Principles of Financial Computing

Prof. Yuh-Dauh Lyuu
Dept. Computer Science & Information Engineering
and
Department of Finance
National Taiwan University

References

- Yuh-Dauh Lyuu. *Financial Engineering & Computation: Principles, Mathematics, Algorithms*. Cambridge University Press. 2002.
- Official Web page is

  `www.csie.ntu.edu.tw/~lyuu/finance1.html`
- Check

  `www.csie.ntu.edu.tw/~lyuu/capitals.html`

  for some of the software.

Useful Journals

- *Applied Mathematical Finance.*
- *Finance and Stochastics.*
- *Financial Analysts Journal.*
- *Journal of Computational Finance.*
- *Journal of Derivatives.*
- *Journal of Economic Dynamics & Control.*
- *Journal of Finance.*
- *Journal of Financial Economics.*
- *Journal of Fixed Income.*

Useful Journals (continued)

- *Journal of Futures Markets.*
- *Journal of Financial and Quantitative Analysis.*
- *Journal of Portfolio Management.*
- *Journal of Real Estate Finance and Economics.*
- *Management Science.*
- *Mathematical Finance.*

Useful Journals (concluded)

- *Quantitative Finance.*
- *Review of Financial Studies.*
- *Review of Derivatives Research.*
- *Risk Magazine.*
- *Stochastics and Stochastics Reports.*

Introduction

A Very Brief History of Modern Finance

- 1900: Ph.D. thesis *Mathematical Theory of Speculation* of Bachelier (1870–1946).
- 1950s: modern portfolio theory (MPT) of Markowitz.
- 1960s: the Capital Asset Pricing Model (CAPM) of Treynor, Sharpe, Lintner (1916–1984), and Mossin.
- 1960s: the efficient markets hypothesis of Samuelson and Fama.
- 1970s: theory of option pricing of Black (1938–1995) and Scholes.
- 1970s–present: new instruments and pricing methods.

A Very Brief and Biased History of Modern Computers

- 1930s: theory of Gödel (1906–1978), Turing (1912–1954), and Church (1903–1995).
- 1940s: first computers (Z3, ENIAC, etc.) and birth of solid-state transistor (Bell Labs).
- 1950s: Texas Instruments patented integrated circuits; Backus (IBM) invented FORTRAN.
- 1960s: Internet (ARPA) and mainframes (IBM).
- 1970s: relational database (Codd) and PCs (Apple).
- 1980s: IBM PC and Lotus 1-2-3.
- 1990s: Windows 3.1 (Microsoft) and World Wide Web (Berners-Lee).

What This Course Is About

- Financial theories in pricing.
- Mathematical backgrounds.
- Derivative securities.
- Pricing models.
- Efficient algorithms in pricing financial instruments.
- Research problems.
- Finding your thesis directions.

Outstanding U.S. Debts (bln)

Year	Municipal	Treasury	Mortgage—related	U.S. corporate	Fed agencies	Money market	Asset—backed	Total
85	859.5	1,437.7	372.1	776.5	293.9	847.0	0.9	4,587.6
86	920.4	1,619.0	534.4	959.6	307.4	877.0	7.2	5,225.0
87	1,010.4	1,724.7	672.1	1,074.9	341.4	979.8	12.9	5,816.2
88	1,082.3	1,821.3	772.4	1,195.7	381.5	1,108.5	29.3	6,391.0
89	1,135.2	1,945.4	971.5	1,292.5	411.8	1,192.3	51.3	7,000.0
90	1,184.4	2,195.8	1,333.4	1,350.4	434.7	1,156.8	89.9	7,745.4
91	1,272.2	2,471.6	1,636.9	1,454.7	442.8	1,054.3	129.9	8,462.4
92	1,302.8	2,754.1	1,937.0	1,557.0	484.0	994.2	163.7	9,192.8
93	1,377.5	2,989.5	2,144.7	1,674.7	570.7	971.8	199.9	9,928.8
94	1,341.7	3,126.0	2,251.6	1,755.6	738.9	1,034.7	257.3	10,505.8
95	1,293.5	3,307.2	2,352.1	1,937.5	844.6	1,177.3	316.3	11,228.5
96	1,296.0	3,459.7	2,486.1	2,122.2	925.8	1,393.9	404.4	12,088.1
97	1,367.5	3,456.8	2,680.2	2,346.3	1,022.6	1,692.8	535.8	13,102.0
98	1,464.3	3,355.5	2,955.2	2,666.2	1,296.5	1,978.0	731.5	14,447.2
99	1,532.5	3,281.0	3,334.2	3,022.9	1,616.5	2,338.2	900.8	16,026.4
00	1,567.8	2,966.9	3,564.7	3,372.0	1,851.9	2,661.0	1,071.8	17,056.1
01	1,688.4	2,967.5	4,125.5	3,817.4	2,143.0	2,542.4	1,281.1	18,565.3
02	1,783.8	3,204.9	4,704.9	3,997.2	2,358.5	2,577.5	1,543.3	20,170.1

What This Course Is *Not* About

- How to program.
- Basic mathematics in calculus, probability, and algebra.
- Details of the financial markets.
- How to be rich.
- How the markets will perform tomorrow.

Analysis of Algorithms

It is unworthy of excellent men
to lose hours like slaves
in the labor of computation.
— Gottfried Wilhelm Leibniz (1646–1716)

Computability and Algorithms

- Algorithms are precise procedures that can be turned into computer programs.
- Uncomputable problems.
  - Does this program have infinite loops?
  - Is this program bug free?
- Computable problems.
  - Intractable problems.
  - Tractable problems.

Complexity

- Start with a set of basic operations which will be assumed to take one unit of time.
- The total number of these operations is the total work done by an algorithm (its computational complexity).
- The space complexity is the amount of memory space used by an algorithm.
- Concentrate on the abstract complexity of an algorithm instead of its detailed implementation.
- Complexity is a good guide to an algorithm's *actual* running time.

Asymptotics

- Consider the search algorithm on p. 15.
- The worst-case complexity is n comparisons (why?).
- There are operations besides comparison.
- We care only about the asymptotic growth rate not the exact number of operations.
  - So the complexity of maintaining the loop is subsumed by the complexity of the body of the loop.
- The complexity is hence $O(n)$.

Algorithm for Searching an Element

```
for k = 1, 2, 3, ..., n do
    if x = A_k then
        return k;
    end if
end for
return not-found;
```

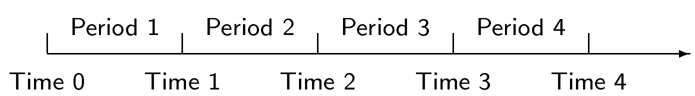

Common Complexities

- Let n stand for the "size" of the problem.
  - Number of elements, number of cash flows, etc.
- Linear time if the complexity is $O(n)$.
- Quadratic time if the complexity is $O(n^2)$.
- Cubic time if the complexity is $O(n^3)$.
- Exponential time if the complexity is $2^{O(n)}$.
- Superpolynomial if the complexity is less than exponential but higher than polynomials, say $2^{O(\sqrt{n})}$.
- It is possible for an exponential-time algorithm to perform well on "typical" inputs.

Basic Financial Mathematics

In the fifteenth century mathematics was mainly concerned with questions of commercial arithmetic and the problems of the architect.
— Joseph Alois Schumpeter (1883–1950)

The Time Line



Time Value of Money

$$\begin{aligned} \text{FV} &= \text{PV}(1+r)^n, \\ \text{PV} &= \text{FV} \times (1+r)^{-n}. \end{aligned}$$

- FV (future value).
- PV (present value).
- r: interest rate.

Periodic Compounding

If interest is compounded m times per annum,

$$\text{FV} = \text{PV}\left(1 + \frac{r}{m}\right)^{nm}. \tag{1}$$

Common Compounding Methods

- Annual compounding: $m = 1$.
- Semiannual compounding: $m = 2$.
- Quarterly compounding: $m = 4$.
- Monthly compounding: $m = 12$.
- Weekly compounding: $m = 52$.
- Daily compounding: $m = 365$.

Easy Translations

- An interest rate of r compounded m times a year is "equivalent to" an interest rate of r/m per $1/m$ year.
- If a loan asks for a return of 1% per month, the annual interest rate will be 12% *with monthly compounding*.

Example

- Annual interest rate is 10% compounded twice per annum.
- Each dollar will grow to be
$$[\,1 + (0.1/2)\,]^2 = 1.1025$$
one year from now.
- The rate is equivalent to an interest rate of 10.25% compounded once *per annum*.

Continuous Compounding

- Let $m \to \infty$ so that
$$\left(1 + \frac{r}{m}\right)^m \to e^r$$
in Eq. (1) on p. 21.
- Then
$$\text{FV} = \text{PV} \times e^{rn},$$
where $e = 2.71828\ldots$.

Continuous Compounding (concluded)

- Continuous compounding is easier to work with.
- Suppose the annual interest rate is r_1 for n_1 years and r_2 for the following n_2 years.
- Then the FV of one dollar will be
$$e^{r_1 n_1 + r_2 n_2}.$$