PDE and FDM in Computational Finance

Daniel J. Duffy (dduffy@datasim.nl)

Goals of Talk

- Overview A-Z of approximating partial differential equations (PDEs) by the finite difference method (FDM)
- Focus on computational finance and 2 factor models (but has wider applicability)
- Model -> **initial boundary value problem** -> FDM scheme -> C++ code
- Improvements to current practice and some original results
- (Unifying, rigorous treatment and standardisation)

Background

- Much of the original research dates from the 1960s (natural gas, fluid dynamics, nuclear)
- USA (ADI method) and Soviet Union (splitting methods)
- Many methods in use in the West were already known in USSR 10-20 years earlier (problem was access to Russian journals in Russian)
- My PhD 1977-1980 built on these results (“problems with small parameter in highest derivative” (later realised it was Black Scholes PDE)

Resources (1)

- “Financial Instrument Pricing Using C++” (Wiley 2018)
- “NUMERICAL METHODS IN COMPUTATIONAL FINANCE A Partial Differential Equation (PDE/FDM) Approach” (Wiley 2021)
- C++ fundamental and advanced C++ courses quantnet.com and Baruch College, NYC
- Datasim site datasim.nl for theses, online courses etc.
- New online course “Intro FDM for beginners and newbies” January 2022 start

Resources (2)

- Davidson, A. and Levin, A. (2014) Mortgage Valuation Methods Oxford University Press. (first to use Marchuk's scheme)
- Sheppard, R. (2007) Pricing Equity Derivatives under Stochastic Volatility: A Partial Differential Equation Approach, Master of Science Thesis University of the Witwatersrand, Johannesburg, South Africa.
- (Yanenko schemes for Heston models)
- Supervisor of MSc student' theses University of Birmingham, UK
- Zuzana Bučková (PhD and research on ADE)

Current "PDE Practice" (My 2-minute Rant) aka attention points for improvement (1)

- P1: PDEs are of linear convection-diffusion-reaction type defined in (semi)infinite domains
- (Domain truncation and trial-and-error boundary conditions)
- P2: PDEs are mostly written in non-conservative form
- (other forms exist and they may be more convenient)
- P3: PDEs hardly ever get transformed to alternative form
- P4: PDE used for equities; less pervasive in fixed interest and hybrid problems (for example, HW, CIR, convertible bonds)

Current “PDE Practice” (2)

- P5: Crank Nicolson, Alternating Direction Implicit (ADI, Craig-Sneyd for mixed derivatives) are the “evergreens”
- P6: FDM “frozen in time” (1970s methods)
- P7: Using divided differences to compute greeks (we have six methods!)
- P8: Mapping from algorithms to code is not documented

Current "PDE Practice" (3)

- " *It is better to solve one problem five different ways, than to solve five problems one way.*" George Pólya
- We don't need an analytic solution (we use it as second option)
- We model a whole range of PDE types with various payoffs by a single (generic) algorithm
- *"The source of all great mathematics is the special case, the concrete example. It is frequent in mathematics that every instance of a concept of seemingly great generality is in essence the same as a small and concrete special case.* "Paul Halmos

So, What’s New?

- “Full lifecycle” of, and support for PDE models in computational finance
- Integration of pure, applied and numerical mathematics
- State-of-art FDM
- Reuse results from other branches of mathematics
- New research by author of book

Specifics (1)

- Proving well-posedness, existence and maximum principles for PDEs
- Defined process (less handwaving) to discover boundary conditions
- Domain transformation and the Fichera/Feller theory for boundary conditions
- Transform a PDE with mixed derivatives to *canonical form*
- Equivalence of PDEs and stochastic equations (SDEs) and probabilistic representations

Specifics (2)

- A range of FD schemes for 1-factor and 2-factor models in time and space
- Crank-Nicolson and Richardson extrapolation
- Alternating Direction Explicit (ADE) (V.K. Saul'yev) (a well-kept secret, see Pealat and Duffy 2011 on ADE for UVM)
- Method of Lines (MOL), flexible but slow in 2 factor problems
- Soviet Splitting (Yanenko, Marchuk, Strang (operator splitting))
- Alternating Direction Implicit (ADI) (Douglas, Rachford)

Specifics (3) (Sensitivities)

- Exact formulae (if you are lucky); divided differences
- Cubic splines
- Automatic Differentiation (AD); Complex Step Method (CSM)
- Continuous Sensitivity Equations (CSE) … a greek can itself be defined as the solution of a PDE
- (See thesis Matt Robinson 2019 on datasim.nl/blogs)
- Something for everybody ☺ (discussed in my August 5 2020 Thalesians' online talk)

Test Cases

- Standard European and American Black Scholes equation
- Multi-asset/basket options
- Asian-style (Average, Cheyette, Anchoring model)
- Interest rate (CIR, HW, HW2)
- Commodities (spread options)
- (Convertible bonds)

Structure of Book

- 26 chapters in 5 major blocks
- Part A: Mathematical Foundations and 1-Factor Models
- B: Mathematical Foundations for 2-Factor Models
- C: Foundations of FDM
- D: Advanced FDM
- E: Test Cases in Computational Finance

Numeric Nitty Gritty: Current Practice..

- Based on a number of ("not even wrong") assumptions
- 1. The von Neumann Fourier stability analysis only applicable to linear initial value problems
- 2. There is more to life than Crank Nicolson (it is not L-stable)
- (D. Duffy a Critique of the Crank Nicolson method …, Wilmott Magazine July 2004)
- 3. Most schemes are not *monotone*
- 4. Well-posedness discussions not on the radar

Numeric Nitty Gritty: Nirvana

- Formulation of abstract Cauchy problem (continuous semigroup theory, coming soon)
- All splitting methods derived from this theory
- Component (dimensional) splitting versus operator splitting
- Energy inequalities and well-posedness
- Maximum principle and monotone schemes

The Process PDE/FDM

- Devise a plan to set up FDM for PDE
- Can view problem as a Data Flow Diagram in a sense
- “Closing to code” in a sequence of stages
- Decisions to be made at each stage depending on the requirements
- We take one test scenario

The Process PDE/FDM (2)
Splitting for Spread Option PDE

- Two-factor linear convection-diffusion-reaction PDE with mixed derivative term
- Problem defined in a quarter plane
- Transform domain into unit square (results in modified PDE)
- Imposition of boundary conditions mathematically justified (Fichera, Green's formula (integration by parts), energy inequalities)
- Choose PDE in conservative or non-conservative form (more usual)

Two-Factor Model Test Case

$$\frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 f}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 f}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 f}{\partial S_1 \partial S_2} +$$

$$(r - q_1) S_1 \frac{\partial f}{\partial S_1} + (r - q_2) S_2 \frac{\partial f}{\partial S_2} = r\, f - \frac{\partial f}{\partial t}$$

The Process PDE/FDM (3)

- Analysis of Yanenko (Lie-Trotter) and Marchuk Two-Cycle Method
- Both based on splitting on time levels (n, n+1) and (n-1, n+1), respectively
- Notation: differential operators A1 and A2 in x and y directions, respectively
- (full story in my PDE book)

Payoff

$$S(t) = S_1(t) - S_2(t), \quad t \geq 0$$

$$f_T(S) = \max(0, S(T) - K) \; \equiv \; (S(T) - K)^+ \text{ (Call)}$$

$$.$$

$$f_T(S) = \max(0, K - S(T)) \equiv (K - S(T))^+ \text{ (Put)}$$

where in general $a^+ \equiv \max(0, a)$.

PDE (S1, S2)

$$\frac{\partial u}{\partial t} = \tfrac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 u}{\partial S_1^2} + (r - q_1) S_1 \frac{\partial u}{\partial S_1}$$

$$+\tfrac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 u}{\partial S_2^2} + (r - q_2) S_2 \frac{\partial u}{\partial S_2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 u}{\partial S_1 \partial S_2} - ru, \quad 0 < S_1, S_2 < \infty.$$

PDE (x,y) Conservative Form

$$\frac{\partial u}{\partial t} = \tfrac{1}{2}\sigma_1^2 x^2 \tfrac{\partial}{\partial x}\left\{(1-x)^2 \tfrac{\partial u}{\partial x}\right\} + rx(1-x)\tfrac{\partial u}{\partial x}$$

$$+\tfrac{1}{2}\sigma_2^2 y^2 \tfrac{\partial}{\partial y}\left\{(1-y)^2 \tfrac{\partial u}{\partial y}\right\} + ry(1-y)\tfrac{\partial u}{\partial y} + \rho\sigma_1\sigma_2 x(1-x)y(1-y)\tfrac{\partial^2 u}{\partial x \partial y} - ru$$

where:

$$x = \tfrac{S_1}{S_1+\alpha_1}, \quad y = \tfrac{S_2}{S_2+\alpha_2}; \quad 0 < x, y < 1.$$

PDE (x,y) Non-conservative

$$\frac{\partial u}{\partial t} = \tfrac{1}{2}\sigma_1^2 x^2 (1-x)^2 \frac{\partial^2 u}{\partial x^2} + \left\{ rx(1-x) - \sigma_1^2 x^2 (1-x) \right\} \frac{\partial u}{\partial x}$$

$$+\tfrac{1}{2}\sigma_2^2 y^2 (1-y)^2 \frac{\partial^2 u}{\partial y^2} + \left\{ ry(1-y) - \sigma_2^2 y^2 (1-y) \right\} \frac{\partial u}{\partial y}$$

$$+\rho\sigma_1\sigma_2 x(1-x)y(1-y)\frac{\partial^2 u}{\partial x \partial y} - ru$$

Lie-Trotter

$$\frac{\partial u^*}{\partial t} = A_1 u^*, u^*(t_n) \text{ given}$$

$$\frac{\partial u^{**}}{\partial t} = A_2 u^{**}, u^{**}(t_n) = u^*(t_{n+1})$$

$$u(t_{n+1}) = u^{**}(t_{n+1}).$$

Intermezzo: Error Analysis (1)

- Foundation for splitting methods is *Co semigroup theory* (generalisation of the exponential function)
- Abstract Cauchy problem

$$u' \equiv \frac{du}{dt} = Au + f,\ 0 < t \leq T,\ \ u(0) = u_0 \text{ where } f \in C([0,T];X),$$

$$u_0 \in X.$$

- Solution is

$$u(t) = e^{tA}u_0 + \int_0^t e^{(t-s)A} f(s)ds.$$

Error Analysis (2)

- Special example of operator splitting

$$\begin{cases} \frac{du}{dt} = (A_1 + A_2)u, \quad 0 < t \leq T \\ u(0) = u_0 \end{cases}$$

- Solution

$$u(t) = e^{(A_1+A_2)t}u_0, \quad A_1, A_2 \in \mathbb{R}^{n,n}.$$

Error Analysis (3)

- Splitting error

$$\begin{cases} \frac{dv}{dt} = A_1 v, \quad 0 < t \le \tau \\ v(0) = u_0 \\ \frac{dw}{dt} = A_2 w, \quad 0 < t \le \tau \\ w(0) = v(\tau). \end{cases}$$

Error Analysis (4)

- Solution (semigroup property)

$$w(\tau) = e^{A_2\tau} v(\tau) = e^{A_2\tau} e^{A_1\tau} u_0.$$

- Analytical solution

$$u(\tau) = e^{\tau(A_1+A_2)} u_0.$$

Local Splitting Error

$$\text{Error } (\tau) = \exp \tau(A_1 + A_2) - \exp(A_2\tau)\exp(A_1\tau)u_0.$$

- Use Taylor expansion

$$\exp \tau(A_1 + A_2) - \exp(\tau A_2)\exp(\tau A_1) = (A_1A_2 - A_2A_1)\tau^2 + 0(\tau^3).$$

- Punch line: 2nd order accuracy if the operators commute
- (this technique can be applied to all splitting schemes)

Marchuk Scheme

$$\frac{u^{n-1/2}-u^{n-1}}{\triangle t} = A_1^n \frac{u^{n-1/2}+u^{n-1}}{2}$$

$$\frac{u^{n}-u^{n-1/2}}{\triangle t} = A_2^n \frac{u^{n}+u^{n-1/2}}{2}$$

$$\frac{u^{n+1/2}-u^{n}}{\triangle t} = A_2^n \frac{u^{n+1}+u^{n}}{2}$$

$$\frac{u^{n+1}-u^{n+1/2}}{\triangle t} = A_1^n \frac{u^{n+1}+u^{n+1/2}}{2}$$

where $A_j^n \equiv A_j(t_n), \quad j = 1, 2.$

Predictor-Corrector

$$\frac{u^{n+1/4}-u^n}{\triangle t/2} = A_1 u^{n+1/4} \text{ (Predictor)}$$

$$\frac{u^{n+1/2}-u^{n+1/4}}{\triangle t/2} = A_2 u^{n+1/2} \text{ (Predictor)}$$

$$\frac{u^{n+1}-u^n}{\triangle t} = A u^{n+1/2} \text{ (Corrector).}$$

ADE Scheme (2d heat equation)

$$\frac{U^{n+1}_{i,j}-U^{n}_{i,j}}{k} = \frac{1}{h_1^2}\left(U^{n+1}_{i-1,j} - U^{n+1}_{i,j} - U^{n}_{i,j} + U^{n}_{i+i,j}\right) + \frac{1}{h_2^2}\left(U^{n+1}_{i,j-1} - U^{n+1}_{i,j} - U^{n}_{i,j} + U^{n}_{i,j+1}\right)$$

$$\frac{V^{n+1}_{i,j}-V^{n}_{i,j}}{k} = \frac{1}{h_1^2}\left(V^{n}_{i-1,j} - V^{n}_{i,j} - V^{n+1}_{i,j} + V^{n+1}_{i+i,j}\right) + \frac{1}{h_2^2}\left(V^{n}_{i,j-1} - V^{n}_{i,j} + V^{n+1}_{i,j} + V^{n+1}_{i,j+1}\right)$$

$$u^{n}_{i,j} = \frac{1}{2}\left(U^{n}_{i,j} + V^{n}_{i,j}\right)$$

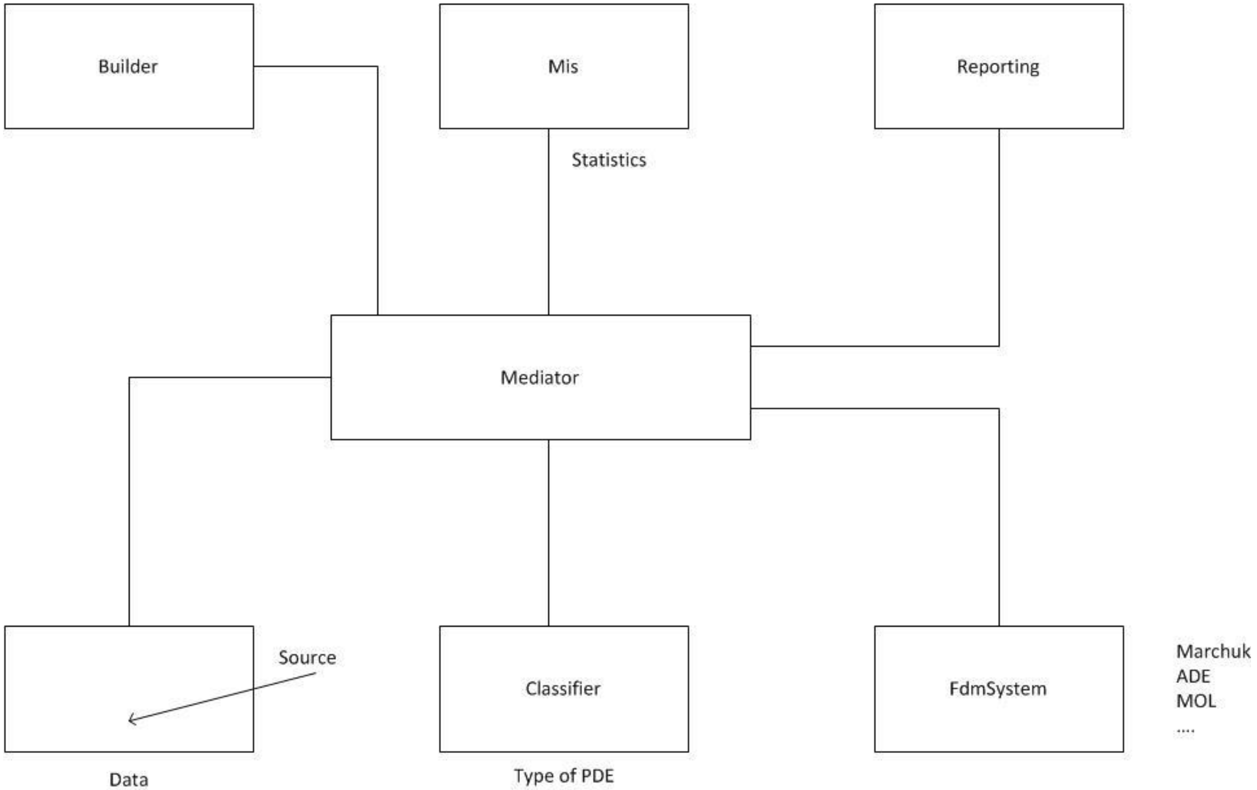
Strang Splitting

- Similar to Marchuk splitting (code very easy to modify)
- We can split PDE into operators with different time scales (e.g. chemical reactions in fluid dynamics)
- Second-order accurate (use Baker-Campbell-Hausdorf f (BCH) formula)
- Q. given X, Y in a Banach algebra, is there a Z s.t. exp(Z) = exp(X) exp(Y)?

Mapping to Code

- How to design efficient and flexible software for these kinds of problems
- Adopt a modular/system decomposition (Domain Architectures (Duffy 2004))
- Single Responsibility Principle (SRP) for each module
- Many ways to implement (polymorphic) behaviour (subtype, CRTP, function pointers, C++20 Concepts, “flat” C functions

Context Diagram



Quality Results
Which FD scheme is best?

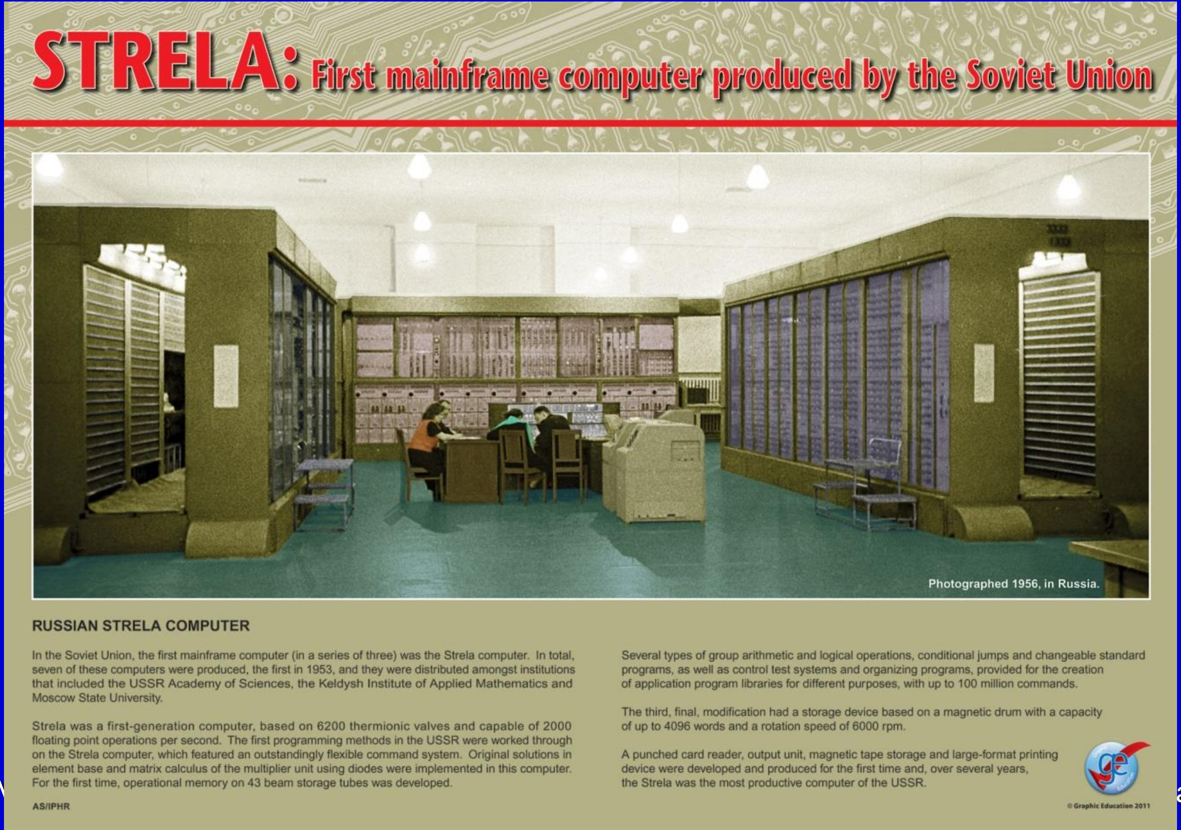
- Quality metrics (accuracy, efficiency, ease of implementation)
- Test: American spread option, large correlation, compare schemes
- All schemes give same answer, but with varying levels of effort
- (Saul'yev ADE, B&C ADE, Yanenko, PredictorCorrector, Marchuk)
- (22, 64, 74, 91, 150) time
- The method from the 1950s wins! (on *Strela* computer (Novosibirsk) machine 10 minutes with h = 1/60 to solve heat equation; fast!)
- On laptop we get 10,000 speedup for spread options

C++ Code Snippet

```
// LR Sweep U
for (std::size_t i = 1; i <= N1 - 2; ++i)
{
       xm = xMesh[i];
       for (std::size_t j = 1; j <= N2 - 2; ++j){
       ym = yMesh[j];
       MNew(i, j) = (MNew(i, j) * (1.0 - A1 - A2)
       +MNew(i - 1, j) * (A1 - B1) + MNew(i + 1, j) * (A1 + B1)
       + MNew(i, j - 1) * (A2 - B2)+ MNew(i, j + 1) * (A2 + B2)
       +YanenkoMixedDerivative(i,j,tnow,MNew)) / (1.0 + A1 + A2 - C);

// Early exercise test
if (early){
       MNew(i, j) =
       std::max<double>(MNew(i, j), ICUnit(xMesh[i], yMesh[j]));}
}
```

Стрела -> спутник 1 (1957)



STRELA: First mainframe computer produced by the Soviet Union

Photographed 1956, in Russia.

RUSSIAN STRELA COMPUTER

In the Soviet Union, the first mainframe computer (in a series of three) was the Strela computer. In total, seven of these computers were produced, the first in 1953, and they were distributed amongst institutions that included the USSR Academy of Sciences, the Keldysh Institute of Applied Mathematics and Moscow State University.

Strela was a first-generation computer, based on 6200 thermionic valves and capable of 2000 floating point operations per second. The first programming methods in the USSR were worked through on the Strela computer, which featured an outstandingly flexible command system. Original solutions in element base and matrix calculus of the multiplier unit using diodes were implemented in this computer. For the first time, operational memory on 43 beam storage tubes was developed.

AS/IPHR

Several types of group arithmetic and logical operations, conditional jumps and changeable standard programs, as well as control test systems and organizing programs, provided for the creation of application program libraries for different purposes, with up to 100 million commands.

The third, final, modification had a storage device based on a magnetic drum with a capacity of up to 4096 words and a rotation speed of 6000 rpm.

A punched card reader, output unit, magnetic tape storage and large-format printing device were developed and produced for the first time and, over several years, the Strela was the most productive computer of the USSR.

Conclusions/Question Time

- Book can serve as foundation for new research (build on a proven basis)
- Use domain transformation and ADE for calibration
- Finite Volume Method (FVM) … integral formulation for PDE problems (ease of use FDM << FVM << FEM)
- PDE for 3 asset spread options (hypothesis … only Saul'yev ADE is feasible/achievable). Use CSE approach for hedge ratios
- Gradient (dynamical) systems … ODEs for Machine Learning