¡¡¡¡¡¡¡ HEAD ======= ¿¿¿¿¿¿¿
99e13a429f77bd79d7022a28280ad875663020dc

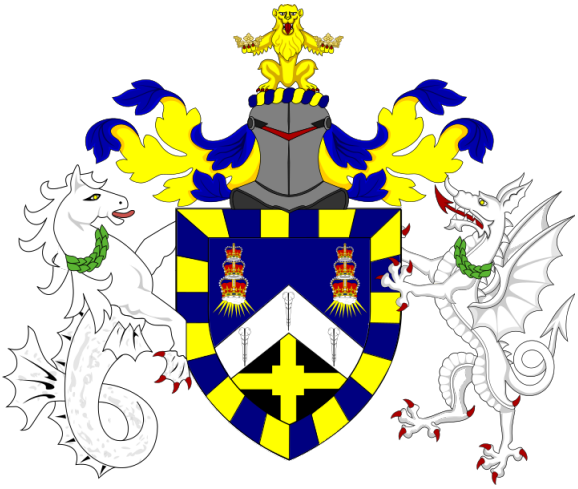
Mathematical Finance MSc Dissertation MTH775P, 2018/19

Disquisitiones Arithmeticæ

High Performance Computing techniques for
numerically solving
financial PDEs

Mustafa Berke Erdis, ID 180883925

Supervisor: Dr. Sebastian del Bano Rollin



A thesis presented for the degree of
Master in Sciences in *Mathematical Finance*

School of Mathematical Sciences
and *School of Economics and Finance*
Queen Mary University of London

Declaration of original work

This declaration is made on July 8, 2019.

Student's Declaration: I, Mustafa Berke Erdis, hereby declare that the work in this thesis is my original work. I have not copied from any other students' work, work of mine submitted elsewhere, or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Referenced text has been flagged by:

1. Using italic fonts, **and**
2. using quotation marks "...", **and**
3. explicitly mentioning the source in the text.

This work is dedicated to my dog Charles Frederick.

Acknowledgements

Here you thank people that have helped you in the journey.
Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Abstract

Here you write a short summary, around 10 lines, of your work.
Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Preface

Here you write a summary of the work. A paragraph on the motivation, previous work, then maybe a brief chapter by chapter summary.

Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Queen Mary University of London
12$^{\text{th}}$ August 2019

Contents

Chapter 1

Introduction

Partial differential equations o

Numerical partial differential equations is a large area of study. The subject includes components in the areas of applications, mathematics and computers. These three aspects of a problem are so strongly tied together that it is virtually impossible to consider the applied aspect of a problem without considering at least some of the mathematical counting aspects of that problem.[j.w. thomas]

The mathematical theory of partial differential equations describing financial markets plays an important role in mathematical finance. In most cases these equations are too complicated to be solved explicitly, therefore different methods of finding an approximate numerical solution is needed.

The most common framework is finite difference which tries to find approximate solutions to the problem at a discrete set of points, normally on a rectangular grid of points. It is simple to construct and analyse but can compromise performance because of increased computational complexity when there are high dimensions. The alternate direction implicit (ADI) method is used to numerically solve two dimensional parabolic PDEs. ADI schemes give us advantages of implicit finite difference method and computationally only requires to solve tridiagonal matrices [1]. Finally, the Monte Carlo

method is used to find the numerical solution when dimensions are too high by calculating an expectation (Feynman-Kac Theorem) [2].

The existing numerical methods for partial differential equations are all constrained by the computational complexity. Motivated by present results and methods employed in high performance computing, we believe there are interesting and challenging topics in numerical solutions of PDEs for finance.

1.1 Parabolic Partial Differential Equations

1.1.1 Heat Equation

1.1.2 Black-Scholes Equation

1.2 Numerical Methods for Parabolic Differential Equations

1.2.1 Explicit Method

1.2.2 Implicit Method

1.2.3 Theta Method and Crank-Nicholson Method

1.2.4 Rannacher Trick

1.2.5 Monte-Carlo Simulation

1.3 Motivation for this work

The idea of this project is to study how to take advantage of this parallelism and explore how much faster we can make these calculations.
Being fast when evaluating new information is crucial for operations of hedge

funds and investment banks. The aim of this project is to utilize High Performance Computing techniques to speed up the existing numerical methods using hardware and software that can be installed in a trading floor. Industry experience is the driver for the project to make an impact.

Theorem 1.3.1 ([P99, Theorem 2.3], see also [BS, pg. 45]). *The Gramm matrix for E_8 is:*

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Recall the theorem of Petri 1.2.1 Look at section **??**.

1.3.1 The problem of numerical solutions

Numerical analysis and computer simulations will be undertaken to put theory and observation together to gain insight into the workings of numerical solutions of partial differential equations.

We plan to develop the methods used for heat equation ($u_t(x,t) = u_{xx}(x,t)$) as our basis point. As we go further into the project, the plan is to extend to Black-Scholes model ($\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = r(V - S\frac{\partial V}{\partial S})$) and variations of Black-Scholes with increasing complexity such as the Multi-Asset Black-Scholes Model and Heston Stochastic Volatility Model.

First step is to write a simple version on a simple framework that can be calculated by hand and with Excel. Following the verifications, next step is porting the simple version in a high level programming language like

python for prototyping and validating all calculations. The penultimate step is moving into a low level programming language such as C++ and utilize high performance computing principles.

High performance computing techniques that can be implemented for CPUs are pipelining and use of SSE/SIMD[3] registers with Advanced Vector Extensions(AVX 512), multithreading with Open Multi-Processing(OpenMP) and compiler intrinsics. In the case of General Purpose GPUs, CUDA or Open Computing Language(OpenCL) can be utilized but can be challenging because of requirement of delicate memory management.

The project will be finalized by comparing the efficiency and speed of different implementations.

1.3.2 Testing the High Performance Computing Techniques

32 bit vs 64 bit

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Compilers VS/gcc/Intel

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl.

Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

AVX/Intrinsics

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Multithreading/OpenMP

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla

egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

CUDA

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

1.3.3 Timing the Code

Windows API

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Chrono Library

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Chapter 2

Work Done Optimizing

2.1 Parallelizing Tridiagonal Systems

Chapter 3

Conclusions

Appendix A

Implementation of the `BarrierOptionCVA` class

Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Appendix B

Additional details on the Gundermanian determinant

Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

Bibliography

[1] Thomas, James W. *Numerical Partial Differential Equations: Finite Difference Methods.*, pp. 164, Springer, 1998.

[2] Klebaner, Fima C. *Introduction to Stochastic Calculus with Applications.*, pp. 155, Imperial College Press, 2005.

[3] Kusswurm, Daniel. *Modern x86 Assembly Language Programming: 32-Bit, 64-Bit, SSE, and AVX.* Apress, 2015.

[P99] William Petri, *Analysis of infinitely generated frog complexes*, Rendicoti Ranæ Analysorum, 234 **(4)**, 34–21, 2015

[Ross] Sheldon Ross, *An Elementary Introduction to Mathematical Finance*, 3rd Edition, Cambridge University Press, 2011

[Hull] John C. Hull, *Options, Futures, and Other Derivatives*, 8th Edition, Pearson Education, 2011

[BS] Fischer Black and Myron Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy 81 **3**, 637–654, (1973)