimistic cases. For the pessimistic case, the net cash flow is only positive up to 2.5%. The optimistic case has a positive cash flow up to 5.5%. Mr. Throckmorton could, for example, make his decision based on a compromise between the optimistic and pessimistic scenarios, and use 4% as the cut-off point, i.e., if he can get more than 4% with an alternative investment of similar risk to the restaurant business then he should **not** go ahead with the restaurant business.

*Table 21. NPV for Optimistic and Pessimistic Scenarios*

| Cost of Capital per Year | Pessimistic    | Optimistic    |
|--------------------------|----------------|---------------|
| 0.02                     | \$18,534.07    | \$144,789.90  |
| 0.025                    | \$604.08       | \$122,676.11  |
| 0.03                     | (\$16,452.34)  | \$101,616.42  |
| 0.035                    | (\$32,680.60)  | \$81,556.23   |
| 0.04                     | (\$48,123.51)  | \$62,444.10   |
| 0.045                    | (\$62,821.42)  | \$44,231.52   |
| 0.05                     | (\$76,812.39)  | \$26,872.75   |
| 0.055                    | (\$90,132.34)  | \$10,324.61   |
| 0.06                     | (\$102,815.13) | (\$5,453.63)  |
| 0.065                    | (\$114,892.78) | (\$20,500.44) |

### 7.1.5 Initial Investment is Loaned

All the previous examples in this section had the tacit assumption that the parties involved had the money required for the initial investment. It is likely, however, that some or all of the money may need to be borrowed. This clearly affects the NPV analysis since the loan repayments need to be subtracted from the cash flow estimates.

Consider the Acme Furniture example in the case where Alice and Bob need to take out a 5-year small business loan in the amount of \$50,000 at a nominal interest rate of 3% (repaid semi-annually). Under this scenario, Alice and Bob would need to make payments of \$5,069 twice per year for five years. These payments need to be subtracted from the cash flow estimates as shown in Table 22.

Notes:

- In some locales, interest payments on debt are tax deductible. This would reduce the impact on cash flow, but we didn't consider this issue in our analysis of the project.
- While investment is entirely within the cash outflow category, operating cash flow entails both cash inflow and outflow.

*Table 22. Acme Furniture - Cash Flow Estimates with Loan Repayments*

| Time in years               | 0         | .5      | 1       | 1.5     | 2       | 2.5     | 3       | 3.5     |
|-----------------------------|-----------|---------|---------|---------|---------|---------|---------|---------|
| <b>Operating Cash Flow</b>  |           | 20,000  | 22,000  | 24,000  | 26,000  | 28,000  | 30,000  | 32,000  |
| <b>Investment Cash Flow</b> | (225,000) | (5,069) | (5,069) | (5,069) | (5,069) | (5,069) | (5,069) | (5,069) |
| <b>Net Cash Flow</b>        | (225,000) | 14,931  | 16,931  | 18,931  | 20,931  | 22,931  | 24,931  | 26,931  |

| 3.5     | 4       | 4.5     | 5       | 5.5    | 6      |
|---------|---------|---------|---------|--------|--------|
| 32,000  | 34,000  | 36,000  | 38,000  | 41,000 | 44,000 |
| (5,069) | (5,069) | (5,069) | (5,069) | 0      | 0      |
| 26,931  | 28,931  | 30,931  | 32,931  | 41,000 | 44,000 |

Table 23 shows the NPVs for various rates of return for alternative investments. In this scenario, there is positive NPV up to and including 5%. Recall from the associated scenario without the loan that it still made business sense to go ahead with the improvements to Acme Furniture up to something over 7% but less than 8%. This is to be expected since the loan decreases net cash inflow for each period.

*Table 23. Investment Analysis for Acme Furniture with Loan Repayments*

| Semi-annual<br>Cost of Capital | NPV           |
|--------------------------------|---------------|
| 2%                             | \$53,625.16   |
| 3%                             | \$34,429.15   |
| 4%                             | \$17,289.98   |
| 5%                             | \$1,971.44    |
| 6%                             | (\$11,732.98) |

In the above example, the length of the loan and the project were the same. If the term of the loan was

longer than the project, then the results of the NPV analysis would not accurately reflect the value of the proposed capital expenditure since one is not considering the effects of the loan on cash flow after the end of the project. To address this type of problem one could extend the project length to coincide with the length of the loan but that could involve making long-term (and likely inaccurate) cash flow projections.

### 7.1.6 Comparison of Projects

It is possible to use NPV analysis to decide where to make an investment among several projects of similar risk. Table 24 shows projected yearly cash flows for Projects A and B, based on an initial investment of \$155,000.

Table 24. Cash Flows for Projects A and B based on Same Initial Investment

| Time in Years | 0         | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      |
|---------------|-----------|--------|--------|--------|--------|--------|--------|--------|--------|
| Project A     | (155,000) | 20,000 | 20,000 | 30,000 | 40,000 | 30,000 | 30,000 | 25,000 | 5,000  |
| Project B     | (155,000) | 50,000 | 10,000 | 20,000 | 20,000 | 20,000 | 30,000 | 25,000 | 25,000 |

Table 25 shows the computed NPVs for Projects A and B at various costs of capital. For lower rates (from .02 to .035), Project A has a higher NPV and would thus be a slightly better choice for investment. For rates between .04 and .06, Project B has a higher NPV and would be a better choice in this case. For rates higher than .06, the NPV is negative for both projects. In this case, it is better to put the capital into the alternative investment.

Table 25. NPV Comparison for Projects A and B

| Cost of Capital | NPV Project A      | NPV Project B      |
|-----------------|--------------------|--------------------|
| 0.02            | <b>\$28,330.62</b> | \$28,244.78        |
| 0.025           | <b>\$24,550.62</b> | \$24,474.07        |
| 0.03            | <b>\$20,912.86</b> | \$20,855.77        |
| 0.035           | <b>\$17,411.50</b> | \$17,383.12        |
| 0.04            | \$14,040.96        | <b>\$14,049.69</b> |
| 0.045           | \$10,795.93        | <b>\$10,849.36</b> |
| 0.05            | \$7,671.35         | <b>\$7,776.33</b>  |
| 0.055           | \$4,662.36         | <b>\$4,825.07</b>  |
| 0.06            | \$1,764.37         | <b>\$1,990.33</b>  |
| 0.065           | (\$1,027.06)       | (\$732.89)         |
| 0.07            | (\$3,716.12)       | (\$3,349.36)       |

## 7.2 Internal Rate of Return (IRR)

In the examples from the previous section, we evaluated the NPV of a series of cash flows against a rate of return for an alternative (and assumed to be external) investment such as bank notes, stocks or bonds. However, we did not pinpoint the exact rate of return where the NPV is 0, i.e., the point where there is an exact balance between the internal investment and the external alternative (i.e., the cost of capital). This balance point is known as the **internal rate of return**. From Wikipedia [11] we have the following definition:

The internal rate of return (IRR) is a measure of an investment's expected future rate of return. As the IRR is an estimate of a future annual rate of return, IRR should not be confused with the actual achieved investment return of an historical investment. The term internal refers to the fact that the calculation excludes external factors, such as the risk-free rate, inflation, the cost of capital, or various financial risks.

Technically, to find the IRR, one solves the equation for  $i$ :

$$NPV = -C_0 + \sum_{k=1}^n C_k(1+i)^{-k} = 0$$

which is equivalent to

$$C_0 = \sum_{k=1}^n C_k(1+i)^{-k}$$

In words, we are looking for the value (or values) of  $i$  such that the initial investment equals the present value of the subsequent cash flows resulting from the initial investment.

The decision rules concerning the IRR are as follows:

- If the IRR is greater than the cost of capital, go forward with the project.
- If the IRR is less than the cost of capital, do not go forward with the project.
- In the unlikely event that the IRR exactly equals the cost of capital, the analysis is indifferent as to whether the project should go forward or not.

### 7.2.1 Acme Furniture Example Revisited

Consider Table 14 regarding the Acme Furniture example from Section 7.1.1. From the analysis in Table 15, we see that the IRR is somewhere between 7% and 8%. Using the Microsoft Excel IRR function with the information from Table 14, we get the more precise value of 7.71% for the IRR. The Excel IRR function is very easy to use (just input the initial investment and the subsequent cash flows). For more details, see the description at <https://support.microsoft.com/en-us/office/irr-function-64925eaa-9988-495b-b290-3ad0c163c1bc>.

So, if the cost of capital is less than 7.71%, the Acme Furniture should invest in the project. For cost of capital rates higher than 7.71%, the NPV for the project becomes negative.

### 7.2.2 Multiple IRR Values for a Given Cash Flow

It is possible for a given cash flow to have more than one IRR, i.e., for the NPV equation to have several values of  $i$  that give an NPV value of 0. For example, the cash flow in Table 26 has two IRRs.

*Table 26. Cash Flow with Two IRRs*

| Time                 | 0           | 1         | 2         | 3         | 4         | 5           |
|----------------------|-------------|-----------|-----------|-----------|-----------|-------------|
| <b>Net Cash Flow</b> | \$ (10,000) | \$ 10,000 | \$ 11,000 | \$ 13,000 | \$ 10,000 | \$ (37,000) |

The existence of two IRRs can be seen by graphing the equation for the NPV as a function of  $i$  (divided by 1000 to simplify the equation), i.e.,

$$-10 + 10(1+i)^{-1} + 11(1+i)^{-2} + 13(1+i)^{-3} + 10(1+i)^{-4} - 37(1+i)^{-5}$$

The graph of the NPV as a function of  $i$  is shown in Figure 13. In the graph, the y-axis is the NPV and the x-axis represent the cost of capital  $i$ . As one can see, the graph crosses the x-axis around .05 and then again around .83. The Microsoft Excel IRR function has an optional parameter called “guess” that allows one to start the numerical algorithm for root determination around a given value. This is exactly what is needed in the case of several IRRs. If we use a guess of .05, the IRR



function in Excel returns the more precise value of .0507, and if we guess .8, the Excel IRR function returns the value of .8243. Thus, the two IRRs are 5.07% and 82.43%. Between these two values, the NPV is positive and it makes sense to go ahead with the initial investment of \$10,000. For values of  $i$  less than 5.07% or greater than 82.43%, the NPV is negative and so, the initial investment is not advised in these cases (at least not based on IRR analysis).

The possibility of multiple IRRs is one of the shortcomings with this approach to capital budgeting. As we shall see in Section 7.3, there is a modification to the IRR concept that eliminates the possibility of multiple solutions to the equation  $NPV = 0$ .

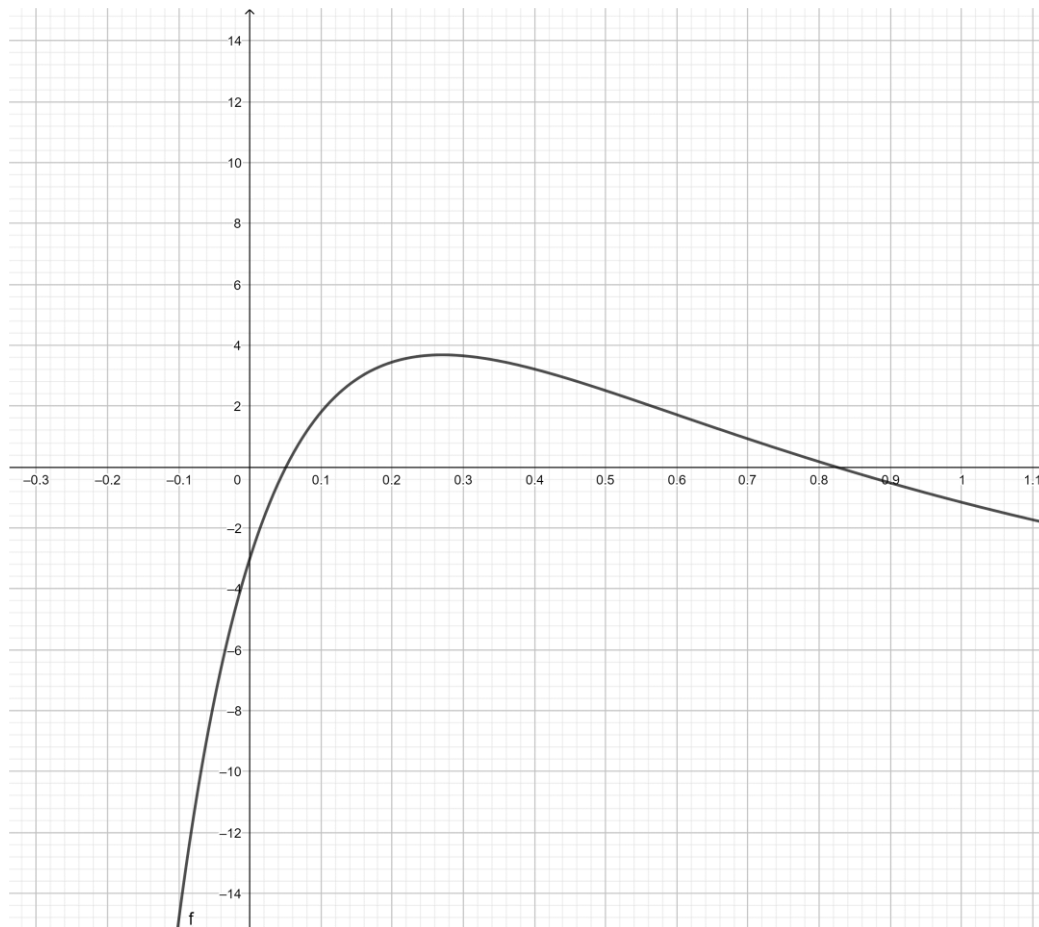


Figure 13. Graph of NPV as a Function of the Rate of Return

### 7.3 Modified IRR

When comparing mutually exclusive projects, the project with the highest IRR may not be the one with the highest NPV. The IRR approach can lead to a different decision than the NPV approach when evaluating mutually exclusive projects because of the underlying assumptions in these approaches:

- The NPV approach assumes that net cash flows are reinvested at the cost of capital.

- The IRR approach assumes that net cash flows are reinvested at the internal rate of return. This can lead to extreme overestimates of the future value of a project. Issues with the IRR approach are well explained in an article from McKinsey & Company [12].

**[Author's remark:** The above assumptions are common in finance books and articles, but have nothing to do with the actual computation of the NPV or IRR. It is the decision of the party involved concerning what interest rate they use to compute the reinvestment value of the net cash flows. In fact, the net cash flows could be spent in a subsequent period and not reinvested at all.]

The **Modified Internal Rate of Return** (MIRR) is a measure of an investment's attractiveness. It is used in capital budgeting to rank alternative investments of equal size. The MIRR is a modification of the IRR concept and as such aims to resolve some problems with the IRR.

The MIRR is calculated as follows for a given investment:

- Determine the future value of all periods with positive cash flows using the cost of reinvestment (call it  $i$ ). The calculation is done from the time of the inflow until the end of the project. The compute value is referred to as the Terminal Value (TV) of the cash flows. In the equations that follow, we call this value  $x$ .
- Determine the present value (at time zero) of all the periods with negative cash flows (including the initial investment), using the cost of financing (call it  $j$ ). In the equations that follows, we call this value  $y$ .
- Determine the MIRR, i.e., the rate of return at which the value  $y$  accumulates to the value  $x$  by the end of the project. Assume the project transpires over  $n$  time periods. In the equations that follows, the MIRR is represented by the variable  $z$ .

More formally, we have the following equation

$$-y(1+z)^n = x$$

which can be solved for  $z$  as follows

$$(1+z)^n = -\frac{x}{y}$$

$$z = \sqrt[n]{-\left(\frac{x}{y}\right)} - 1$$

where

$$x = \sum_{k=1}^u C_k (1+i)^{n-t_k}, y = \sum_{h=1}^v C_h (1+j)^{-t_h} \text{ and } C_m \text{ is the cash flow for period } t_m.$$

The MIRR is used in a manner similar to the IRR. If the MIRR for a project is greater than the cost of capital, then go forward with the project; otherwise, do not go forward with the project. When comparing several projects, the project with the highest MIRR is preferred. Further, the MIRR can be used to rank projects of different sizes.

Unlike the IRR, there is only one MIRR for a given project. Further, and as noted in the definitions above, the MIRR computation allows one to use a reinvestment rate of return  $i$  that is different from the MIRR itself.

As an example, consider the investment and cash flow data for a potential project in Table 27.

*Table 27. Input for MIRR Example*

| Year                   | 0        | 1       | 2     | 3      | 4      | 5      |
|------------------------|----------|---------|-------|--------|--------|--------|
| Net Cash Flow for year | (25,000) | (5,000) | 5,000 | 20,000 | 20,000 | 20,000 |

Assume that the cash inflows in years 2, 3 and 4 are reinvested at rate  $i = .04$ . The cash inflow in year 5 is at the end of the project and so, there is no reinvestment (at least not during the time duration of the project). We now have sufficient information to calculate  $x$ .

$$x = 5,000(1.04)^3 + 20,000(1.04)^2 + 20,000(1.04)^1 + 20,000 = 68,056.32$$

Next, we compute the net cash outflows at  $t = 0$ , assuming a financing rate  $j = .05$

$$y = -(25,000 + 5,000(1.05)^{-1}) = -29,761.90476$$

The MIRR is

$$\sqrt[5]{-\left(\frac{68,056.32}{-29,761.90476}\right)} - 1 \cong 0.1799$$

This can be cross checked with the Microsoft Excel MIRR function, which does in fact return the result of 17.99%. A description of the Excel MIRR function can be found at <https://support.microsoft.com/en-us/office/mirr-function-b020f038-7492-4fb4-93c1-35c345b53524>.

Using the Excel IRR function, we find that the IRR for this problem is 23.99%.

For an additional example, see the Wikipedia article on MIRR [13].

## 7.4 Payback and Discounted Payback Period

**Payback Period** and **Discounted Payback Period** (DPP) are decision techniques used to evaluate investment projects. The payback period approach entails a calculation of the time duration required to recover the cost of an initial investment. The shorter the time, the better the investment. The discounted payback period approach is similar except that one uses the discounted values of the cash flows.

From the Investopedia article on Payback Period [14]:

The payback period refers to the amount of time it takes to recover the cost of an investment. Simply put, the payback period is the length of time an investment reaches a break-even point.

From the Investopedia article on DDP [15]:

The discounted payback period is a capital budgeting procedure used to determine the profitability of a project. A discounted payback period gives the number of years it takes to break even from undertaking the initial expenditure, by discounting future cash flows and recognizing the time value of money. The metric is used to evaluate the feasibility and profitability of a given project.

The DDP concept is very simple as can be illustrated via an example. Consider Projects A and B from the example in Section 7.1.3, with an assumed cost of capital of 2.5%. Table 28 shows the discounted cash flows for Project A and the cumulative sum of the discounted cash flows. As can be seen, it is not until the end of year 6 that Project A has recovered the initial investment. (The discounting is done with respect to time 0.)

*Table 28. DPP for Project A*

| Years                             | 0         | 1         | 2         | 3        | 4        | 5        | 6      | 7      | 8      |
|-----------------------------------|-----------|-----------|-----------|----------|----------|----------|--------|--------|--------|
| Cash Flows                        | (155,000) | 20,000    | 20,000    | 30,000   | 40,000   | 30,000   | 30,000 | 25,000 | 5,000  |
| Discounted Cash Flows             | (155,000) | 19,512    | 19,036    | 27,858   | 36,238   | 26,516   | 25,869 | 21,032 | 4,104  |
| Accumulated Discounted Cash Flows | (155,000) | (135,488) | (116,452) | (88,594) | (52,356) | (25,840) | 29     | 21,061 | 25,164 |

Table 29 shows similar calculations for Project B, which takes an additional year to break even.

*Table 29. DPP for Project B*

| Years                             | 0         | 1         | 2        | 3        | 4        | 5        | 6        | 7      | 8      |
|-----------------------------------|-----------|-----------|----------|----------|----------|----------|----------|--------|--------|
| Cash Flows                        | (155,000) | 50,000    | 10,000   | 20,000   | 20,000   | 20,000   | 30,000   | 25,000 | 25,000 |
| Discounted Cash Flows             | (155,000) | 48,780    | 9,518    | 18,572   | 18,119   | 17,677   | 25,869   | 21,032 | 20,519 |
| Accumulated Discounted Cash Flows | (155,000) | (106,220) | (96,701) | (78,129) | (60,010) | (42,333) | (16,464) | 4,567  | 25,086 |

Based on DPP analysis, the decision should be to go forward with Project A since it achieves payback of the initial investment in 6 years whereas payback for Project B is not until the 7<sup>th</sup> year.

## 7.5 Profitability Index

The **Profitability Index** (PI) is the ratio of payoff to investment for a proposed project. It is useful for ranking projects because it allows one to quantify the amount of value created per unit of investment. More precisely, the profitability index is defined as the NPV of the cash inflows for a project divided by the NPV of the cash outflows. In terms of a formula, we have

$$PI = \frac{\sum_{k=1}^n CIF_k (1+i)^{-t_k}}{\sum_{j=1}^m COF_j (1+i)^{-t_j}}$$

where  $CIF_k$  stands for  $k^{th}$  cash inflow at time  $t_k$  and  $COF_j$  stands for  $j^{th}$  cash outflow at time  $t_j$ . Keep in mind that the cash outflows are converted to positive numbers for the calculation of the PI.

**[Author's remarks:** This looks worse than it is. It's just a matter of computing the NPV for the cash inflows and outflows separately and then dividing to get the PI.

Also, some (perhaps most) sources use an alternative definition of PI where the denominator only includes the initial investment and the numerator is the sum of the present values of all other cash flows. See, for example, the article on PI in Investopedia [16] which uses the alternate definition of PI.]

As an example, we will compute the PI for Projects A and B from Section 7.1.3, under the assumption of a 2.5% cost of capital:

$$\text{PI for Project A} = \frac{180,164}{155,000} = 1.1624$$

$$\text{PI for Project A} = \frac{180,086}{155,000} = 1.1618$$

The general rule is to select the project with the highest PI. So, in this case, we should go forward with Project A (albeit by a small margin).

## 7.6 Computational Support via Microsoft Excel

There are no supporting Python programs for this section. However, there is an Excel spreadsheet that covers many of the concepts in this section. The spreadsheet is in a file named “Capital Budgeting.xlsx”. The spreadsheet is located in the same Github directory as the Python programs.

A brief summary of the sheets within the spreadsheet is listed below:

- Acme Furniture – this sheet shows the NPV calculations for the Acme Furniture example under the 5 year and 6 year scenarios. The calculation of the IRR is also provided (using the Excel IRR function).
- Resort – this sheet provides an NPV analysis for the resort example in Section 7.1.2 concerning negative cash flows.
- Multiple Inv – this sheet supports the multiple investments example from Section 7.1.3.
- Restaurant – this sheet provides an NPV analysis for pessimistic and optimistic scenarios regarding a restaurant (from Section 7.1.4).
- Acme Furniture – scroll down on this sheet and there is a separate calculation for the loan scenario in Section 7.1.5
- Project Comparison – this sheet provides an analysis of Projects A and B in Section 7.1.6.
- Multiple IRRs – this sheet supports the multiple IRR example in Section 7.2.2.
- MIRR – this supports the MIRR example in Section 7.3.
- DDP and PI – this sheet supports DDP and PI examples that span Sections 7.4 and 7.5.
- Ex 1 – this sheet provides the answer to Exercise 1 in Section 7.7.
- Ex 2 – this sheet provides the answer to Exercise 2 in Section 7.7.

## 7.7 Exercises

1. Given an initial investment of \$70,000 and projected yearly cash flows of \$5,000, \$7,500, \$10,000, \$12,500, \$15,000, \$17,500 and \$20,000, compute the NPV for rates of return of 2, 3, 4, 5, 6 and 7 percent. **Answer:** See the sheet entitled "Ex 1" in the Excel file "Capital Budgeting.xlsx".
2. For the example in Section 7.1.5, assume the term of the loan is 10 years at 3% per year with semi-annual payments. Further, assume the project returns no cash inflows after year 6. Compute the NPV for the (semi-annual) cost of capital ranging from 2% to 8% in increments of .5%. **Hint:** Use the Amortization Calculator to determine the semi-annual payments. **Answer:** See the sheet entitled "Ex 2" in the Excel file "Capital Budgeting.xlsx".
3. For the example in Section 7.1.5, what is the maximum amount that Alice and Bob can loan (under the same terms as the example) and still have a positive NPV at 2% rate of return? **Hint:** Use the amortization calculator to determine the semi-annual loan repayments for different loan amounts, subtract the repayment amount from each of the cash flow estimates in Table 14 (just for the first years) and then compute the NPV for a 2% rate of return. You will need to experiment with several different loan amounts. **Answer:** For a loan in the amount of \$110,075, the NPV is about -\$9.86 (at 2% cost of capital). So, for a positive NPV (at 2% cost of capital), the loan needs to be a little less than \$110,075 (assuming a 3% nominal interest rate paid semi-annually on the loan).
4. Use the Microsoft Excel IRR function to compute the IRRs for Projects A and B in the example from Section 7.1.6. **Answer:** 6.31% for Project A and 6.36% for Project B.
5. Use the Microsoft Excel MIRR function to compute the MIRRs for Project A and B in the example from Section 7.1.6. Assume a reinvestment rate of .015 and a financing rate of .025. Do the calculation again with a reinvestment rate of .08 and a financing rate of .06. **Answer:** In the first scenario, 3.96% for Project A and 3.95% for Project B. In the second scenario, 7.15% for Project A and 7.20% for Project B.

## 8 Python Programs

### 8.1 Arithmetic Progression

```
# Arithmetic progressions
import tkinter as tk
import tkinter.ttk as ttk
import math
from functools import partial

root = tk.Tk()
root.title('Arithmetic Progression Calculator')

def quit():
    root.destroy()

def clear():
    try:
        t.destroy()
        t1.destroy()
    except:
        print('No need to destroy "progression" field')

    try:
        s.destroy()
        s1.destroy()
    except:
        print('No need to destroy "sum of progression" field')

    try:
        r.destroy()
        r1.destroy()
    except:
        print('No need to destroy "nth term in progression" field')

q1.delete(0,tk.END)
x.set(0)

q2.delete(0,tk.END)
```

```

y.set(0)

q3.delete(0,tk.END)
z.set(0)

a = []

def aprog (x,y,z):
# Function that computes the first n terms of an arithmetic progression and the sum of those terms
    global a,t,t1,s,s1
    start = x.get()
    step = y.get()
    n = z.get()

    # Array to hold the progression
    a = []

    # Compute progression
    next = start
    for i in range(n):
        a.append(next)
        next = next + step

    # Compute sum of the progression
    sum = round((n/2)*(a[0]+a[n-1]),3)

    # Output the progression
    # Compute width of text widgets given that 10 numbers will be print per row
    # This is under the assumption that the terms have three decimal places
    w = math.trunc(abs(start) + n * abs(step))
    # Number of digits in the longest number in the progression, plus the three digits after the
    decimal point
    # Add two for the comma and space after each number, and one for the decimal point
    w = len(str(w))+6
    if start+n*step > 0: # multiple by 10 to get the amount of space needed for each row of the
    output
        w = 10*w
    else: # need to add an extra space for the minus sign
        w = 10*(w+1)

```



```

# Create a text widget
t1= tk.Label(root, text="The progression is :")
t1.grid(row=7, column=1,padx=3,pady=11)
t = tk.Text(root, height=7, width=w)
t.grid(row=7,column=2,padx=3,sticky='w')

# Create a scrollbar and associate it with the text widget t
scrollb = ttk.Scrollbar(root, command=t.yview)
scrollb.grid(row=7, column=3, sticky='nesw')
t['yscrollcommand'] = scrollb.set

i=0
for x in a:
    i=i+1
    if i%10 != 0 and i != n:
        t.insert(tk.END, str(round(x,3)) + ', ')
    elif i != n:
        t.insert(tk.END, str(round(x,3)) + '\n')
    else:
        t.insert(tk.END, str(round(x,3)))

# Output sum of the progression
s1 = tk.Label(root, text="The sum of the progression is :")
s1.grid(row=8, column=1,padx=3,pady=11)
s = tk.Text(root,height=1,width=w)
s.insert(tk.END, str(sum))
s.grid(row=8,column=2,padx=3,sticky='w')

def nth_term (x,y,z):
# Function that computes the nth term in an arithmetic progression
    global r,r1
    start = x.get()
    step = y.get()
    n = z.get()

    # Compute nth term in progression
    n_term = start + (n-1)*step

```

```
# Print nth term
r1 = tk.Label(root, text="Term number %s in the progression:" %n)
r1.grid(row=6, column=1, padx=3, pady=11)
r = tk.Text(root, height=1, width=11)
r.insert(tk.END, str(n_term))
r.grid(row=6, column=2, padx=3, sticky='w')

# Input
tk.Label(root, text="Initial term in progression:").grid(row=1, column=1)
x = tk.DoubleVar()
q1=tk.Entry(root, textvariable=x)
q1.grid(row=1, column=2, padx=3, sticky='w')

tk.Label(root, text="Length of a step:").grid(row=2, column=1)
y = tk.DoubleVar()
q2=tk.Entry(root, textvariable=y)
q2.grid(row=2, column=2, padx=3, sticky='w')

# Enter the number (index) of the final term
tk.Label(root, text="Number of terms (n):").grid(row=3, column=1)
z = tk.IntVar()
q3=tk.Entry(root, textvariable=z)
q3.grid(row=3, column=2, padx=3, sticky='w')

aprog = partial(aprog, x, y, z)
button1 = tk.Button(root, text="Generate Progression and its
Sum", command=aprog).grid(row=4, column=1, padx=3, pady=3)

nth_term = partial(nth_term, x, y, z)
button2 = tk.Button(root, text="Generate nth term in
Progression", command=nth_term).grid(row=4, column=2, padx=3, pady=3, sticky='w')

button3 = tk.Button(root, text='Clear', command=clear).grid(row=11, column=1, pady=5)

button4 = tk.Button(root, text='Quit', command=quit).grid(row=11, column=2, padx=3,
pady=5, sticky='w')

root.mainloop()
```

## 8.2 Geometric Progression

```
# Geometric progressions
import tkinter as tk
import tkinter.ttk as ttk
import math
from functools import partial

root = tk.Tk()
root.title('Geometric Progression Calculator')

def quit():
    root.destroy()

def clear():
    try:
        t.destroy()
        u.destroy()
    except:
        print('No need to destroy "progression" field')

    try:
        s.destroy()
        v.destroy()
    except:
        print('No need to destroy "sum of progression" field')

    try:
        r.destroy()
        d.destroy()
    except:
        print('No need to destroy "nth term in progression" field')

q1.delete(0,tk.END)
x.set(0)
q2.delete(0,tk.END)
y.set(0)
q3.delete(0,tk.END)
z.set(0)
```

```
a = []
```

```
def aprog (x,y,z):
```

```
# Function that computes the first n terms of an arithmetic progression and the sum of those terms
```

```
    global a,t,s,u,v
```

```
    start = x.get()
```

```
    ratio = y.get()
```

```
    n = z.get()
```

```
# Array to hold the progression
```

```
a = []
```

```
# Compute progression
```

```
next = start
```

```
for i in range(n):
```

```
    a.append(next)
```

```
    next = next * ratio
```

```
# Compute sum of the progression
```

```
sum = start*(1-ratio**n)/(1-ratio)
```

```
# Create a text widget
```

```
u = tk.Label(root, text="The progression is :")
```

```
u.grid(row=7, column=1,padx=3,pady=11)
```

```
t = tk.Text(root, height=7, width=25)
```

```
t.grid(row=7,column=2,padx=3,sticky='w')
```

```
# Create a scrollbar and associate it with the text widget t
```

```
scrollb = ttk.Scrollbar(root, command=t.yview)
```

```
scrollb.grid(row=7, column=3, sticky='nesw')
```

```
t['yscrollcommand'] = scrollb.set
```

```
i=0
```

```
for x in a:
```

```
    i=i+1
```

```
    if i != n:
```

```
        t.insert(tk.END, round(x,6))
```

```
        t.insert(tk.END, '\n')
```

```
    else:
```

```

t.insert(tk.END, round(x,6))

# Output sum of the progression
v = tk.Label(root, text="The sum of the progression is :")
v.grid(row=8, column=1, padx=3, pady=11)
s = tk.Text(root, height=1, width=25)
s.insert(tk.END, round(sum,6))
s.grid(row=8, column=2, padx=3, sticky='w')

def nth_term (x,y,z):
# Function that computes the nth term in an geometric progression
    global r,d
    start = x.get()
    ratio = y.get()
    n = z.get()

    # Compute nth term in progression
    n_term = start*(ratio**(n-1))

    # Print nth term
    d = tk.Label(root, text="Term number %s in the progression:" %n)
    d.grid(row=6, column=1, padx=3, pady=11)
    r = tk.Text(root, height=1, width=25)
    r.insert(tk.END, round(n_term,6))
    r.grid(row=6, column=2, padx=3, sticky='w')

# Input
tk.Label(root, text="Initial term in progression:").grid(row=1, column=1)
x = tk.DoubleVar()
q1=tk.Entry(root, textvariable=x)
q1.grid(row=1, column=2, padx=3, sticky='w')

tk.Label(root, text="Common ration:").grid(row=2, column=1)
y = tk.DoubleVar()
q2=tk.Entry(root, textvariable=y)
q2.grid(row=2, column=2, padx=3, sticky='w')

# Enter the number (index) of the final term
tk.Label(root, text="Number of terms (n):").grid(row=3, column=1)

```

```

z = tk.IntVar()
q3=tk.Entry(root, textvariable=z)
q3.grid(row=3,column=2,padx=3,sticky='w')

aprog = partial(aprog,x,y,z)
button1 = tk.Button(root,text="Generate Progression and its
Sum",command=aprog).grid(row=4,column=1,padx=3,pady=3)

nth_term = partial(nth_term,x,y,z)
button2 = tk.Button(root,text="Generate nth term in
Progression",command=nth_term).grid(row=4,column=2,padx=3,pady=3,sticky='w')

button3 = tk.Button(root,text='Clear', command=clear).grid(row=11, column=1,pady=5)

button4 = tk.Button(root,text='Quit', command=quit).grid(row=11, column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

### 8.3 Interest Compounding Calculator

```

# Interest Compounding Calculator
import tkinter as tk
import tkinter.ttk as ttk
import math
from functools import partial

root = tk.Tk()
root.title('Interest Compounding Calculator')

def quit():
    root.destroy()

def clear():
    s.destroy()
    t.destroy()

q1.delete(0,tk.END)
w.set(0)
q2.delete(0,tk.END)

```

```

x.set(0)
q3.delete(0,tk.END)
y1.set(0)
q4.delete(0,tk.END)
y2.set(0)
q5.delete(0,tk.END)
z.set(0)

def principal (w,x,y1,y2,z):
# Function that computes final principal
    global s,t
    orig_prin = w.get()
    rate = x.get()/100
    try:
        freq = y1.get()
    except:
        freq = 0
    continuous = y2.get()
    time = z.get()

# Compute final principal
    if continuous == False:
        fin_prin = orig_prin*(1+(rate/freq)**(freq*time))
    else:
        fin_prin = orig_prin*math.exp(rate*time)

# Display final principal
    t = tk.Label(root, text="The final principal is:")
    t.grid(row=7, column=1,padx=3,pady=5,sticky='w')
    s = tk.Text(root,height=1,width=15)
    s.insert(tk.END, round(fin_prin,2))
    s.grid(row=7,column=2,padx=3,sticky='w')

# Input
tk.Label(root, text="Original principal:").grid(row=1,column=1,padx=3,sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

```

```

tk.Label(root, text="Annual interest percentage:").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Compounding frequency per year: \n (leave this blank if continuous \n
compounding is set to True)").grid(row=3,column=1,padx=3,sticky='w')
y1 = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y1)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Continuous compounding
(True/False):").grid(row=4,column=1,padx=3,sticky='w')
y2 = tk.BooleanVar()
q4=tk.Entry(root, textvariable=y2)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')

# The user can also enter fractions, e.g., 1.25
tk.Label(root, text="Time that interest accrues in years: \n (Need not be a whole number of years)",
justify='left').grid(row=5,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q5=tk.Entry(root, textvariable=z)
q5.grid(row=5, column=2,padx=5,pady=3,sticky='w')

# Call function to compute the final principal
principal = partial(principal,w,x,y1,y2,z)
button1 = tk.Button(root,text="Compute final
principal",command=principal).grid(row=6,column=1,padx=3,pady=7)

button2 = tk.Button(root,text='Clear', command=clear).grid(row=11,column=1,pady=5)

button3 = tk.Button(root,text='Quit', command=quit).grid(row=11,column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

## 8.4 Block Annuity Calculator

# Block Annuity Calculator

# This program computes the total value of a savings annuity over several blocks.



# Each block has a fixed interest rate per period and fixed payment (i.e., investment)  
 # per period. The interest rate and payment may vary among the blocks.

```
import csv, math
```

```
class Block:
```

```
    # Instances of this class represent one block of a savings annuity
    # Each block has a fixed interest rate per period and a given number of periods.
    # At the end of each period, a fixed investment is made and the existing investment
    # is compounded. The following attributes are required for each block:
    #   i - interest rate per period
    #   r - the amount of money invested by the annuity owner at the end of each period
    #   n - the number of periods in the block
    #   v - value of the block when it ends, i.e., when the interest rate changes and a
    #       new block is created
```

```
    def __init__(blk, i, r, n):
```

```
        blk.i = i
        blk.r = r
        blk.n = n
        blk.v = 0
```

```
    def acc_value(blk):
```

```
        # This function computes the accumulate value of the annuity for the block
        g = (1 + blk.i)**blk.n
        blk.v = blk.r*(g-1)/blk.i
        return blk.v
```

```
    def fin_value(blk, i_new, n_new):
```

```
        # After a block X is complete, i.e., when the interest rate changes and a new block is started,
        # Block X will continue to earn interest based on one or more interest rates for the subsequent
        # blocks, if any. This function needs to be called once for each block subsequent to a given
        block.
```

```
        blk.i_new = i_new
        blk.n_new = n_new
        blk.v = blk.v * (1 + blk.i_new)**blk.n_new
        return blk.v
```

```
# B is an array that holds the classes that represent the blocks
```

```
B = []
```

```
# Read in the csv file that has information concerning each block.
# Each line of the csv file represents on block.
# Each line has the payment per period, interest per period and
# the number of periods for a given block. The items are delimited commas.
```

```
name = input('Enter name of csv file (do not add .csv to the name): ')
```

```
with open('%s.csv' % name) as f:
```

```
    blocks = csv.reader(f, delimiter=',')
```

```
    for row in blocks:
```

```
        # Create a class for each block and put the classes in an array B
```

```
        B.append(Block(float(row[0]), float(row[1]), float(row[2])))
```

```
num_blocks = len(B)
```

```
# V is an array that holds the value for each block
```

```
V = []
```

```
# Compute the accrued value for each block but only during the period
# when the block is active. In an second step, the value of the block
# at the end of the term is computed (this includes accrued interest
# while subsequent blocks are active).
```

```
for k in range(num_blocks):
```

```
    V.append(B[k].acc_value())
```

```
print("")
```

```
print('The value of each block without interest accrued during subsequent blocks:')
```

```
for k in range(num_blocks):
```

```
    print("Block " + str(k) + ': ' + '${:,.2f}'.format(V[k]))
```

```
# The second step is to compute the interest accrued to each block during subsequent blocks.
```

```
# This entails multiple calls to the fin-value function.
```

```
for k in range(num_blocks-1):
```

```
    for j in range(k+1,num_blocks):
```

```
        V[k] = B[k].fin_value(B[j].i,B[j].n)
```

```
# Print out the value of each block at the end of the annuity's term
```

```
print("")
```

```
print('The value of each block at the end of the annuity\'s term, \
```

i.e., with the interest accrued during subsequent blocks:')

```
for k in range(num_blocks):
    print("Block " + str(k) + ': ' + '${:,.2f}'.format(V[k]))
```

```
# Total value of annuity at term
```

```
total_value = 0
```

```
for k in range(num_blocks):
```

```
    total_value = total_value + V[k]
```

```
print('')
```

```
print('Total value of the annuity at the end of its term:')
```

```
print('${:,.2f}'.format(total_value))
```

## 8.5 Annuity Payout Calculator

```
# Discounted Value of Ordinary, Simple, Certain Annuities
```

```
# This program covers the payouts from an ordinary, simple, certain annuity
```

```
# It is assumed that the compounding and the periodic payouts happen at the same time
```

```
# One can enter any two of n, q or r and the program will compute the third:
```

```
# i = interest (per period)
```

```
# n = number of periods
```

```
# q = initial investment amount
```

```
# r = payment to investor per period
```

```
import tkinter as tk
```

```
from tkinter import *
```

```
import math
```

```
from functools import partial
```

```
root = tk.Tk()
```

```
root.title('Simple Annunity Payout Calculator')
```

```
def quit():
```

```
    root.destroy()
```

```
def clear():
```

```
    try:
```

```
        s.destroy()
```

```
    except:
```

```
        print("No need to destroy attribute")
```

```
    try:
```

```

        t.destroy()
    except:
        print("No need to destroy attribute")
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q4.delete(0,tk.END)
    z.set(0)

def compute_var(w,x,y,z):
    # Function that computes the unknown variable
    global s, t
    q = w.get()    # initial investment amount
    i = x.get()    # interest (per period)
    try:
        n = y.get() # number of payouts
    except:
        n = 0      # this is need is the use delete the default setting
    try:
        r = z.get() # amount per payout
    except:
        r = 0      # this is need is the use delete the default setting

    # Determine which one of n, q or r has not be entered and compute the third
    # variable and then display the value.
    if n==0:
        # Compute n
        if r - i*q <= 0: # payout is less than accrued interest per period
            # Display n
            t = tk.Label(root, text="Number of payouts:")
            t.grid(row=9, column=1,padx=3,pady=5,sticky='w')
            s = tk.Text(root,height=1,width=11)
            s.insert(tk.END, 'infinite')
            s.grid(row=9,column=2,padx=3,sticky='w')

        elif r > q*(i+1):

```

# desired periodic payout is greater than initial investment amount accrued interest in first period

```
print('Error: Not possible since requested periodic payout is greater')
print('than value of initial investment at the end of the first period')
```

else:

```
n = (math.log10(r) - math.log10(r - i*q))/math.log10(1 + i)
```

# Display n

```
t = tk.Label(root, text="Number of payouts:")
```

```
t.grid(row=9, column=1, padx=3, pady=5, sticky='w')
```

```
s = tk.Text(root, height=1, width=11)
```

```
s.insert(tk.END, round(n,2))
```

```
s.grid(row=9, column=2, padx=3, sticky='w')
```

elif q==0:

# Compute q

```
q = (r/i)*(1-(1+i)**(-n))
```

# Display q

```
t = tk.Label(root, text="Initial investment:")
```

```
t.grid(row=9, column=1, padx=3, pady=5, sticky='w')
```

```
s = tk.Text(root, height=1, width=11)
```

```
s.insert(tk.END, '${:,.2f}'.format(q))
```

```
s.grid(row=9, column=2, padx=3, sticky='w')
```

else:

# Compute r

```
r = (i*q)/(1-(1+i)**(-n))
```

# Display r

```
t = tk.Label(root, text="Payout per period:")
```

```
t.grid(row=9, column=1, padx=3, pady=5, sticky='w')
```

```
s = tk.Text(root, height=1, width=11)
```

```
s.insert(tk.END, '${:,.2f}'.format(r))
```

```
s.grid(row=9, column=2, padx=3, sticky='w')
```

# Input

```
tk.Label(root, text="Initial investment in annuity:").grid(row=1, column=1, padx=3, sticky='w')
```

```
w = tk.DoubleVar()
```

```
q1=tk.Entry(root, textvariable=w)
```

```

q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Interest per period (must provide a
value:)").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of payouts:").grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Amount of each payout:").grid(row=4,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q4=tk.Entry(root, textvariable=z)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')

# Call function to compute unknown variable
compute_var = partial(compute_var,w,x,y,z)
button1 = tk.Button(root,text="Compute Remaining
Variable",command=compute_var).grid(row=6,column=1,padx=3,pady=7)

button2 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)

button3 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

## 8.6 Deferred Annuity Calculator

```

# Deferred Annuities
# This program determines the present value (q) or the regular payout amount (r) for a
# deferred annuity, given
# i = interest (per period)
# n = number of payments
# m = number of deferral periods before payments start
# To be clear, one needs to enter q or r, and the other value is computed
import tkinter as tk

```

```
from tkinter import *
import math
from functools import partial

root = tk.Tk()
root.title('Deferred Annuity Calculator')

def quit():
    root.destroy()

def clear():
    try:
        s.destroy()
    except:
        print("No need to destroy attribute")
    try:
        t.destroy()
    except:
        print("No need to destroy attribute")
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q4.delete(0,tk.END)
    z.set(0)
    q5.delete(0,tk.END)
    v.set(0)

def compute_var(w,x,y,z,v):
# Function that computes the unknown variable
    global s, t
    q = w.get()    # initial investment amount
    i = x.get()    # interest (per period)
    n = y.get()    # number of payouts
    m = z.get() - 1 # number of deferral periods
    r = v.get()    # amount per payout
```

# Determine which one of q or r has not be entered and compute the other  
# variable, and then display the value.

```
if q==0:
    # Compute q
    a = (1 + i)**(-m-1)
    b = (1 + i)**-n
    c = (1 + i)**-1
    q = r * a * (1 - b)/(1 - c)

    # Display n
    t = tk.Label(root, text="Present value of annuity:")
    t.grid(row=9, column=1, padx=3, pady=5, sticky='w')
    s = tk.Text(root, height=1, width=11)
    s.insert(tk.END, '${:,.2f}'.format(q))
    s.grid(row=9, column=2, padx=3, sticky='w')
```

```
else:
    # Compute r
    a = (1 + i)**(-m-1)
    b = (1 + i)**-n
    c = (1 + i)**-1
    r = q * (1 - c)/(a * (1 - b))

    # Display r
    t = tk.Label(root, text="Payout per period:")
    t.grid(row=9, column=1, padx=3, pady=5, sticky='w')
    s = tk.Text(root, height=1, width=11)
    s.insert(tk.END, '${:,.2f}'.format(r))
    s.grid(row=9, column=2, padx=3, sticky='w')
```

# Input

```
tk.Label(root, text="Initial investment, i.e., present value of  
annuity:").grid(row=1, column=1, padx=3, sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2, padx=5, pady=3, sticky='w')
```

```
tk.Label(root, text="Interest per period:").grid(row=2, column=1, padx=3, sticky='w')
x = tk.DoubleVar()
```



```

q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of payouts:").grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of deferral periods:").grid(row=4,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q4=tk.Entry(root, textvariable=z)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Amount of each payout:").grid(row=5,column=1,padx=3,sticky='w')
v = tk.DoubleVar()
q5=tk.Entry(root, textvariable=v)
q5.grid(row=5, column=2,padx=5,pady=3,sticky='w')

# Call function to compute unknown variable
compute_var = partial(compute_var,w,x,y,z,v)
button1 = tk.Button(root,text="Compute Unknown
Variable",command=compute_var).grid(row=6,column=1,padx=3,pady=7)

button2 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)

button3 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

## 8.7 Amortization Calculator

```

# Amortization Calculator
# The program offers three different capabilities, i.e.,
# - Compute the periodic payments on a mortgage given the loan amount, interest rate,
#   number of payments per year and the length of the loan in years (doesn't need to be a whole
#   number)
# - Compute the maximum affordable loan given the interest rate, number of payments per year,
#   length
#   of the loan in years and the maximum affordable payment per period

```

# - Compute the interest versus principal payback breakdown

```
import tkinter as tk
import tkinter.ttk as ttk
import csv

root = tk.Tk()
root.title('Amortization Calculator')

def quit():
    root.destroy()

def clear():
    # This function clears the various input and output lines in the user interface
    try:
        s.delete('1.0',tk.END)
    except:
        print('No need to clear epa field')
    try:
        t.delete('1.0',tk.END)
    except:
        print('No need to clear max_loan field')
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q4.delete(0,tk.END)
    z.set(0)
    q5.delete(0,tk.END)
    z2.set(0)

def adj_input(w,x,y,z,z2):
    # This function gets the input needed for the other functions
    global p,r,n,max_a
    try:
        p = w.get()
    except:
```

```

    p = 0

    yr_rate = x.get()
    pay_per_yr = y.get()
    num_yrs = z.get()
    r = (yr_rate/100)/12    # interest rate per repayment period
    n = pay_per_yr * num_yrs    # total number of payments

    try:
        max_a = z2.get()    # max amount that user can afford per repayment period
    except:
        max_a = 0

def payment():
    # Function that equal repayment amounts per time period
    global s
    #Get input
    adj_input(w,x,y,z,z2)

    # Compute equal payment amount (epa) per period
    g = (1+r)**n
    epa=p*r*g/(g-1)

    # Display payment amount per period
    tk.Label(root, text="The payment amount per period is:").grid(row=9,
column=1,padx=3,pady=5,sticky='w')
    s = tk.Text(root,height=1,width=15)
    s.insert(tk.END, '${:,.2f}'.format(epa))
    s.grid(row=9,column=2,padx=3,sticky='w')

    return epa

def max_afford_loan():
    global t
    # Get input
    adj_input(w,x,y,z,z2)

    # Compute maximum mortgage that use can afford
    g = (1+r)**n

```

```

max_loan = (max_a*(g-1))/(r*g)

# Display maximum affordable loan
tk.Label(root, text="Maximum affordable mortgage:").grid(row=18,
column=1,padx=3,pady=5,sticky='w')
t = tk.Text(root,height=1,width=15)
t.insert(tk.END, '${:,.2f}'.format(max_loan))
t.grid(row=18,column=2,padx=3,sticky='w')

def breakdown():
    global t
    a = payment()
    m = int(n)

    It = [] # Interest repayment array
    Pr = [] # Principal repayment array
    It.append(r*p)
    Pr.append(a - It[0])
    for i in range(1,m):
        It.append((r+1)*It[i-1] - r*a)
        Pr.append(a - It[i])

    # Output breakdown to a file
    with open('breakdown.csv', 'w',newline='') as f:
        writer = csv.writer(f)
        writer.writerow(["Period", "Interest", "Principal"])
        for i in range(m):
            writer.writerow([i+1,round(It[i],2),round(Pr[i],2)])

# Input
tk.Label(root, text="Amount of loan:").grid(row=1,column=1,padx=3,sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Annual interest rate:").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

```

```

num_pay=tk.Label(root, text="Number of payments per year:")
num_pay.grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Length of loan in years:").grid(row=4,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q4=tk.Entry(root, textvariable=z)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Max affordable payment per period \n e.g., max one can afford to pay per
month:").\
    grid(row=5,column=1,padx=3,sticky='w')
z2 = tk.DoubleVar()
q5=tk.Entry(root, textvariable=z2)
q5.grid(row=5, column=2,padx=5,pady=3,sticky='w')

# Retrieve and adjust input, and make adjusted input global
adj_input(w,x,y,z,z2)

button1 = tk.Button(root,text="Compute periodic
payment",command=payment).grid(row=8,column=1,padx=3,pady=7,sticky='nesw')

button2 = tk.Button(root,text='Max affordable loan given interest rate, \n number of payments per
year, \
length of loan \n and max affordable payment per period', \
    command=max_afford_loan).grid(row=17,column=1,padx=5, pady=5,sticky='nesw')

button3 = tk.Button(root,text='Compute Interest-Principal breakdown \n Result placed in
breakdown.csv', \
    command=breakdown).grid(row=19,column=1,padx=5, pady=5,sticky='nesw')

button4 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)

button5 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

## 8.8 Balloon Loan Calculator

```
# Balloon Loan Calculator
# This calculator can be used to compute either the regular payment for
# a balloon loan, or the present value of a balloon loan given the amount
# of the regular payments.
import tkinter as tk
import tkinter.ttk as ttk

root = tk.Tk()
root.title('Balloon Loan Calculator')

def quit():
    root.destroy()

def clear():
    s.destroy()
    t.destroy()
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q5.delete(0,tk.END)
    z1.set(0)
    q6.delete(0,tk.END)
    z2.set(0)

def adj_input(w,x,y,z1,z2):
    try:
        p = w.get()      # current value of loan
    except:
        p = 0           # All the variables are initial set to zero in the GUI, but the user may blank out
        the value
    r = x.get()          # interest rate per payment period
    n = y.get()          # total number of payments
    b = z1.get()          # amount of the balloon payment
```

```

try:
    epa = z2.get()    # amount of each recurring payment
except:
    epa = 0          # All the variables are initial set to zero in the GUI, but the user may blank out
the value

return (p,r,n,b,epa)

def compute_var():
# Function computes the unknown variable either P (current value) or
# the payment amount
    global s,t
    p,r,n,b,epa = adj_input(w,x,y,z1,z2)

    if epa==0:
        # Compute equal payment amount (epa) per period
        g = (1+r)**n
        epa=(r*p*g - r*b)/(g-1)

        # Display final amount of regular payment
        t = tk.Label(root, text="The payment amount per period is:")
        t.grid(row=9, column=1,padx=3,pady=5,sticky='w')
        s = tk.Text(root,height=1,width=15)
        s.insert(tk.END, '${:,.2f}'.format(epa))
        s.grid(row=9,column=2,padx=3,sticky='w')
    else:
        # Compute present value of loan
        g = (1+r)**n
        h = (epa * (g-1)) + r*b
        k = r*g
        p = h/k

        # Display final amount of regular payment
        t = tk.Label(root, text="The current value of the loan is:")
        t.grid(row=9, column=1,padx=3,pady=5,sticky='w')
        s = tk.Text(root,height=1,width=15)
        s.insert(tk.END, '${:,.2f}'.format(p))
        s.grid(row=9,column=2,padx=3,sticky='w')

```

```
    return

# Input
tk.Label(root, text="Amount of loan:").grid(row=1,column=1,padx=3,sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Interest rate per period:").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

num_pay=tk.Label(root, text="Total number of payments:")
num_pay.grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Amount of the balloon payment:").grid(row=5,column=1,padx=3,sticky='w')
z1 = tk.DoubleVar()
q5=tk.Entry(root, textvariable=z1)
q5.grid(row=5, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Amount of each periodic payment:").grid(row=6,column=1,padx=3,sticky='w')
z2 = tk.DoubleVar()
q6=tk.Entry(root, textvariable=z2)
q6.grid(row=6, column=2,padx=5,pady=3,sticky='w')

# Compute the unknown variable
button1 = tk.Button(root,text="Compute unkown
variable",command=compute_var).grid(row=8,column=1,padx=3,pady=7)

# Clear and quit command buttons
button2 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)

button3 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')
```



```
root.mainloop()
```

## 8.9 Loan Refinance Calculator

```
# Loan Refinance Calculator
# This program analyzes the refinancing of a loan (such as a mortgage).
#
# It is assumed that the period for the original loan (e.g., a month) is
# the same as the period for the new loan.

import tkinter as tk

root = tk.Tk()
root.title('Loan Refinancing Calculator')

def quit():
    root.destroy()

def clear():
    # This function clears the various input and output lines in the user interface
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q4.delete(0,tk.END)
    z.set(0)

    q5.delete(0,tk.END)
    w1.set(0)
    q6.delete(0,tk.END)
    x1.set(0)
    q7.delete(0,tk.END)
    y1.set(0)

def get_input():
    # This function gets the input needed for the other functions
```

```

global p0,i0,n0,n0_com,cost,i1,n1

# Information for the initial loan that is to be refinanced
p0 = w.get()    # principal for initial loan
i0 = x.get()    # interest rate per period for initial loan
n0 = y.get()    # number of periods for initial loan
n0_com = z.get() # number of periods completed when refinancing is to start

# Information for refinancing
cost = w1.get() # Cost of refinancing, including penalty for terminating initial loan
i1 = x1.get()  # interest rate for new loan
n1 = y1.get()  # number of periods for new loan

print('\n' + '___' + '\n')

def analyze():
# Function that analyzes a potential refinancing of a loan
#Get input
get_input()

# Compute equal payment amount (epa0) per period for initial loan
g = (1+i0)**n0
epa0 = p0 * i0 * g/(g-1)

# Compute balance of initial load at the time of planned refinancing
h = (1+i0)**n0_com
bal = h*p0 - epa0*(h-1)/i0
print("The balance on the initial loan is " + '${:,.2f}'.format(bal) + '\n')

# Compute equal payment amount (epa1) per period for refinanced loan
# The initial balance (principal) for the new loan is the balance from the
# initial loan.
g1 = (1+i1)**n1
epa1 = bal * i1 * g1/(g1-1)

# Refinancing analysis

# Periodic payments

```

```

print('The periodic payment for the initial loan is ' + '${:,.2f}'.format(epa0))
print('The periodic payment for the new loan is ' + '${:,.2f}'.format(epa1) + '\n')

# Comparison of payments (including principal payback) from the time of the refinancing decision
point
# For initial loan if not refinanced
t_pay0 = epa0 * (n0 - n0_com)
print('The remaining payments for initial loan, if not refinanced, equal ' +
      '${:,.2f}'.format(t_pay0))
print('of which ' + '${:,.2f}'.format(t_pay0 - bal) + ' are interest payments.' + '\n')

# For refinanced loan, including cost to terminate initial loan and cost for new loan
t_pay1 = (epa1 * n1) + cost
print('The payments in the refinancing scenario, which includes the costs to')
print('terminate the initial loan and to initiate the new loan, equal ' + '${:,.2f}'.format(t_pay1))
print('The interest paid in this scenario is ' + '${:,.2f}'.format(t_pay1 - bal) + '\n')

# Amount of time to recover cost of refinancing
if epa1 < epa0:
    cost_recover = int(cost / (epa0 - epa1)) + 1
    print("It will take " + str(cost_recover) + " periods to recover the cost of refinancing.")
else:
    print("The new monthly payment would be more than the existing monthly payment, and so,
it")
    print("is not recommended to refinance.")

#
# Input
#
tk.Label(root, text="Amount of initial loan:").grid(row=1,column=1,padx=3,sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Interest rate per period for initial
loan:").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

```

```
tk.Label(root, text="Number of periods for initial loan:").grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')
```

```
tk.Label(root, text="Number of periods completed in initial \n loan when
refinancing:").grid(row=4,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q4=tk.Entry(root, textvariable=z)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')
```

```
# It is assumed that the entire remaining balance on the initial loan is to be refinanced
```

```
tk.Label(root, text="Penalties on termination of initial loan \n and financing costs for new
loan:").grid(row=7,column=1,padx=3,sticky='w')
w1 = tk.DoubleVar()
q5=tk.Entry(root, textvariable=w1)
q5.grid(row=7, column=2,padx=5,pady=3,sticky='w')
```

```
tk.Label(root, text="Interest rate per period for refinanced
loan:").grid(row=8,column=1,padx=3,sticky='w')
x1 = tk.DoubleVar()
q6=tk.Entry(root, textvariable=x1)
q6.grid(row=8, column=2,padx=5,pady=3,sticky='w')
```

```
tk.Label(root, text="Number of periods for refinanced
loan:").grid(row=9,column=1,padx=3,sticky='w')
y1 = tk.DoubleVar()
q7=tk.Entry(root, textvariable=y1)
q7.grid(row=9, column=2,padx=5,pady=3,sticky='w')
```

```
# Control buttons
```

```
button1 = tk.Button(root,text="Analyze
refinancing",command=analyze).grid(row=13,column=1,padx=3,pady=7,sticky='nesw')
```

```
button4 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)
```

```
button5 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')
```

```
root.mainloop()
```

## 8.10 Loan Refinance – Breakeven Interest Rate Analyzer

```
# Loan Refinance - Breakeven Interest Rate Analyzer
# This program analyzes the refinancing of a loan (such as a mortgage).
# The program prepares a list of scenarios for various interest rates up until the
# refinancing rate leads to a situation where the remaining payments are more
# than that of the original loan.
#
# It is assumed that the period for the original loan (e.g., a month) is
# the same as the period for the new loan.

import tkinter as tk
import csv

root = tk.Tk()
root.title('Loan Refinance - Interest Rate Analyzer')

def quit():
    root.destroy()

def clear():
    # This function clears the various input and output lines in the user interface
    q1.delete(0,tk.END)
    w.set(0)
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q4.delete(0,tk.END)
    z.set(0)

    q5.delete(0,tk.END)
    w1.set(0)
    q6.delete(0,tk.END)
    x1.set(0)
    q7.delete(0,tk.END)
```

```

y1.set(0)
q8.delete(0,tk.END)
z1.set(0)

def get_input():
    # This function gets the input needed for the other functions
    global p0,i0,n0,n0_com,cost,i1,i_low,n1,n_per_y

    # Information for the initial loan that is to be refinanced
    p0 = w.get()    # principal for initial loan
    i0 = x.get()    # interest rate per period for initial loan
    n0 = y.get()    # number of periods for initial loan
    n0_com = z.get() # number of periods completed when refinancing is to start

    # Information for refinancing
    # cost of refinancing, including penalty for terminating initial loan
    cost = w1.get()
    # interest rate at which to start analysis - this should be less than the rate for the existing loan
    i_low = x1.get()
    # number of periods for new loan
    n1 = y1.get()
    # number of payments (and compoundings) per year
    # This is used to convert the interest rates from a decimal per month to nominal yearly rates
    n_per_y = z1.get()

def analyze():
    # Function that analyzes a potential refinancing of a loan
    #Get input
    get_input()

    # Compute equal payment amount (epa0) per period for initial loan
    g = (1+i0)**n0
    epa0 = p0 * i0 * g/(g-1)

    # Compute balance of initial load at the time of planned refinancing
    h = (1+i0)**n0_com
    bal = h*p0 - epa0*(h-1)/i0

    # Breakeven interest table

```

```

It = [] # Periodic interest rate in increments of .0001
T_pay = [] # Total payments for new loan at given interest rates
Epa = [] # Equal payment amounts at given interest rates
It.append(i0)
t_pay0 = epa0 * (n0 - n0_com)
T_pay.append(t_pay0)
Epa.append(epa0)

# Start with an interest rate .5 of i0 and add .0001 to the rate until
# the total payments for the new loan are greater than the remaining
# payments on the existing loan, i.e., epa0 * (n0 - n0_com)

t_payment = 0
interest = i_low
It.append(interest)
while t_payment < t_pay0:
    # Compute total payment for new loan with updated interest rate
    g1 = (1 + interest)**n1
    epa = bal * interest * g1/(g1-1)
    Epa.append(epa)
    t_payment = (epa * n1) + cost
    T_pay.append(t_payment)

    # Increment interest
    interest = interest + .00005
    It.append(interest)

# Output breakdown to a file
# In the output file, the first line of data concerns the existing laon.
# The following lines show total payments and payments per period for different interest,
# up to the point where the payments for the new loan are greater than the remaining
# payments on the existing loan.

m = len(T_pay)
with open('breakeven_int.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(["Interest Rate per Period", "Nominal Annual Interest Rate", "Total Payments",
"Payment per Period"])

```

```

    for k in range(m):
        if k==0:
            writer.writerow([ round(It[k],7), '{:,.2f}%'.format(12*100*It[k]), '${:,.2f}'.format(T_pay[k]),
                              '${:,.2f}'.format(Epa[k]), 'Info for existing loan' ])
        else:
            writer.writerow([ round(It[k],7), '{:,.2f}%'.format(12*100*It[k]), '${:,.2f}'.format(T_pay[k]),
                              '${:,.2f}'.format(Epa[k]) ])

#
# Input
#
tk.Label(root, text="Amount of initial loan:").grid(row=1,column=1,padx=3,sticky='w')
w = tk.DoubleVar()
q1=tk.Entry(root, textvariable=w)
q1.grid(row=1, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Interest rate per period for initial
loan:").grid(row=2,column=1,padx=3,sticky='w')
x = tk.DoubleVar()
q2=tk.Entry(root, textvariable=x)
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of periods for initial loan:").grid(row=3,column=1,padx=3,sticky='w')
y = tk.DoubleVar()
q3=tk.Entry(root, textvariable=y)
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of periods completed in initial \n loan when
refinancing:").grid(row=4,column=1,padx=3,sticky='w')
z = tk.DoubleVar()
q4=tk.Entry(root, textvariable=z)
q4.grid(row=4, column=2,padx=5,pady=3,sticky='w')

# It is assumed that the entire remaining balance on the initial loan is to be refinanced

tk.Label(root, text="Penalties on termination of initial loan \n and financing costs for new
loan:").grid(row=7,column=1,padx=3,sticky='w')
w1 = tk.DoubleVar()
q5=tk.Entry(root, textvariable=w1)
q5.grid(row=7, column=2,padx=5,pady=3,sticky='w')

```



```

tk.Label(root, text="Interest rate at which to start analysis - \n should be less than rate for existing
loan:").grid(row=8,column=1,padx=3,sticky='w')
x1 = tk.DoubleVar()
q6=tk.Entry(root, textvariable=x1)
q6.grid(row=8, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of periods for refinanced
loan:").grid(row=9,column=1,padx=3,sticky='w')
y1 = tk.DoubleVar()
q7=tk.Entry(root, textvariable=y1)
q7.grid(row=9, column=2,padx=5,pady=3,sticky='w')

tk.Label(root, text="Number of payments per year for refinanced
loan:").grid(row=10,column=1,padx=3,sticky='w')
z1 = tk.DoubleVar()
q8=tk.Entry(root, textvariable=z1)
q8.grid(row=10, column=2,padx=5,pady=3,sticky='w')

# Control buttons

button1 = tk.Button(root,text="Analyze refinancing interest rates \n Solution is put in file named
breakeven_int.csv",command=analyze).grid(row=13,column=1,padx=3,pady=7,sticky='nesw')

button4 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)

button5 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,
pady=5,sticky='w')

root.mainloop()

```

## 8.11 Bond Schedule Creator

```

# Bond Schedule Creator
# This program creates a schedule of payments for a bond.
# The output (in a csv file) shows the time period, coupon interest,
# interest on book value and book value adjustment for each period.
# The program works for bonds purchased at a premium, at a discount
# and at face value.

```

```
import tkinter as tk
import csv
import numpy as np

root = tk.Tk()
root.title('Bond Schedule Creator')

def quit():
    root.destroy()

def clear():
    q2.delete(0,tk.END)
    x.set(0)
    q3.delete(0,tk.END)
    y.set(0)
    q5.delete(0,tk.END)
    z1.set(0)
    q6.delete(0,tk.END)
    z2.set(0)

def adj_input():
    face = x.get()    # face value of bond (assumed to be the same as the redemption value)
    n = y.get()       # number of payment periods
    i = z1.get()      # coupon rate (this is applied to the face value to determine periodic payouts)
    yr = z2.get()     # yield rate
    return (face,n,i,yr)

def compute_schedule():
    # Function that computes the bond schedule
    # Retrieve input
    face,n,i,yr = adj_input()

    # Determine the coupon (periodic payment)
    a = face * i

    # Determine purchase price that results in given yield
    g = (1 + yr)**(-n)
    h = (1 - g)/yr
```

```

p = a * h + face * g

# Define arrays to be populated with information for bond schedule table
bv = [] # array for book value
adj = [] # array for book value adjustment
ibv = [] # array for interest on book value

# Initialize arrays
bv.append(p)
adj.append(0)
ibv.append(0)

# Compute book value, book value adjustment and interest on book value for each period
for k in range(1,int(n)+1):
    ibv.append(yr*bv[k-1])
    adj.append(a - ibv[k])
    bv.append(bv[k-1] - adj[k])

# Generate bond schedule and write to file
with open('bond_schedule.csv', 'w',newline='') as f:
    writer = csv.writer(f)

    # Header row of table
    writer.writerow(["Period", "Coupon","Interest on Book Value", "Book Value
Adjustment","Book Value"])

    # Row of table body
    writer.writerow([ '0','-','-','-','${:,.2f}'.format(bv[0]) ])

    # Rest of table body
    for k in range(1,int(n)+1):
        writer.writerow([ k,'${:,.2f}'.format(a),'${:,.2f}'.format(ibv[k]),\
            '${:,.2f}'.format(adj[k]),'${:,.2f}'.format(bv[k]) ])

    # Totals
    coupon_total = n * a
    ibv_total = np.sum(ibv)
    adj_total = np.sum(adj)
    writer.writerow([ 'Totals','${:,.2f}'.format(coupon_total),\

```

```
'${:,.2f}'.format(ibv_total),'${:,.2f}'.format(adj_total),'-' ])  
  
    return  
  
# Input  
tk.Label(root, text="Redemption value of bond:").grid(row=2,column=1,padx=3,sticky='w')  
x = tk.DoubleVar()  
q2=tk.Entry(root, textvariable=x)  
q2.grid(row=2, column=2,padx=5,pady=3,sticky='w')  
  
num_per=tk.Label(root, text="Number of payment periods:")  
num_per.grid(row=3,column=1,padx=3,sticky='w')  
y = tk.DoubleVar()  
q3=tk.Entry(root, textvariable=y)  
q3.grid(row=3, column=2,padx=5,pady=3,sticky='w')  
  
tk.Label(root, text="Coupon rate per period:").grid(row=5,column=1,padx=3,sticky='w')  
z1 = tk.DoubleVar()  
q5=tk.Entry(root, textvariable=z1)  
q5.grid(row=5, column=2,padx=5,pady=3,sticky='w')  
  
tk.Label(root, text="Yield rate per period:").grid(row=6,column=1,padx=3,sticky='w')  
z2 = tk.DoubleVar()  
q6=tk.Entry(root, textvariable=z2)  
q6.grid(row=6, column=2,padx=5,pady=3,sticky='w')  
  
# Compute the bond schedule  
button1 = tk.Button(root,text="Compute bond  
schedule",command=compute_schedule).grid(row=8,column=1,padx=3,pady=7)  
  
# Clear and quit command buttons  
button2 = tk.Button(root,text='Clear', command=clear).grid(row=23,column=1,pady=5)  
  
button3 = tk.Button(root,text='Quit', command=quit).grid(row=23,column=2,padx=3,  
pady=5,sticky='w')  
  
root.mainloop()
```

## Acronyms and Symbols

AER – Annual Equivalent Rate

APY – Annual Percentage Yield

CSV – Comma-Separated Values

DPP – Discounted Payback Period

EIR – Effective Interest Rate

EMI – Equated Monthly Installment

FV – Future Value

GUI – Graphical User Interface

IRR – Internal Rate of Return

MIRR – Modified Internal Rate of Return

NPV – Net Present Value

P – typically used to represent Principal

PV – Present Value

TV – Terminal Value

## References

- [1] *Decimal separator*, Wikipedia, [en.wikipedia.org/wiki/Decimal\\_separator](https://en.wikipedia.org/wiki/Decimal_separator), accessed on 19 December 2020.
- [2] Brown, R., Kopp, S., *Financial Mathematics: Theory and Practice*, McGraw-Hill Ryerson Limited, 2012.
- [3] Broverman, S., *Mathematics of Investment and Credit*, ACTEX Publications, 2010.
- [4] Alhabeeb, M.J., *Mathematical Finance*, John Wiley & Sons, Inc., 2012.
- [5] *Fisher equation*, Wikipedia, [en.wikipedia.org/wiki/Fisher\\_equation](https://en.wikipedia.org/wiki/Fisher_equation), accessed on 21 December 2020.
- [6] Peterson Drake, P., Fabozzi, F., *The Basics of Finance: An Introduction to Financial Markets, Business Finance, and Portfolio Management*, John Wiley & Sons, Inc., 2012.
- [7] *Cost of capital*, Wikipedia, [en.wikipedia.org/wiki/Cost\\_of\\_capital](https://en.wikipedia.org/wiki/Cost_of_capital), accessed on 27 August 2020.
- [8] Bennett, T., *What is a cash flow statement? - MoneyWeek Investment Tutorials*, YouTube Video, <https://youtu.be/GkGdlgX3xYI>, accessed 16 August 2020.
- [9] Tuovila, A., *Cash Flow*, Investopedia article and embedded video, <https://www.investopedia.com/terms/c/cashflow.asp>, accessed 16 August 2020.
- [10] *Net present value*, Wikipedia, [en.wikipedia.org/wiki/Net\\_present\\_value](https://en.wikipedia.org/wiki/Net_present_value), accessed 17 August 2020.
- [11] *Internal rate of return*, Wikipedia, [en.wikipedia.org/wiki/Internal\\_rate\\_of\\_return](https://en.wikipedia.org/wiki/Internal_rate_of_return), accessed 21 August 2020.
- [12] Kelleher, J.C., and Justin J. MacCormack, J.J., *Internal rate of return: A cautionary tale*, McKinsey Quarterly, April 2004. [Document available at <https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/internal-rate-of-return-a-cautionary-tale>]
- [13] *Modified internal rate of return*, Wikipedia, [en.wikipedia.org/wiki/Modified\\_internal\\_rate\\_of\\_return](https://en.wikipedia.org/wiki/Modified_internal_rate_of_return), accessed on 26 August 2020.
- [14] *Payback Period*, Investopedia, <https://www.investopedia.com/terms/p/paybackperiod.asp>, accessed 30 December 2020.
- [15] *Discounted Payback Period*, Investopedia, <https://www.investopedia.com/terms/d/discounted-payback-period.asp>, accessed 30 December 2020.
- [16] Chen, J., *Profitability Index*, Investopedia, <https://www.investopedia.com/terms/p/profitability.asp>, accessed on 25 August 2020.

## Index of Terms

Accumulated value of an annuity, 28  
Amortization, 44  
Annual percentage yield, 20  
Annuity, 28  
Annuity certain, 28  
Annuity-due, 28, 33  
Balloon loan, 17  
Bond, 56  
Callable bond, 56  
Clean price of a bond, 65  
Contingent annuity, 28  
Continuous annuity, 37  
Cost of capital, 71  
Coupon rate, 56  
Deferred annuity, 28  
Dirty price of a bond, 63  
Discount, 21  
Discounted Payback Period, 83  
Discounted value of an annuity, 28  
Effective rate of interest, 19  
Face value of a bond, 56  
Focal date, 24

Forborne annuity, 35  
General annuity, 28  
Internal rate of return, 79  
Issue price, 56  
Marketable bonds, 57  
Maturity date of a bond, 56  
Modified internal rate of return, 82  
Net present value, 71  
Nominal interest rate, 17  
Ordinary annuity, 28  
Payback Period, 83  
Perpetuity, 37  
Present value, 21  
Profitability index, 84  
Real rate of return, 26  
Redeemed at par, 56  
Redemption value of a bond, 56  
Savings bonds, 57  
Simple annuity, 28  
Term of annuity, 28  
Yield of a bond, 56  
Zero-coupon bond, 56