OCaml in Quantitative Finance

Authors: Arthur Wayne (asw263)[1], Advay Koranne (ak845)[2],
Maxwell Zweig (maz72)[3], Jack Young (jry28)[4]

Plan:

We plan to have meetings Sundays and Thursdays at 7:00pm in person or on zoom, depending on our respective schedules.

Proposal:

All of the members in our team are extremely interested in the field of Quantitative Finance and are members of the Cornell Quant Fund.[5] Quantitative finance is - in essence - the art of applying mathematical models to financial markets. Jane Street (one of the world's leading trading firms) is known to be the only trading firm to use OCaml and they have not only provided numerous open source projects but have also released specific information about how to use OCaml in the trading industry that we have taken inspiration from. Further, Cornell PhD Yaron Minsky has written numerous articles such as using OCaml in the financial industry that Cornell Quant Fund will use to leverage OCaml's power to become the world's second trading organization to utilize OCaml.

We intend to build an options pricing model using both the Binomial Options Pricing model as well as the Black-Scholes model. The key features will be:

1. An implementation of the Black-Scholes model.
2. An implementation of the Binomial Options Pricing model.
3. Monte-Carlo methods for option pricing.
4. The ability to validate against historical options data.
5. A visualizer to see the payoff curves (if time permits).

Roadmap:

We have divided our project into three different phases: data-loading/preliminary model building, finalization of the models/historical validation, and GUI integration. For MS1,

MS1 (alpha):

One of the key components of any quant project is data collection and pre-processing. During our alpha stage we hope to download historical options pricing data from a source similar to Yahoo Finance (yfinance), and then develop code to parse and read in the data from a CSV file for analysis later in the project.

In tandem with this, it is our goal to develop the backbone of our financial models, which will include rudimentary prototypes of the Black-Scholes and Binomial Pricing Models, as well as a prototype of our own proprietary Cornell Quant Fund.[6] Monte-Carlo pricing simulator. Furthermore, given the complexity of the Black-Scholes model and the numerous mathematical equations and precision required it is critical we properly write specifications for functions and rigorously test them to ensure they are working. For example, the Black-Scholes model requires taking partial derivatives so we will have to not only learn the tools in OCaml to do this but also how to make functions easily readable and well designed so we can piece them all together.

[1] President and Portfolio Manager at Cornell Quant Fund
[2] Blockchain Team Lead at Cornell Quant Fund
[3] Software Engineer at Cornell Quant Fund
[4] Quant Trader at Cornell Quant Fund
[5] Cornell's first and only organization dedicated solely to the divine juxtaposition of Art and Science that is Quantitative Finance.
[6] Join us during our Fall 2022 recruiting season

Demos:

1. Satisfactory: Have the basic documentation and understanding of the Black-Scholes model (i.e be able to do practice examples by hand). Given that the Black-Scholes model relies on numerous mathematical equations the helper functions should be well documented and divided so that all the pieces of the model can all fit together to form the final equation.
2. Good: A basic implementation of the Black Scholes model that uses well documented functions. This implementation should be able to properly price models based on set values and example problems which we will work out by hand and find online.
3. Excellent: A working Black Scholes model that can read data using our data sources and provide an accurate price. This model should be well optimized as time plays a critical role in live options pricing on actual markets.

MS2 (beta):

Assuming we have implemented the Black-Scholes model above accurately and we are able to use actual Options Data we will implement another options pricing model called the Binomial Option Pricing Model. The model relies on a binomial lattice (Tree) and very much like the Black-Scholes model has numerous different mathematical equations.

Demos:

1. Satisfactory: Have the basic documentation and understanding of the Binomial Options pricing model (i.e be able to do practice examples by hand). Much like above we plan to have documentation and divide the functions in a proper manner.
2. Good: A basic implementation of the Binomial Options Pricing model that uses well documented functions. This implementation should be able to properly price models based on set values and example problems which we will work out by hand and find online.
3. Excellent: A working Binomial Options Pricing model that can read data using our data sources and provide an accurate price. This model should be well optimized as time plays a critical role in live options pricing on actual markets.

MS3 (release):

Since we are not certain as to exactly how difficult creating dynamic models such as the Binomial Pricing Model and Monte-Carlo simulations will be in OCaml, we are leaving the full some wiggle room in the final scope of MS3. If the rest of the project turns out to be relatively trivial and easily implemented, we will spend time developing more sophisticated historical tests of our models. For example we can use two week prior options data and test to see if our models accurately prices the models based on the later prices. Furthermore, we can also possibly develop a rudimentary GUI as outlined in points 4 and 5 of our proposal.