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 the input price point, and result in a trade decision. This trade decision will the be executed with the built in Financial Informational Exchange (FIX) and Financial Information Exchange Adapted for Streaming (FAST) class that will send orders through a designated pipeline towards exchanges.

4 Input Methodologies

4.1 NYSE Data Feed

The NYSE Data feed consists of: 1) NYSE Alert 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. 2) The NYSE 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. 3) the 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, 4) 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 NASDAQ Data Feed

The NASDAQ Data feed consists of 1) NASDAQ TotalView-ITCH which provides tick-by-tick details for all displayable orders in the Nasdaq execution system. Also includes Net Order Imbalance Indicator (NOII) data for the Nasdaq Opening and Closing Crosses, 2) NASDAQ Basic which provides best bid and offer quotations from the Nasdaq execution system as well as trade data from the Nasdaq execution system and FINRA/Nasdaq Trade Reporting Facility (TRF), 3) NASDAQ Last Sale facilitates market transparency by offering real-time last sale data from the Nasdaq execution system and FINRA/Nasdaq Trade Reporting Facility (TRF) at a low cost to market data distributors.

4.3 Bloomberg Data Feed

The Bloomberg Market Data Feed (B-PIPE) enables global connectivity to consolidated, normalized market data in real time. Built for the front office and designed for simplicity of use, B-PIPE provides complete coverage of all the same asset classes as the Bloomberg Professionalő service. The resulting real-time data, along with streaming delayed data, can nourish a wide array of Bloomberg applications, as well as third-party, internal proprietary, and non-display (black box) applications, for increased efficiency throughout the front office. It has consolidated access to 35 million instruments across all asset classes, aggregated from 330+ exchanges and 5,000+ contributors, as well as Bloomberg composite tickers and market indices.

5 Pricing Methodologies

5.1 European options

The option pricing framework written in mathematical formula is defined below. Let $S(t)$ denote the price of the stock at time t. Consider a call option granting the holder the right to buy a stock at a fixed price K at a fixed time T in the future, the current time is t = 0. If at time T the stock price $S(T)$ exceeds the strike price K, the holder exercises the option for a profit of $S(T) - K$; if on the other hand $S(T) \leq K$, the option then expires worthless.

$$(S(T) - K)^+ = max 0, S(T) - K$$

For this expectation we could discount it using the factor e^{-rT} with an r a continuously compounded interest rate. We denote the expected present value of our derivative security as

$$E[e^{-rT}(S(T) - K]^+$$

We could write this in terms of a differential equation

$$\frac{dS(t)}{S(t)} = rdt + \sigma dW(t)$$

where W defines a standard Brownian motion.

5.2 Asian options

Let $S_t = S_0 \exp(\sigma W_t + (r - \frac{1}{2}\sigma^2)$ and a bank account returns a continuously-compounded rate of interest r. Consider the derivative which pays

$Y = (\exp(T^{-1} \int_0^T \log(S_u) \mathrm{d}u) - K)^+$ at time T.

$S_0 \exp(-\frac{1}{2}(r + \sigma^2/6)T)N(-d_1 + \sigma\sqrt{T/3}) - Ke^{-rT}F(-d_1)$

where $d_1 = \log(K/S_0 - 1/2(r - \sigma^2)T)/(\sigma\sqrt{T/3})$.

5.3 Bermuda options

Let t_0 be initial time and $t_1, ..., t_M$ be pre-specified exercise dates with $t_0 < t_1 < \ < t_M = T$, the final maturity, and $?t = t_m?t_{m?1}$. Without a loss of generality it is assumed exercise dates are equidistant. To price a Bermudan option, its value is split into two parts, the continuation value and the immediate exercise payoff. At time $t_{m?1}$, the value of $v(x, t_{m?1})$ consists of the continuation value and the early exercise payoff value.An approximated continuation value, assuming the option is not exercised in the current period, is ([look article][1])

$$c(x, t_{m-1}) = \sum_{k=0}^{N-1} Re\left[\phi\left(\frac{k\pi}{b-a}; y|x\right) exp\left(-ik\pi\frac{a}{b-a}\right)\right] V_k(t_m) \tag{1}$$

where

x : be the modeled quantity at t, often the log asset price.

y : be the modeled quantity at T, often the log asset price.

$f(y|x)$: be the probability density function under the pricing measure.

and

$$V_k(t_m) = \frac{2}{b-a}\int_a^b v(y, t_m) cos\left(k\pi\frac{y-a}{b-a}\right) dy \tag{2}$$

5.4 American options

In the Black-Scholes Model or Heston Model, the American option satisfies the same PDE, but with different boundaries.For an American call option $C_A(S, \tau)$, we can therefore write

$$\frac{\partial C_A}{\partial \tau} = +\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C_A}{\partial S^2} + (r-q)S\frac{\partial C_A}{\partial S} - rC_A \tag{3}$$

or(Heston) $C_A(S, v, \tau)$ satisfy

$$\frac{\partial C_A}{\partial \tau} = +\frac{1}{2}vS^2\frac{\partial^2 C_A}{\partial S^2} + \rho\sigma\, vS\frac{\partial^2 C_A}{\partial S \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 C_A}{\partial v^2} - rC_A + (r-q)S\frac{\partial C_A}{\partial S} + \kappa(\theta - v)\,\frac{\partial C_A}{\partial v} \tag{4}$$

where τ is the time until maturity. The PDE holds for $0 \leq \tau < T$, where T is the maturity calendar time, and for $0 < S \leq b(v, \tau)$, where $b(v, ?)$ is the early exercise boundary. Essentially, this means that as long as the stock price is within the early exercise boundary, the American call option behaves like its European counterpart and the PDE holds. Building on the work of [Chiarella and Ziogas(2006)][1], [Tzavalis, and Wang(2003)][2]. approximate the early exercise boundary $b(v, \tau)$ with the log-linear function.

$$b(v, \tau) = exp(b_0(\tau) + b_1(\tau)v) \tag{5}$$

They show the American call is obtained by adding the early exercise premium to the price of the European call

$$C_A = C_E + V \tag{6}$$

where V is the early exercise premium on an American call with strike K and maturity is τ

6 Numerical Solve of Pricing Models

6.1 Binomial Model

The binomial model enables a multi-period view of the underlying asset price as well as the price of the option. In contrast to the Black-Scholes model, which provides a numerical result based on inputs, the binomial model allows for the calculation of the asset and the option for multiple periods along with the range of possible results for each period (see below).

The advantage of this multi-period view is that the user can visualize the change in asset price from period to period and evaluate the option based on making decisions at different points in time. For an American option, which can be exercised at any time before the expiration date, the binomial model can provide insight into when exercising the option may look attractive and when it should be held for longer periods. By looking at the binomial tree of values, one can determine in advance when a decision on exercise may occur. If the option has a positive value, there is the possibility of exercise, whereas if it has a value less than zero, it should he held for longer periods.

6.2 Monte Carlo Method

Monte Carlo methods are based on the analogy between probability and volume. The mathematics of measure formalizes the intuitive notion of probability, associating and event with a set of outcomes and defining the probability of the event to be its volume or measure relative to that of a universe of possible outcomes. Monte Carlo uses this identity in reverse, calculating the volume of a set by interpreting the volume as a probability. In the simplest case, this means sampling randomly from a universe of possible outcomes and taking the fraction of random draws that fall in a given set as an estimate of the set's volume. The law of large number ensures that this estimate converges to the correct value as the number of draws increases.

To solve the problem of option pricing

```
for $i = 1.....n$ do
    generate $Z_i$;
    set $S_i(T) = S(0)exp(|r - frac12\sigma^2T + \sigma\sqrt{T}Z_i$;
    set $C_i = e^{-rT}(S(T) - K)^+$;
end
set $C_n = (C_1 + C_2 + .... + C_n)$
```

Algorithm 1: Monte Carlo framework for calculating option prices

6.3 Finite Difference Methods

The finite difference method provides an accurate and robust schemes for second-order parabolic and first-order hyperbolic partial differential equations in two independent variables, usually called x and t. The first variable x plays the role of a space coordinate and the second variable t plays the role of time. We model the partial differential equations by approximating the derivatives using divided differences. These latter quantities are defined at so-called discrete mesh points. Having motivated FDM in a generic setting we then apply the resulting finite difference schemes to the one-factor Black?Scholes model

An ODE has one independent variable and hence it is conceptually easier to under- stand and to approximate than equations in two or more variables. In particular, we examine a special kind of problem. This is called first-order initial value problems (IVP). They are useful objects of study in their own right and our objective is to approximate them using FDM in order to pave the way for more complex applications later

6.4 Finite Element Methods

The finite element method has its roots in papers by Richard Courant (Courant, 1943) and John Lighton Synge (see Synge, 1952 and 1957). The 1960?s were the golden years of FEM. Engineers started to apply the method to a wide range of applications in structural and civil engineering and fluid dynamics (Hughes, 2000). It was in the late 1960?s that mathematicians started to take an interest in the field and they developed a rigorous foundation for future study and there has since been a rapid growth in the number of mathematics books on the subject. For example, when the current author embarked on FEM very few books was available (Strang and Fix, 1973)

FEM was first applied to equilibrium and time-independent problems, and was later applied to time-dependent problems. More recently, the method is becoming popular in financial engineering applications, as witnessed by the number of articles being published on the subject and the arrival of monographs (see, for example, Topper, 2005).

There are many similarities between the finite element method and the finite difference method. Since they both address the same kinds of issues and problems in financial engineering the reader might be wondering which method to use in general or in a particular context. There is no black and white answer, but we shall try to give some answers.

1. Learning curve: This is steeper with FEM than with FDM. Some people see FEM as a branch of applied functional analysis and they use concepts such as Hilbert and Sobolev spaces, variational formulations and domain triangulation in their work. FDM is easier because it just replaces derivatives by divided differences. FEM has its roots in engineering and structural analysis. It is also extremely useful for integral equations.

2. Accuracy: In theory, higher order accuracy is possible with FEM but we must construct piecewise polynomial spaces of higher degree. We get ?polynomial snaking? effects, which means that the number of sub-intervals where the piecewise basis polynomial is non-zero increases with the degree of the polynomial.

Since FEM is an integral formulation it is better at approximating discontinuous coefficients than FDM.

1. Multi-factor problems: FEM suffers from the same curse of dimensionality as FDM does. Three dimensions is the limit (it would seem), after which things tend to become intractable. A possible cure for this problem is to use Meshless or some form of operator splitting.

2. Domain of integration: FEM is particularly good at modelling problems with irregular domains, while FDM has difficulties with such domains. On the other hand, most problems in financial engineering are defined in boxes and cubes.

7 Output Methodologies

From an equities trading perspective, FAST is more widely used for market data dissemination, where message rates are much higher. FIX is the protocol of choice for interoperability between firms, and often internal systems as well, although different implementations can vary widely in the specific messages and attributes used.

Brokers and trading venues will generally offer order entry via some flavour of FIX, and offer a complementary native binary protocol for the most performance-sensitive clients or specialised features. The FIX interface is often just a wrapper around the native one, with an more limited set of message types and parameters.

A good example of this is the London Stock Exchange, with offers FIX 5.0 for order entry, along with their own low-latency native protocol. For market data they offer a combination of FAST and ITCH, although even using FAST, the full-depth market data feed isn't available to subscribers, and requires ITCH, as described here

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.

Here we will see the computation speed of the finite difference method for the European option and American Put options.

average timing for European option calculation is 1.29e-05 seconds
average timing for American put option Explicit method calculation is 1.248e-05 seconds
average timing for American put Implicit method calculation is 1.198e-05 seconds
average timing for American put option CN method calculation is 1.438e-05 seconds

9 Conclusion Remarks

10 Appendix