Finite Difference Methods for Financial Problems

Part 0: Introduction

Kwant Skool
Copenhagen
March 2011

Jesper Andreasen
Danske Markets
kwant.daddy@danskebank.com

Outline

- References.
- About the lecturer: CV and work examples.
- Introduction: FD & MC, trees, Greed for speed, course outline …
- Questions?

References

- Andersen, L. and V. Piterbarg (2010): “Interest Rate Modeling: Models, Products, Risk Management.“ Chapter 2, Volume 1.

- Andreasen, J. and B. Huge (2010): “Volatility Interpolation and Finite Difference Based Simulation.” ICBI Global Derivatives Presentation.

- Andreasen, J. (2009): “Planck-Fokker Boundary Conditions.” Danske Markets working paper.

- Andreasen, J. and B. Huge (2010): “Volatility Interpolation.” Danske Markets working paper.

- Dupire, B. (1994): “Pricing with a Smile.” *Risk* July, 18-20.

- Mitchell, A. and D. Griffiths (1980): “The Finite Difference Method in Partial Differential Equations.” John Wiley & Sons.

- Nabben, R. (1999): "On Decay Rates of Tridiagonal and Band Matrices." *SIAM J. Matrix Anal. Appl.* 20, 820-837.

- Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (1988): "Numerical Recipes in C." Cambridge University Press.

- Randall, C. and D. Tavella (2000): "Pricing Financial Instruments – the Finite Difference Method." Wiley.

- Østerby, O. (2002): "The Finite Difference Method for Parabolic Partial Differential Equations." Aarhus University.

About the Lecturer

- My checkered career:

  - 93-97 PhD in mathematics and economics, thesis: Essays on Contingent Claim Pricing.
  - 97 Senior analyst, Bear Stearns, London.
  - 97-00 Vice President, General Re Financial Products, London.
  - 00-02 Principal, Bank of America, London.
  - 02-05 Director, Nordea, Copenhagen
  - 05-08 Managing Director, Bank of America, London
  - 08- K Daddy, Danske Markets, Copenhagen

- Implemented 1-2-3 dimensional (plus time) PDE solvers for complex models for front line trading desk use.

- *Not* a finite difference or numerical methods expert or C++. Never read a book on it.

- Solely gained knowledge from experiments and reading while coding.

FD at Work

- Past examples:
  - 1D and 1D+J solvers for Asian, lookback, passport, and barriers.
  - 1D+J local vol.
  - 1-2D convertible models.
  - 1-2D Gaussian interest rate models.
  - 1.5-2.5D Cheyette interest rate models.
  - 2D stochastic volatility and stochastic local volatility models.
  - 3D PRDC models.
  - 1D soft market feedback models.
  - 1D expansion models.

- Current work:
  - 1D volatility interpolation scheme.
  - 1-2D SV local volatility forward/backward/simulation.
  - 2D volatility interpolation scheme.
  - 3D+ local volatility forward/backward/simulation schemes.

Finite Difference & Monte Carlo

- Today Monte-Carlo (MC) and Finite Difference (FD) methods can solve pretty much the same problems.

- Over recent years technology has been developed for giving very tight lower and upper bounds on American option pricing problems in MC (Longstaff-Schwartz, Andersen, others). However, full control problems (portfolio optimisation) and general non-linear problems are not *yet* fully mastered in MC. But this is just a question of time.

- MC holds a speed advantage over FD for >=4D but FD has the upper hand for <=3D.

- In practice the two methods supplement each other well and typically models are implemented both in FD and in MC implementation.

- FD is used when we can: Bermudans, chooser caps, barriers, American options, PRDC, etc, etc…

- MC is used for strongly path-dependent and/or multi underlying stuff: snowballs, TARNs, Asians, Baskets, Napoleons, etc etc.

- Correctly implemented 1-2 FD schemes are *awesome* in terms of accuracy and speed.

- Examples: you can solve a 30y Bermuda swaption in a 1.5D Cheyette model to trading and risk accuracy on a 50x100x10 ($t \times x \times y$) in around 0.2 seconds.

- *Anything* Black-Scholes can be solved in 0.01 seconds.

Binomial Tree Hugging

- "Trees" are a subset of the general FD toolbox.

- Trees used to be widely spread on Wall Street. But you have to be a hardcore tree hugger to use them today.

- Why?

  - Formally and practically they are inferior to FD (Crank-Nicolson).

  - You will have to do 10,000 time steps in a binomial tree to get the same accuracy as a 100x100 CN grid. *Roughly* 25 times less CPU.

  - They are a real bitch to implement for anything but the simplest models and problems.

  - Even simple things such as barrier options in BS model are troublesome in a tree setting.

- Personally, I have only *once* implemented a tree. I was young and I needed the money.

Greed for Speed

- Why this *greed for speed*?
- Isn’t raw computer muscle the answer?
- No. More power just means that you want to solve more and/or harder problems: more trades, more model complexity.
- So fast and accurate methods are always in demand and the language is C++, not Java, Python, S+, whatever.
- In 1997 GRFP ran all its trades on 1-4 (400MHz) CPUs.
- By 2000 we had assembled the first PC cluster of 32 CPUs.
- Today banks have farms of hundreds and thousands of CPUs. And yes, also Scandi banks,
- …and we are toying with GPUs with 512 processors per PC.

- So even though computers are getting faster and our implementations are getting better, there is still huge expansion in computer power.

- This is driven by more complex products *and* more complex market conditions *and* more sophisticated models.

- And, actually, we are not really more *real-time* today than what we used to be.

- Despite all this talk about auto-trading and quant strategies....

Course Outline

- 1 Dim PDE:

  - Explicit, implicit, Crank-Nicolson, tridag solver, forward equations.
  - Theory: stability and convergence with live examples.
  - How-to-do: transforms, dimensions, boundary conditions, stability and convergence in practice.

- 2-3 Dim PDE:

  - ADI and split methods.
  - Coordinate transforms, correlation, and predictor-corrector methods.

- Applications and examples:

  - Forward/backward/simulation of Dupire's (stochastic) local volatility model.
  - Penta diagonal schemes for the Cheyette model.
  - Jump-diffusion models.

  - Non-linear problems: passport options, uncertain volatility, feedback models.

- To get the full benefit of this course, implement everything. Code it yourselves.

- At Danske our 2D solver (which also does 1D and penta) which runs both forward and backward with two different methods is only around 2000 lines of C++.

- Further, all FD stuff is based on very simple math: linear algebra and Taylor expansions. Only +-*/. No special functions or non-linear schemes.

- So there is actually not that much or very complex code to do.

- But there are still surprisingly many ways to mess it up. My employees continuously develop on this.

- Not before you have gotten your fingers dirty, pulled your hair out, and produced correct numbers on a machine, will you really know what I have been talking about.

- Attempting to learn FD without doing numbers is like trying to learn the breaststroke from a book.

- You are completely fooling yourselves if you think that decorating paper with Greek letters is intellectually superior to producing good numbers on a computer in finite time.

- Questions?