

THE EXPERT'S VOICE® IN C++

Options and Derivatives Programming in C++

Algorithms and Programming
Techniques for the Financial Industry

Carlos Oliveira

Apress®

Options and Derivatives Programming in C++

Algorithms and Programming Techniques
for the Financial Industry



Carlos Oliveira

Apress®

Options and Derivatives Programming in C++

Carlos Oliveira
Monmouth Junction, New Jersey
USA

ISBN-13 (pbk): 978-1-4842-1813-6
DOI 10.1007/978-1-4842-1814-3

ISBN-13 (electronic): 978-1-4842-1814-3

Library of Congress Control Number: 2016954432

Copyright © 2016 by Carlos Oliveira

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director: Welmoed Spahr

Lead Editor: Jonathan Gennick

Technical Reviewer: Don Reamey

Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan,

Jonathan Gennick, Todd Green, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal,

James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing

Coordinating Editor: Jill Balzano

Copy Editor: Kezia Endsley

Compositor: SPi Global

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springer.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Printed on acid-free paper

To my family, my real source of inspiration.

Contents at a Glance

About the Author	xiii
About the Technical Reviewer	xv
Introduction	xvii
■ Chapter 1: Options Concepts	1
■ Chapter 2: Financial Derivatives.....	19
■ Chapter 3: Basic Algorithms.....	35
■ Chapter 4: Object-Oriented Techniques	67
■ Chapter 5: Design Patterns for Options Processing	85
■ Chapter 6: Template-Based Techniques	101
■ Chapter 7: STL for Derivatives Programming	115
■ Chapter 8: Functional Programming Techniques.....	127
■ Chapter 9: Linear Algebra Algorithms	143
■ Chapter 10: Algorithms for Numerical Analysis	161
■ Chapter 11: Models Based on Differential Equations	175
■ Chapter 12: Basic Models for Options Pricing.....	189
■ Chapter 13: Monte Carlo Methods	207
■ Chapter 14: Using C++ Libraries for Finance	223
■ Chapter 15: Credit Derivatives.....	241
Index.....	255

Contents

About the Author	xiii
About the Technical Reviewer	xv
Introduction	xvii
■ Chapter 1: Options Concepts	1
Basic Definitions	2
Option Greeks.....	6
Using C++ for Options Programming.....	8
Availability	8
Performance	9
Standardization	9
Expressiveness.....	10
Modeling Options in C++	10
Creating Well-Behaving Classes	10
Computing Option Value at Expiration	12
Complete Listing.....	13
Building and Testing	16
Further References.....	18
Conclusion.....	18
■ Chapter 2: Financial Derivatives.....	19
Models for Derivative Pricing	19
Credit Default Swaps	21
Collateralized Debt Obligations.....	22

FX Derivatives.....	23
Derivative Modeling Equations	23
Numerical Models.....	24
Binomial Trees	24
Simulation Models.....	25
Using the STL	26
Generating a Random Walk	27
Complete Listing.....	29
Building and Testing	32
Further References.....	33
Conclusion.....	34
■ Chapter 3: Basic Algorithms.....	35
Date and Time Handling	35
Date Operations	36
Complete Listings	39
A Compact Date Representation	48
Complete Listings	49
Building and Testing	53
Working with Networks	53
Creating a Dictionary Class	54
Calculating a Shortest Path	56
Complete Listings	58
Building and Testing	66
Conclusion.....	66
■ Chapter 4: Object-Oriented Techniques	67
OO Programming Concepts	67
Encapsulation	69
Inheritance.....	72
Polymorphism.....	72
Polymorphism and Virtual Tables.....	75

Virtual Functions and Virtual Destructors	76
Abstract Functions.....	78
Building Class Hierarchies.....	80
Object Composition.....	82
Conclusion.....	83
■ Chapter 5: Design Patterns for Options Processing	85
Introduction to Design Patterns.....	86
The Factory Method Design Pattern.....	87
The Singleton Pattern.....	90
Clearing House Implementation in C++	91
The Observer Design Pattern	93
Complete Code	96
Conclusion.....	100
■ Chapter 6: Template-Based Techniques	101
Introduction to Templates.....	101
Compilation-Time Polymorphism	103
Template Functions	104
Implementing Recursive Functions.....	106
Recursive Functions and Template Classes	108
Containers and Smart Pointers	109
Avoiding Lengthy Template Instantiations.....	111
Pre-Instantiating Templates.....	111
Conclusion.....	113
■ Chapter 7: STL for Derivatives Programming	115
Introduction to Algorithms in the STL.....	115
Sorting.....	116
Presenting Frequency Data	119
Copying Container Data.....	121
Finding Elements.....	123

Selecting Option Data.....	124
Conclusion.....	126
■ Chapter 8: Functional Programming Techniques.....	127
Functional Programming Concepts	128
Function Objects	128
Functional Predicates in the STL.....	131
The Bind Function	133
Lambda Functions in C++11	135
Complete Code	137
Conclusion.....	142
■ Chapter 9: Linear Algebra Algorithms	143
Vector Operations.....	143
Scalar-to-Vector Operations	144
Vector-to-Vector Operations	146
Matrix Implementation	148
Using the uBLAS Library.....	154
Complete Code	156
Conclusion.....	160
■ Chapter 10: Algorithms for Numerical Analysis.....	161
Representing Mathematical Functions.....	161
Using Horner’s Method	163
Finding Roots of Equations.....	165
Newton’s Method.....	165
Integration	170
Conclusion.....	173
■ Chapter 11: Models Based on Differential Equations	175
General Differential Equations.....	175
Ordinary Differential Equations	176

Euler’s Method	177
Implementing the Method	178
The Runge-Kutta Method	182
Runge-Kutta Implementation	183
Complete Code	185
Conclusion.....	187
■ Chapter 12: Basic Models for Options Pricing.....	189
Lattice Models.....	189
Binomial Model	190
Binomial Model Implementation.....	192
Pricing American-Style Options.....	195
Solving the Black-Scholes Model.....	197
Numerical Solution of the Model	198
Complete Code	202
Conclusion.....	205
■ Chapter 13: Monte Carlo Methods.....	207
Introduction to Monte Carlo Methods.....	207
Random Number Generation	208
Probability Distributions	210
Using Common Probability Distributions	213
Creating Random Walks	218
Conclusion.....	221
■ Chapter 14: Using C++ Libraries for Finance	223
Boost Libraries	223
Installing Boost.....	225
Solving ODEs with Boost	225
Solving a Simple ODE	227

The QuantLib Library	229
Handling Dates	230
Working with Calendars.....	231
Computing Solutions for Black-Scholes Equations	233
Creating a C++ Interface.....	235
Complete Code	237
Conclusion.....	240
■ Chapter 15: Credit Derivatives.....	241
Introduction to Credit Derivatives.....	241
Modeling Credit Derivatives	242
Using Barrier Options	243
A Solver Class for Barrier Options	244
Barrier Option Classes in QuantLib.....	245
An Example Using QuantLib.....	247
Complete Code	249
Conclusion.....	253
Index.....	255

About the Author



Carlos Oliveira works in the area of quantitative finance, with more than 10 years of experience in creating scientific and financial models in C++. During his career, Carlos has developed several large-scale applications for financial companies such as Bloomberg L.P. and Incapital LLC. Oliveira obtained a PhD in Operations Research and Systems Engineering from the University of Florida, an MSc in Computer Science from UFC (Brazil), and a BSc in Computer Science from UECE (Brazil). He has also performs academic research in the field of combinatorial optimization, with applications in diverse areas such as finance, telecommunications, computational biology, and logistics. Oliveira has written more than 30 academic papers on optimization and authored three books, including *Practical C++ Financial Programming* (Apress, 2015).

About the Technical Reviewer



Don Reamey is currently a Senior Engineering Manager at Apptio Corporation. Previously he worked at Microsoft as a Principal Software Engineer and Program Manager. Don also managed engineering teams at TIBCO Software and Bank of America. He enjoys learning and creating new programming languages, creating web-based applications, and playing guitar and weight lifting.

Introduction

On Wall Street, the use of algorithmic trading and other computational techniques has skyrocketed in the last few years, as can be seen from the public interest in automated trading as well as the profits generated by these strategies. This growing trend demonstrates the importance of using software to analyze and trade markets in diverse areas of finance. One particular area that has been growing in importance during the last decade is options and derivatives trading.

Initially used only as a niche investment strategy, derivatives have become one of the most common instruments for investors in all areas. Likewise, the interest in automated trading and analysis of such instruments has also increased considerably.

Along with scientists and economists, software engineers have also greatly contributed to the development of advanced computational techniques using financial derivatives. Such techniques have been used at banks, hedge funds, pension funds, and other financial institutions. In fact, every day new systems are developed to give a trading advantage to the players in this industry.

This book attempts to provide the basic programming knowledge needed by C++ programmers working with options and derivatives in the financial industry. This is a hands-on book for programmers who want to learn how C++ is used to develop solutions for options and derivatives trading. In the book's chapters, you'll explore the main algorithms and programming techniques used in the implementation of systems and solutions for trading options and other derivatives.

Because of stringent performance characteristics, most of these trading systems are developed using C++ as the main implementation language. This makes the topic of this book relevant to everyone interested in programming skills used in the financial industry in general.

In *Options and Derivatives Programming in C++*, I cover the features of the language that are more frequently used to write financial software for options and derivatives. These features include the STL, templates, functional programming, and support for numerical libraries. New features introduced in the latest updates of the C++ standard are also covered, including additional functional techniques such as lambda functions, automatic type detection, custom literals, and improved initialization strategies for C++ objects.

I also provide how-to examples that cover the major tools and concepts used to build working solutions for quantitative finance. The book teaches you how to employ advanced C++ concepts as well as the basic building libraries used by modern C++ developers, such as the STL, Boost, and QuantLib. It also discusses how to create correct and efficient applications, leveraging knowledge of object-oriented and template-based programming. I assume only a basic knowledge of C and C++ and extensively use concepts already mastered by developers who are fluent in these languages.

In the process of writing this book, I was also concerned with providing value for readers who are trying to use their current programming knowledge in order to become proficient at the style of programming used in large banks, hedge funds, and other investment institutions. Therefore, the topics covered in the book are introduced in a logical and structured way. Even novice programmers will be able to absorb the most important topics and competencies necessary to develop C++ for the problems occurring on the analysis of options and other financial derivatives.

Audience

This book is intended for readers who already have a working knowledge of programming in C, C++, or another mainstream language. These are usually professionals or advanced students in computer science, engineering, and mathematics, who have interest in learning about options and derivatives programming using the C++ language, for personal or for professional reasons. The book is also directed at practitioners of C++ programming in financial institutions, who would use the book as a ready-to-use reference of software development algorithms and best practices for this important area of finance.

Many readers are interested in a book that would describe how modern C++ techniques are used to solve practical problems arising when considering options on financial instruments and other derivatives. Being a multi-paradigm language, C++ usage may be slightly different in each area, so the skills that are useful for developing desktop applications, for example, are not necessarily the same ones used to write high-performance software.

A large part of high-performance financial applications are written in C++, which means that programmers who want to enter this lucrative market need to acquire a working knowledge of specific parts of the language. This book attempts to give developers who want to develop their knowledge effectively this choice, while learning one of the most sought-after and marketable skillsets for modern financial application and high-performance software development.

This book is also targeted at students and new developers who have some experience with the C++ language and want to leverage that knowledge into financial software development. This book is written with the goal of reaching readers who need a concise, algorithms-based strategy, providing basic information through well-targeted examples and ready-to-use solutions. Readers will be able to directly apply the concepts and sample code to some of the most common problems faced regarding the analysis of options and derivative contracts.

What You Will Learn

Here is a sample of topics that are covered in the following chapters:

- Fundamental problems in the options and derivatives market
 - Options market models
 - Derivative valuation problems
 - Trading strategies for options and derivatives
- Pricing algorithms for derivatives
 - Binomial method
 - Differential equations method
 - Black-Scholes model
- Quantitative finance algorithms for options and derivatives
 - Linear algebra techniques
 - Interpolation
 - Calculating roots
 - Numerical solution for PDEs

- Important features of C++ language as used in quantitative financial programming, such as
 - Templates
 - STL containers
 - STL algorithms
 - Boost libraries
 - QuantLib
 - New features of C++11 and C++14

Book Contents

Here is a quick overview of the major topics covered in each chapter.

- **Chapter 1: “Options Concepts.”** An option is a standard financial contract that derives its value from an underlying asset such as a stock. Options can be used to pursue multiple economic objectives, such as hedging against variations on the underlying asset, or speculating on the future price of a stock. Chapter 1 presents the basic concepts of options, including their advantages and challenges. It also explains how options can be modeled using C++. The main topics covered in this chapter are as follows:
 - Basic definitions of options
 - An introduction to options strategies
 - Describing options with Greeks
 - Sample code for options handling
- **Chapter 2: “Financial Derivatives.”** A derivative is a general term for a contract whose price is based on an underlying asset. In the previous decades, the financial industry created and popularized a large number of derivatives. Pricing and trading these derivatives is a large part of the work performed by trading desks throughout the world. Derivatives have been created based on diverse assets such as foreign currency, mortgage contracts, and credit default swaps. This chapter explores this type of financial instrument and presents a few C++ techniques to model specific derivatives. The main topics covered in this chapter are as follows:
 - Credit default swaps
 - Forex derivatives
 - Interest rate derivatives
 - Exotic derivatives

- Chapter 3: “Basic Algorithms.” To become a proficient C++ developer, it is essential to have good knowledge of the basic algorithms used in your application area. Some basic algorithms for tasks such as vector processing, date and time handling, and data access and storage are useful in almost all applications involving options and other financial derivatives. This chapter surveys such algorithms and their implementation in C++, including the following topics:
 - Date and time handling
 - Vector processing
 - Graphs and networks
 - Fast data processing
- Chapter 4: “Object-Oriented Techniques.” For the last 30 years, object-oriented techniques have become the standard for software development. Since C++ fully supports OO programming, it is imperative that you have a good understanding of OO techniques in order to solve the problems presented by options and derivatives. I present summary of what you need to become proficient in the relevant OO techniques used in the financial industry. Some of the topics covered in this chapter are:
 - Problem partitioning
 - Designing solutions using OO strategies
 - OO implementation in C++
 - Reusing OO components
- Chapter 5: “Design Patterns for Options Processing.” Design patterns are a set of common programming practices that can be used to simplify the solution of recurring problems. With the use of OO techniques, design patterns can be cleanly implemented as a set of classes that interact toward the solution of a common goal. In this chapter, you learn about the most common design pattern employed when working with financial options and derivatives, with specific examples. It covers the following topics:
 - The importance of design patterns
 - Factory pattern
 - Visitor pattern
 - Singleton pattern
 - Less common patterns
- Chapter 6: “Template-Based Techniques.” C++ templates allow programmers to write code that works without modification on different data types. Through the careful use of templates, C++ programmers can write code with high performance and low overhead, without the need to employ more computationally expensive object-oriented techniques. This chapter explores a few template-oriented practices used in the solution of options and derivatives-based financial problems.
 - Motivating the use of templates
 - Compile-time algorithms
 - Containers and smart pointers
 - Template libraries

- Chapter 7: “STL for Derivatives Programming.” Modern financial programming in C++ makes heavy use of template-based algorithms. Many of the basic template algorithms are implemented in the standard template library (STL). This chapter discusses the STL and how it can be used in quantitative finance projects, in particular to solve options and financial derivative problems. You will get a clear understanding of how the STL interacts with other parts of the system, and how it imposes a certain structure on classes developed in C++.
 - STL-based algorithms
 - Functional techniques on STL
 - STL containers
 - Smart pointers
- Chapter 8: “Functional Programming Techniques.” Functional programming is a technique that focuses on the direct use of functions as first-class objects. This means that you are allowed to create, store, and call functions as if they were just another variable of the system. Recently, functional techniques in C++ have been greatly improved with the adoption of the new standard (C++11), particularly with the introduction of lambda functions. The following topics are explored in this chapter:
 - Lambdas
 - Functional templates
 - Functions as first-class objects
 - Managing state in functional programming
 - Functional techniques for options processing
- Chapter 9: “Linear Algebra Algorithms.” Linear algebra techniques are used throughout the area of financial engineering, and in particular in the analysis of options and other financial derivatives. Therefore, it is important to understand how the traditional methods of linear algebra can be applied in C++. With this goal in mind, I present a few examples that illustrate how to use some of the most common linear algebra algorithms. In this chapter, you will also learn how to integrate existing LA libraries into your code.
 - Implementing matrices
 - Matrix decomposition
 - Computing determinants
 - Solving linear systems of equations

- Chapter 10: “Algorithms for Numerical Analysis.” Equations are some of the building blocks of financial algorithms for options and financial derivatives, and it is important to be able to efficiently calculate the solution for such mathematical models. In this chapter, you will see programming recipes for different methods of calculating equation roots and integrating functions, along with explanations of how they work and when they should be used. I also discuss numerical error and stability issues that present a challenge for developers in the area of quantitative financial programming.
 - Basic numerical algorithms
 - Root-finding algorithms
 - Integration algorithms
 - Reducing errors in numeric algorithms
- Chapter 11: “Models Based on Differential Equations.” Differential equations are at the heart of many techniques using in the analysis of derivatives. There are several processes for solving and analyzing PDEs that can be implemented in C++. This chapter presents programming recipes that cover aspects of PDE-based option modeling and application in C++. Topics covered include the following:
 - Basic techniques for differential equations
 - Ordinary differential equations
 - Partial difference equations
 - Numerical algorithms for differential equations
- Chapter 12: “Basic Models for Options Pricing.” Options pricing is the task of determining the fair value of a particular option, given a set of parameters that exactly determine the option type. This chapter discusses some of the most popular models for options pricing. They include tree-based methods, such as binomial and trinomial trees. It also discusses the famous Black-Scholes model, which is frequently used as the basis for the analysis of most options and derivative contracts.
 - Binomial trees
 - Trinomial trees
 - Black-Scholes model
 - Implementation strategies
- Chapter 13: “Monte Carlo Methods.” Among other programming techniques for equity markets, Monte Carlo simulation has a special place due to its wide applicability and easy implementation. These methods can be used to forecast prices or to validate options buying strategies, for example. In This chapter provides programming recipes that can be used as part of simulation-based algorithms applied to options pricing.
 - Probability distributions
 - Random number generation
 - Stochastic models for options
 - Random walks
 - Improving performance

- Chapter 14: “Using C++ Libraries for Finance.” Writing good financial code is not an individual task. You frequently have to use libraries created by other developers and integrate them into your own work. In the world of quantitative finance, a number of C++ libraries have been used with great success. This chapter reviews some of these libraries and explains how they can be integrated into your own derivative-based applications. Some of the topics covered include the following:
 - Standard library tools
 - QuantLib
 - Boost math
 - Boost lambda
- Chapter 15: “Credit Derivatives.” Credit derivatives are an increasingly popular type of financial derivative that aims at reducing credit risk—that is, the risk of default posed by contracts established with a counterparty. Credit derivatives can be modeled using some of the tools already discussed for options, although credit derivative have their own peculiarities. This chapter describes how to create the C++ code needed to quantitatively analyze such financial contracts. Here are some of the topics discussed:
 - General concepts of credit derivatives
 - Modeling the problem
 - C++ algorithms for derivative pricing
 - Improving algorithm efficiency

Example Code

The examples given in this book have all been tested on MacOS X using the Xcode 7 IDE. The code uses only standard C++ techniques, so you should be able to build the given examples using any standards-compliant C++ compiler that implements the C++11 standard. For example, gcc is available on most platforms, and Microsoft Visual Studio will also work on Windows.

If you use MacOS X and don't have Xcode installed in your computer yet, you can download it for free from Apple's developer web site at <http://developer.apple.com>.

If you instead prefer to use MinGW on Windows, you can download the MinGW distribution from the web site <http://www.mingw.org>.

Once MinGW is installed, start the command prompt from the MinGW program group in the Start menu. Then, you can type gcc to check that the compiler is properly installed.

To download the source code for all examples in this book, visit the web page of the author at <http://coliveira.net>.