ADVANCED FINITE DIFFERENCE METHOD (FDM) FOR COMPUTATIONAL FINANCE

(CODE DL-FDM)

DATASIM
EDUCATION BV

CONTENTS

Part 1 Financial and Mathematical Models

In this section we discuss the financial and mathematical PDE models that we approximate using the finite difference method in later sections. We examine the properties of the solution in order to provide insights into the problem at hand. We discuss a number of methods such as the Fichera theory and domain transformation that we shall use in examples in computational finance.

PDE Categories

- One-factor, multi-factor
- Linear, semi-linear, non-linear
- Domain (bounded, semi-infinite, infinite)
- Time-dependent and time-independent PDEs
- Conservative and non-conservative PDE forms
- Reduction to first-order systems

Special Kinds of PDE

- Parabolic and elliptic PDE
- First-order hyperbolic PDE
- 'Asian-style' PDE
- Ordinary differential equations (ODEs)

Describing PDEs

- PDE coefficients
- Boundary conditions (Dirichlet, Neumann, none, linearity)
- Initial conditions
- Well-posedness and continuity
- Energy inequality; existence and uniqueness of solution

Special Properties

- Convection dominance
- Discontinuous initial conditions
- Domain truncation and domain transformation
- Mixed derivatives
- The Fichera theory: Feller conditions

Part 2 Finite Difference Method: Fundamental Techniques

We discuss one-factor PDEs in detail and their approximation by finite difference schemes. We apply these schemes to general convection-diffusion-reaction equations and boundary conditions and we show their application to the one-factor Black Scholes PDE.

We also discuss the numerical analysis of the finite difference method in which we give necessary and sufficient conditions for a finite difference scheme to be stable and to converge to the solution of the PDE that it is approximating. Some methods that we discuss are the Method of Lines (MOL), exponential fitting and the Alternating Direction Explicit (ADE) method.

Attention Points

- Continuous to discrete space: meshes and mesh generation
- Approximation of partial derivatives
- One-step and multistep time marching schemes
- Full discretisaton
- Semi-discretisation and Method of Lines (MOL)

Some Well-known Schemes

- Explicit and implicit Euler
- Crank Nicolson
- Richardson extrapolation
- Alternating Direction Explicit (ADE)
- Monotone schemes and M-matrices

Auxiliary Numerical Methods

- Solution of linear and nonlinear systems
- Interpolation and smoothing
- Numerical integration
- Optimisation (Levenberg-Marquardt, Differential Evolution)
- Eigen and Boost Matrix libraries

Analysis of FDM

- Stability, consistency and convergence
- Conditional and unconditional stability
- Von Neumann stability analysis
- Maximum principle
- Order of accuracy and rate of convergence

Example: One-Factor Black Scholes PDE, I

- Domain truncation versus domain transformation
- Call and put options: boundary conditions
- Payoff functions; handling discontinuities
- Crank Nicolson and Rannacher methods
- Using exponential fitting

Example: One-Factor Black Scholes PDE, II

- Avoiding oscillations: fully implicit method and extrapolation
- Critique of the Crank Nicolson method
- BDF2 and TR-BDF2 schemes
- ADE method for the Black Scholes PDE
- More general cases and Fichera boundary conditions
- Approximating the Greeks (sensitivities)

Numerical Tools

- LU and Cholesky decomposition
- Solution of tridiagonal matrix systems
- Linear and cubic spline interpolation
- Univariate and bivariate normal distributions

Part 3 Advanced (Nonlinear) Models

In this section we introduce a number of linear and nonlinear PDEs and finite difference schemes. In particular, we consider free and moving boundary values problems that describe an option's early exercise features. Since this is a nonlinear problem we see that the methods from Part 2 are not directly applicable. We then resort to nonlinear solvers and transformations to make the problem more tractable.

We discuss the Method of Lines (MOL) in detail. This method reduces a PDE to a system of ordinary differential equations (ODEs) by discretising the underlying space variables only. The resulting ODE system can then be handed to a solver such as Mathematica's NDSolve or the Boost C++ library *odeint*. These libraries are suitable for stiff and non-stiff systems of nonlinear ODEs.

Early Exercise Features

- Free and moving boundaries
- Formulations (fixed domain, front tracking)
- Variational inequalities and PSOR
- Brennan-Schwartz method
- Penalty methods

The Method of Lines (MOL) Overview

- Semi-discretisation
- Vertical MOL and horizontal MOL (Rothe's method)
- Example: one-dimensional heat equation
- Advantages of MOL
- Application areas

MOL in Detail

- Stiff and non-stiff ODEs
- Linear and nonlinear systems
- Incorporating non-Dirichlet boundary conditions into MOL
- Adaptive and non-adaptive ODE solvers

MOL PDE Examples

- Black Scholes
- Cox Ingersoll Ross (CIR)
- Uncertain Volatility Model (UVM)
- CEV model
- Pde for credit value adjustment (CVA)
- MOL in Mathematica and Boost C++ *odeint*

ADE for one-Factor Problems

- Background and motivation
- Saul'yev, Barakat-Clark and Larkin variants
- ADE for convection terms
- Conditional consistency; stability
- Boundary conditions

Other Differential Equations

- Fokker-Planck
- First-time exit PDE
- Riccati ODE

Kinds of Boundary Conditions

- Dirichlet, Neumann, Robin
- Linearity
- PDE on boundary (hyperbolic, parabolic)
- Fichera conditions

Part 4 Two-Factor Models

In this section we discuss several popular finite difference methods to approximate the solutions of the PDEs describing two-factor option pricing. We discuss Alternating Direction Implicit (ADI) method and the method of Fractional Steps ("Soviet Splitting") which originated in the United States and the former Soviet Union in the 1960's, respectively. We apply them to several PDEs in computational finance. Of particular importance is the problem of approximating the mixed derivatives in the PDE to ensure that the resulting scheme is monotone and does not lead to spurious oscillations.

We also discuss MOL and ADE for linear and nonlinear PDEs and we compare them with ADI and splitting methods.

Contenders
- Alternating Direction Implicit (ADI)
- Splitting (Fractional Steps method)
- ADE in two dimensions
- Other methods

The ADI Method
- Using ADI for two-factor PDE
- Mixed derivatives using Craig-Sneyd
- Test cases: basket options and Heston model
- Generalising the ADI method

The Operator Splitting Method
- Yanenko, Marchuk and Strang splittings
- Explicit and implicit splitting
- Handling mixed derivatives and boundary conditions
- Splitting and predictor-corrector methods
- Marchuk 1-2-2-1 model

The ADE Method
- Origins and background; how it differs from ADI and splitting
- Motivating ADE: from heat pde to convection-diffusion and mixed derivatives
- One-sided and centred variants of ADE
- ADE in 3 factors

Comparing ADI, Splitting and ADE Methods
- How they handle mixed derivatives
- Boundary conditions
- Accuracy and robustness of the schemes
- Improving accuracy
- Can the scheme be parallelised?

Mixed Derivatives
- Modeling correlation: extreme cases
- Craig-Sneyd, Verwer, Hout_Welfert, Yanenko
- Stress-testing mixed derivatives
- Test case: compare ADI, splitting and ADE for Heston model

Test Cases
- Basket options
- Heston model
- Asian options
- Anchoring model (Wilmott, Lewis and Duffy)
- Exact solutions

Modelling Jumps
- Merton's and Kou models
- Partial Integro-Differential Equations (PIDE)
- Implicit-explicit Euler method
- Implicit-explicit Runge-Kutta method

Part 5 System Assembly, Software Framework and Stress Testing

In this final part we analyse finite difference schemes to determine their accuracy, efficiency and robustness for a range of input parameters. We need to investigate a number of numerical and computational attention points and we employ whatever tools and methods are needed in order to reach a conclusion. At this stage it may be desirable or even a requirement to view the project as having a software design aspect. For this course, it is an optimisation step but it is useful for front-office and middle-office quants who wish to create software systems from reusable software components.

We provide a number of software tools to promote the quality of the software testing process, including visualisation of results in the C++ *Excel Driver* interface, automatic testing by generating random numbers for parameter input and useful numerical tools such as the ability to interpolate values at non mesh points.

Basic Accuracy Testing (Proof-of-Concept)
- Estimating the local truncation error
- The effects of domain truncation and domain transformation
- Which boundary conditions are optimal?
- Choice of matrix solver

Improving Accuracy and Robustness
- Non-smooth payoffs
- Convection-dominance and exponential fitting
- Computing option sensitivities
- Automatic testing: producing a report

Testing Accuracy: Sparring Partners
- Is there an analytic/quasi-analytic solution to test against?
- Monte Carlo solution
- Binomial method solution

- Two-asset, Heston analytic solution

Using C++11 and C++14

- Random number and distributions library
- *C++ Currency,* parallel computing and tasks
- Loop unrolling
- Using *OpenMP*
- *Boost* and *Eigen*

C++ Software Framework

- Domain Architecture (Duffy)
- Object-oriented, generic and functional programming styles
- A defined and reproducible software architecture
- Variations; incremental development

Your Trainer

Daniel J. Duffy started the company Datasim in 1987 to promote C++ as a new object-oriented language for developing applications in the roles of developer, architect and requirements analyst to help clients design and analyse software systems for Computer Aided Design (CAD), process control and hardware-software systems, logistics, holography (optical technology) and computational finance. He used a combination of top-down functional decomposition and bottom-up object-oriented programming techniques to create stable and extendible applications (for a discussion, see Duffy 2004 where we have grouped applications into domain categories). Previous to Datasim he worked on engineering applications in oil and gas and semiconductor industries using a range of numerical methods (for example, the finite element method (FEM)) on mainframe and mini-computers.

Daniel Duffy has BA (Mod), MSc and PhD degrees in pure and applied mathematics and has been active in promoting partial differential equation (PDE) and finite difference methods (FDM) for applications in computational finance. He was responsible for the introduction of the Fractional Step (Soviet Splitting) method and the Alternating Direction Explicit (ADE) method in computational finance. He is also the originator of the exponential fitting method for time-dependent partial differential equations.

He is also the originator of two very popular C++ online courses (both C++98 and C++11/14) on www.quantnet.com in cooperation with Quantnet LLC and Baruch College (CUNY), NYC. He also trains developers and designers around the world. He can be contacted dduffy@datasim.nl for queries, information and course venues, in-company course and course dates