NYU Tandon School of Engineering
Department of Finance and Risk Engineering
FRE-GY 6951 – 1, Selected Topics in Financial Engineering - GPU Financial Software Laboratory - Cuda C for GPU's

Instructor: Louis Scott

Email: ls127@nyu.edu

Office Hours: Before class, or by appointment

Course Objectives:

The purpose of this course is to introduce students to massive parallel processing using CUDA C on GPU's. The class will focus on applications of financial models, including Monte Carlo simulation and finite difference solutions. Students should have completed FRE 6811 – Financial Software Laboratory – C++, or be proficient as a C/C++ programmer. Students should also have course work in asset pricing and numerical methods. By the end of the course, students should be proficient at writing CUDA C code to use parallel processing on GPU's to run financial models. Students will learn how to run models on GPU's by making calls from C++ programs and from Python. Students will also learn how to use multi-threading on CPU's to run financial models in parallel across multiple GPU's. AI and machine learning will not be covered, but programming in CUDA C is an important tool for developing models and algorithms that can be used for AI or machine learning.

References:

CUDA Toolkit Documentation v10.0, or v 9.1, downloaded with CUDA C

CUDA C v 9.1 is available on computers in the Bloomberg Lab

(Go to the C:\ProgramData\NVIDIA Corporation drive on a computer in the lab)

- CUDA C Programming Guide
- CUDA C Best Practices Guide
- CUDA Quick Start Guide
- CUDA Math API
- CURAND Library

Kirk, David B., and Wen-mei W. Hwu, *Programming Massively Parallel Processors*: *A Hands-on Approach*, 3rd Edition, Morgan Kaufman, available at https://store.elsevier.com. E-Book is approximately $80. A good textbook for GPU computing, but not required for FRE 6951.

References for the C++ programming language (from FRE 6811)
C++ Language Tutorial
Wikiversity: Introduction to C++

Hull, John C. (2015), *Options, Futures, and Other Derivatives*, 9th ed.

Grading:

There will be 3 problem sets assigned during the term and a major project due at the end of the term. The project must be a financial application and students are encouraged to use this project to examine a financial model from previous coursework, or to examine a financial model for the major thesis. There will be no exams. Completion of the problem sets will be required, and the major project will determine most of the course grade. Students must submit a written proposal for the major project by April 26, 2019, for approval. Students may work together on problem sets, but each student must turn in his or her own work, including code and output. Students are encouraged to submit problem sets and the major project electronically.

Coding Environment:

Microsoft Visual Studio (Professional Edition or the free Community Edition)
CUDA C, downloaded from Nvidia web site (free download)
Nsight Eclipse (free download from Nvidia for Linux Operating System)
Access to computer with Nvidia GPU with CUDA Compute Capability ≥ 3.0
Student accounts for Prince HPC are available (must run batch jobs)
https://wikis.nyu.edu/display/NYUHPC/High+Performance+Computing+at+NYU

SCHEDULE:

Class Meetings are Friday, 2:00 – 4:41 PM, from April 5, 2019, to May 17, 2019

Course Project: Proposal due April 26, 2019
May 21, 2019 due date for completion of project and write-up
Student proposals to be discussed in class

Course Outline:

Topics to be set for each class period

1. Introduction to Cuda C and parallel programming on GPU's

   Reading assignments from CUDA Toolkit Documentation
   CUDA C Programming Guide, Chapters. 1-2.

   Examples of GPU computing applied to financial models
   Moving C/C++ code from CPU to GPU, device functions
   Allocating and managing memory on GPU device
   Getting Started with CUDA C/C++ (see #2, Your First CUDA Program)
   Nvidia's GPU Technology Conference: https://www.nvidia.com/en-us/gtc/

   Scott, L. "A Stochastic Volatility Jump Model for Pricing SPX and SPY Options with Massive Parallel Processing," Working Paper, December 2017, and presentation at Nvidia's GTC 2018. (See course materials)

2. Random Number Generators and Parallel processing for Monte Carlo simulation

   CURAND Library: Programming Guide, July 2018

   L'Ecuyer, P. and R. Simard (2007) "TestU01: A C library for empirical testing of random number generators," *ACM Transactions on Mathematical Software* 33, Number 4, Article 22, 40 pages.

   L'Ecuyer, Pierre (1999) "Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. *Operations Research* 47(1):159-164. http://dx.doi.org/10.1287/opre.47.1.159

   Bradley, T., J. du Toit, M. Giles, R. Tong, P. Woodhams, "Parallelisation Techniques for Random Number Generators," https://www.nag.co.uk/IndustryArticles/gpu_gems_article.pdf

   Andersen, L., "Simple and efficient simulation of the Heston stochastic volatility model," *Journal of Computational Finance* 11 (Spring 2008): 1-42.

   Rukhin, A., J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S.Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," National Institute of Standards and Technology, April 2010. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf

3. Multi-threading and GPU Computing for Monte Carlo Simulation: Applications for Derivative Pricing

   Examples with Hull-White term structure model and stock price simulation

   Hull, J., *Option, Futures, and Other Derivative Securities* 9th Edition, Chapters 29, 31-33.

   See Bloomberg for access to exchange traded and OTC quotes for interest rate markets derivatives. Swaps, caps/floors, swaptions, and exchange traded interest rate options. For trade structures, open the swap manager by typing SWPM and hit GO

   Strategies for organizing code to run on CPU's and GPU's
   Multi-threading on CPU processors versus parallel processing on GPU's
   CUDA C Programming Guide, Chapter 3.
   CUDA C Best Practices Guide, Chapters 1-9.

4. Parallel processing for finite difference algorithms

   Explicit and implicit methods, Thomas' algorithm for solving by the implicit method
   Example: the 3 factor Hull-White model and a 2 factor stochastic volatility model

   Hull, John C. (2015), *Options, Futures, and Other Derivatives*, 9th ed
   - Chapters 13, 21, 26, and 27

   Reference for applying finite difference methods in Financial Engineering: *Pricing Financial Instruments: The Finite Difference Method*, by Domingo Tavella and Curt Randall, 2000, John Wiley & Sons.

Optional Topics:

5. Running functions on a GPU from Python using PyCuda

   https://documen.tician.de/pycuda/

   https://pypi.org/project/pycuda/

6. High Performance Computing (HPC) at NYU on the Prince Cluster

   https://wikis.nyu.edu/display/NYUHPC/High+Performance+Computing+at+NYU
   Accessing the cluster, Nsight Eclipse IDE, Linux operating system, batch runs

7. Parallel processing for Fourier inversion applications

Carr, P., and D. Madan, "Option valuation using the fast Fourier transform," *Journal of Computational Finance* 3 (1999), 463-520.

Duffie, D., J. Pan, and K. Singleton, "Transform Analysis and Asset Pricing for Affine Jump-Diffusions," *Econometrica* 68(Nov. 2000), 1343-76.

Lee, R.W., "Option Pricing by Transform Methods: Extensions, Unification, and Error Control," *Journal of Computational Finance* 7 (Spring 2004), 51-86.

Scott, L. "Pricing Stock Options in a Jump-Diffusion Model with Stochastic Volatility and Interest Rates: Application of Fourier Inversion Methods," *Mathematical Finance* 7 (1997), 345-358.

Scott, L. "Notes on Using Fourier Inversion Methods to Solve a Stochastic Volatility Jump-Diffusion," May 2017, available in Resource for FRE6951, NYU Classes.

Scott, L. "Notes on Using Fourier Inversion Methods to Solve a Stochastic Volatility Jump-Diffusion for Digital Options," May 2017, available in Resource for FRE 6951, NYU Classes.

8. An Overview of Industry Applications on GPU's

Applications of Industry Models on GPU's:

Richard Hayden, Andrey Zhezherun, Oleg Rasskazov (JP Morgan), "Practical aspects of porting Monte Carlo exotic derivative pricing engines to IBM Power 8+ with P100 GPUs," Nvidia Global Technology Conference, May 2017, at http://on-demand.gputechconf.com/gtc/2017/presentation/s7668-Oleg-Rasskazov-porting-monte-carlo-exotic-derivative-pricing-engines-tesla-p100.pdf

Richard Hayden, Andrey Zhezherun, Oleg Rasskazov (JP Morgan), "Juicing Up Ye Olde GPU Monte Carlo Code," Nvidia Global Technology Conference, May 2018 at http://on-demand.gputechconf.com/gtc/2018/presentation/s8802-juicing-up-ye-olde-gpu-monte-carlo-code.pdf

Serguei Issakov (Numerix), "GPU Acceleration of Monte Carlo simulation for Capital Markets & Insurance," GPU Technology Conference, May 2017, at http://on-demand.gputechconf.com/gtc/2017/presentation/s7417-serguei-issakov-GPU-Acceleration-of-Monte-Carlo-simulation-for-Capital-Markets-and-Insurance.pdf

or watch the GTC 2018 presentation at http://on-demand.gputechconf.com/gtc/2018/video/S8587/

Venkat Bala Rafael Nicolas Fermin Cota, “GPU Accelerated Machine Learning for Bond Price Prediction” GPU Technology Conference, March 2018 at http://on-demand.gputechconf.com/gtc/2018/presentation/s8655-gpu-accelerated-machine-learning-for-bond-price-prediction.pdf

View other finance presentations by choosing finance tags at https://on-demand-gtc.gputechconf.com/gtcnew/on-demand-gtc.php?searchByKeyword=&searchItems=&sessionTopic=22,199,214&sessionEvent=&sessionYear=&sessionFormat=&submit=&select=