North Carolina State University

QuantTrade: A C++ Quantitative Pricing and Trading Library

Tao Pang & Dendi Suhubdy

March 2, 2016

1 Abstract

Quanttrade is a complete framework for quantitative finance starting from obtaining inputs of datasets from real-time trading systems, pricing them, and also outputting a predesignated trade order. This framework has been inspired by the inability for quantitative traders to modify the inner workings of a software libraries in critical systems such as front desk trading applications, risk evaluating software or central exchange trading engine. Using this framework will ease the transition for a beginner in quantitative finance to build automated pricing and trading engines.

2 Introduction

Automated trading systems are often used with electronic trading in automatedămarket centers includingăelectronic communication networks (ECNs) and alternative trading systems (ATS). Automated trading systems and electronic trading platforms can execute repetitive tasks at speeds with orders of magnitude greater than any human equivalent. Traditional risk controls and safeguards that relied on human judgment and manual speeds that were appropriate to manual and/or floor-based trading environments, must now be automated to evaluate and control automated trading.

3 Software Structure

We break down the software structure to three parts 1) the input class, 2) the pricing class, and 3) the output class. The main reason behind this structure is C++ design of Object-Oriented Programming (OOP) which will allow the breakdown the execution codes in chunk of blocks. This will ease other developers to take away chunks of codes that they will need for specific in-house projects and ease future development of the code.

The input class is the code chunk that interfaces with a data feed stream. This data feed could be a real time data feed such as 1) NYSE and NASDAQ SIP which are used extensively in the industry or 2) Bloomberg data feed which is used in academia and research desks. For educational purposes, one could also choose a dummy data feed for software testing purposes. These arise in derivative software developments where dynamic hedging needs extensive testing before deployment. The output of this class would be data objects that could be then processed by the pricing class.

The pricing class is the code chunk that takes the data from the input class and process it into viable price output. This class uses numerical analysis to allow efficient calculation of partial differential equations (PDEs) to solve derivative pricing problems. The output of this class would be the price points in time for each inputted security.

The output class is the code chunk that obtains the price point processed in the pricing class and process it into a trading decision and execution. The main trading decision will be made by comparing the price point inferred by the pricing class and

4 Input Methodologies

4.1 NYSE Data Feed

NYSE Alerts is a real-time data feed that provides real-time messages regarding certain conditions related to the trading of NYSE-traded securities. These alerts include Security Trading Status data such as Market Imbalances, Trading Halts/Delays, ITS Pre-Opening Indications, Trading Collars, Price Indications and Trading Circuit Breakers.

The Real-Time NYSE BBO feed provides NYSE Quotes (best bid/ask quotations) for all NYSE-traded securities.This is a top of book feed that publishes updates for every event in real time.

The new NYSE Integrated Feed provides a comprehensive order-by-order view of events in the NYSE equities market. This single high-performance product integrates orders and trades in sequence, providing a more deterministic and transparent view of the order book and related activity.

NYSE OpenBook provides a real-time view of the Exchange's limit-order book for all NYSE-traded securities. NYSE OpenBook lets traders see displayed limit-order volume at every bid and offer price, thus providing full depth-of-market view to the traders.

4.2 Bloomberg Data Feed

5 Pricing Methodologies

5.1 Binomial Model

5.2 Monte Carlo Method

5.3 Finite Difference Methods

6 Output Methodologies

7 Securities and Partial Differential Equations

7.1 European options

7.2 Asian options

7.3 Bermuda options

7.4 American options

8 Results

Here we do a comparison with the other quantitative analysis library Quantlib. We choose Quantlib because of its closest proximity of programming language with QuantTrade: 1) both software libraries are written in C++, 2) both tries to optimize the performance of the algorithms.

The main focus on our comparisons would be the computational speed of our algorithms. Here we will show the computational speeds of different pricing models such as 1) plain vanilla European option, 2) barrier options, 3) lookback options, 4) Bermuda options, 5) Asian options, 6) American options.

9 Conclusion Remarks

10 Appendix