Numerical Methods in Financial Engineering

Finite Difference Methods

José Figueroa-López

Math 516 / Stat 541 Fall, 2008

Contents

1 Finite-Difference Methods

1.1 Introduction.

Overview of the method.

- *Objective:* Find numerical approximations $\widetilde{u}(\cdot)$ for a function $u(\cdot)$, which is the solution of a well-posed PDE with suitable boundary.
- *Basic steps*:
  1. Discretize time and space on a region of interest, leading to a lattice determined by mesh parameters Δt and Δx:
  $$x_n = x_0 + n\Delta x, \quad t_m = t_0 + m\Delta t, \quad n = 0, \ldots, N, \quad m = 0, \ldots, M.$$
  2. Approximate the derivatives $\partial_t u$ and $\partial_x u$ of the PDE at each point of the lattice by some type “finite difference":

- *Backward difference:* $\Delta_h^1 f(y) = \frac{f(y+)-f(y)}{h}$
- *Forward difference:* $\Delta_h^1 f(y) = \frac{f(y)-f(y-h)}{h}$
- *Forward difference:* $\Delta_h^{1/2} f(y) = \frac{f(y+h)-f(y-h)}{2h}$

3. Impose boundary conditions:
   - *Dirichlet*: Conditions for $u(T, x_M)$, $u(t, x_M)$, $u(t, x_0)$
   - *Neuman:* Conditions for $u(T, x_M)$, $\partial_x u(t, x_M)$, $\partial_x u(t, x_0)$
4. The above discretization process transform the PDE into a *system of finite-difference equations*, which solution $\widetilde{u}(t_m, x_n)$ must be determined.

1.2 The explicit method.

1.2.1 Description

The simplest variation: The explicit method.

- Consider the PDE:

$$\partial_t u(t,x) + \mu(t,x)\, \partial_x u(t,x) + a(t,x)\, \partial_{xx} u(t,x) - r\, u(t,x) = 0. \quad (1)$$

- *Discretization:* Using Backward difference in time and centered differences in space, leads to the system of finite-difference equations:

$$\frac{u_n^m - u_n^{m-1}}{\delta t} + \mu_n^m \frac{u_{n+1}^m - u_{n-1}^m}{\delta x} + a_n^m \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2} - r\, u_n^m = 0,$$

  where $\mu_n^m := \mu(t_m, x_n)$, $a_n^m := a(t_m, x_n)$, and $u_n^m := \widetilde{u}(t_m, x_n) \approx u(t_m, x_n)$.

- *Boundary conditions:* $u(x, T) = g(x) \Longrightarrow u_m^N = g(x_N)$.
- *Solving the system:* The special structure of the system allow us to solve it explicitly using Backward induction in time:

$$u_n^{m-1} = p_u^{n,m} \cdot u_{n+1}^m + p_s^{n,m} \cdot u_n^m + p_d^{n,m} \cdot u_{n-1}^m, \quad (2)$$

  with $p_u = \frac{a_n^m \delta t}{(\delta x)^2} + \frac{\mu_n^m \delta t}{2(\delta x)}$, $p_d = \frac{a_n^m \delta t}{(\delta x)^2} - \frac{\mu_n^m \delta t}{2(\delta x)}$, and $p_s = 1 - (\delta t) r - p_d - p_u$.

1.2.2 About stability and consistency.

Technical problems of practical relevance

1. *Desirable properties of the numerical solution*:
   - Efficiency: “How fast is the algorithm?"
   - Stability: “Are there any error propagation whether the error is in the input data or rounding error?."
   - Consistency: “Does the solution converge to what it should approximate as the discretization meshes shrink ?."

2. *Tools to analyze and guarantee the above conditions*:

   - Transform the PDE into a simpler form (say a LINEAR PDE).
   - Employ other finite difference approximations (this will lead to the so-called "implicit methods").

1.2.3 Transformations

Transforming the Black-Scholes equation

1. For numerical reason and stability analysis, it is convenient to transform the BS PDE in a simpler form.

2. Working with log return $x = \log s$ and discounted option prices $e^{-rt}v(t,x)$: If $v(t,s)$ is the *option price at time t when the spot asset price is s*,

$$\partial_t v(t,s) + rs\partial_s v(t,s) + \frac{\sigma^2}{2}s^2\partial_{ss}v(t,s) - r\,v(t,s) = 0,$$
$$v(T,s) = \Phi(s),$$

   $w(t,x) := v(t,e^x)$ satisfies

$$\partial_t w(t,x) + (r - \frac{\sigma^2}{2})\,\partial_x w(t,x) + \frac{\sigma^2}{2}\,\partial_{xx}w(t,x) - r\,w(t,x) = 0,$$
$$w(T,x) = \Phi(e^x).$$

   $u(t,x) := e^{-rt}w(t,x) = e^{-rt}v(t,e^x)$ satisfies

$$\partial_t u(t,x) + (r - \frac{\sigma^2}{2})\,\partial_x u(t,x) + \frac{\sigma^2}{2}\,\partial_{xx}u(t,x) = 0,$$
$$u(T,x) = e^{-rT}\Phi(e^x).$$

3. *Explicit Method Solution*:

$$\begin{cases} u_n^M = e^{-t_M}\Phi(e^{x_n}), \\ u_n^{m-1} = p_u u_{n+1}^m + p_s u_n^m + p_d u_{n-1}^m, \quad m = M-1\dots 0, \end{cases} \tag{3}$$

   with $p_u = \frac{\sigma^2\delta t}{2(\delta x)^2} + \frac{\mu\delta t}{2(\delta x)}$, $p_d = \frac{\sigma^2\delta t}{2(\delta x)^2} - \frac{\mu\delta t}{2(\delta x)}$, and $p_s = 1 - p_d - p_u$.

Important Remarks.

1. Notice that the solution at each point (t_m, x_n) is a weighted average of the three closest points at t_{m+1}. Hence, we can think of this method as an *"Additive trinomial tree method"*.

2. Indeed, the piece-wise process $\widetilde{X}$ defined by

$$\widetilde{X}((m+1)\,\delta t) = \begin{cases} \widetilde{X}(m\,\delta t) + \delta x & \text{with prob. } p_u \\ \widetilde{X}(m\,\delta t) & \text{with prob. } p_s \\ \widetilde{X}(m\,\delta t) - \delta x & \text{with prob. } p_d \end{cases}$$

converges to $\log S_t := \sigma W_t + (r - \sigma^2/2)t$ as $\delta t \to 0$ and $\delta x \to 0$.

3. The explicit method is essentially Backward induction on a discrete approximation of the Black-Scholes model given by $S_0 e^{\widetilde{X}(t)}$.

4. For positive weights p_u, p_d, p_s, it suffices that

$$\frac{\sigma^2(\delta t)}{(\delta x)^2} < 1,$$

and that δt and δx are small enough.

1.2.4 Stability

Stability

1. *Question*: Under what conditions the explicit method is stable?

2. To analyze the problem, we first notice that $v(t,s)$ is the solution of

$$\partial_t v(t,s) + rs\partial_s v(t,s) + \frac{\sigma^2}{2} s^2 \partial_{ss} v(t,s) - r\, v(t,s) = 0, \quad v(T,s) = \Phi(s)$$

if and only if $v(t,s) = s^{1/2(1-\kappa)} e^{\frac{1}{8}(\kappa+1)^2\sigma^2(T-t)} u(\frac{1}{2}\sigma^2(T-t), \log s)$, with $\kappa := 2r/\sigma^2$ and $u(\tau, x)$ being the solution of the diffusion equation:

$$\partial_\tau u(\tau,x) - \partial_{xx} u(\tau,x) = 0, \quad u(0,x) = e^{1/2(1-\kappa)x}\Phi(e^x). \tag{4}$$

3. Next, apply the explicit method with the initial cond. $u(0,x) = e^{ikx}$. *Does the numerical solution u_n^m remain bounded when $\delta\tau \to 0$ and $\delta x \to 0$?*

4. It turn out that the solution u_n^m takes the general form:

$$u_n^m = \left\{1 + \frac{\delta t}{(\delta x)^2}\left[e^{ik\delta x} + e^{-ik\delta x} - 2\right]\right\}^m e^{ikx_n}$$

5. Thus, the condition $\alpha := \frac{\delta t}{(\delta x)^2} \leq \frac{1}{2}$ is necessary for stability. Since almost any function $g(x)$ can be expressed as series expansions $\sum_{k\in Z} e^{ikx} a_k$, the conditions is both *necessary and sufficient*.

Stability: General case.

1. Consider the method (2). It follows that if $p_u, p_d, p_s \geq 0$, and $a(t,x)$, $\mu(t,x)$, and $g(x)$ are bounded, then

$$\sup_{n,m} |a_n^m| \leq G < \infty.$$

2. Inspection of the coefficients of (2) makes evident that if $|a(t,x)| < A$ and

$$A \cdot \frac{\delta t}{(\delta x)^2} < 1,$$

then $p_u, p_d, p_s \geq 0$ for small enough δt and δx.

3. *How bad can the instability be?* W, H, & D apply the explicit method to (4) and then approximate the value of a put option (Strike=10, Mat=$1/2$ year) in the BS model ($\sigma = 20\%$, $r = 5\%$):

S_0	$\alpha = .25$	$\alpha = .5$	$\alpha = .52$	Exact
7	2.75	2.75	-17.41	2.75
10	.44	.44	625	.44
14	.0028	.0027	-15.21	.0028

1.2.5 Discretization error and convergence.

What is the discretization error?

1. *Will the solution of the finite-difference system converge to the solution of the PDE as the mesh parameters shrink?*

2. *At what rate does the discretization error* $\max_{n,m} |u(t_m, x_n) - u_n^m|$ *converge to* 0*?*

3. *Example:* Consider the explicit method (2) applied to the heat equation (4). It turns out that

$$\max_{n,m} |u(t_m, x_n) - u_n^m| = O(\delta t) + O((\delta x)^2).$$

This follows from plugging $u(t_m, x_n)$ into the finite-difference operator and use Taylor's expansions of u.

4. *The above rate of convergence holds for the general system (1)* under boundedness conditions on the coefficients a and μ.

1.3 Implicit methods.

1.3.1 Description

General description.

1. *Motivation:*
   - The stability condition $\delta t/(\delta x)^2 < 1$ of the explicit methods implies that if we one wishes to improve accuracy by doubling the number of x-points, one must quarter the time mesh, taking about 8 times longer.
   - "Implicit methods" will allow us to increase the number of x-points without having to take very small time mesh.

2. *Basic idea:* Approximate $\frac{\partial u}{\partial t}$ by a linear combination of forward and backward difference approximations:

$$\frac{\partial u(t,x)}{\partial t} \approx \theta \cdot \frac{u(t+\delta t,x) - u(t,x)}{\delta t} + (1-\theta) \cdot \frac{u(t,x) - u(t-\delta t,x)}{\delta t}, \quad (5)$$

3. *Known methods*:
   - $\theta = 0 \Longrightarrow$ Explicit method
   - $\theta = 1 \Longrightarrow$ Fully implicit method
   - $\theta = 1/2 \Longrightarrow$ Crank-Nicolson method

Fully Implicit Method.

- Consider the PDE:

$$\partial_t u(t,x) + \mu(t,x)\, \partial_x u(t,x) + a(t,x)\, \partial_{xx} u(t,x) - r\, u(t,x) = 0. \quad (6)$$

- *Discretization:* Using Forward diff. in time and centered diff. in space:

$$\frac{u_n^{m+1} - u_n^m}{\delta t} + \mu_n^m \frac{u_{n+1}^m - u_{n-1}^m}{\delta x} + a_n^m \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\delta x)^2} - r\, u_n^m = 0, \quad (7)$$

  where $\mu_n^m := \mu(t_m, x_n)$, $a_n^m := a(t_m, x_n)$, and $u_n^m := \widetilde{u}(t_m, x_n) \approx u(t_m, x_n)$.

- *Solving the system:* Notice that $u_\cdot^m$ can not be solved explicitly, and rather, $\mathbf{u}^m := (u_1^m, \ldots, u_{N-1}^m)$ is "implicitly" defined in terms of $\mathbf{u}^{m+1}$ via a linear system of equations:

$$(\mathrm{Id} - \delta t \cdot A_{\delta x}^m)\, \mathbf{u}^m = \mathbf{u}^{m+1}, \quad m < M, \quad (8)$$

  with final cond. $u_n^M = g(x_n)$ and Dirichlet cond. $u_0^m = u_N^m = 0$, m<M.

Fully Implicit Method. Cont.

- The matrix $A^m_{\delta x}$ is given by

$$A^m_{\delta x} = \begin{bmatrix} -\beta^m_1 & \gamma^m_1 & 0 & 0 & 0 & \dots & \dots & 0 \\ \alpha^m_2 & -\beta^m_2 & \gamma^m_2 & 0 & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \alpha^m_{N-1} & -\beta^m_{N-1} \end{bmatrix},$$

  where

$$\alpha^m_n = \frac{a^m_n}{(\delta x)^2} + \frac{\mu^m_n}{(\delta x)}, \quad \gamma^m_n = \frac{a^m_n}{(\delta x)^2} - \frac{\mu^m_n}{(\delta x)}, \quad \beta^m_n = 2\frac{a^m_n}{(\delta x)^2} + r. \tag{9}$$

- The implicit method resulting from the approximation (5) (with general θ) will take a form:

$$(\mathrm{Id} - \theta\delta t \cdot A^m_{\delta x})\,\mathbf{u}^m = (\mathrm{Id} + (1-\theta)\delta t \cdot A^m_{\delta x})\mathbf{u}^{m+1}, \quad m < M, \tag{10}$$

  with final cond. $u^M_n = g(x_n)$ and Dirichlet cond. $u^m_0 = u^m_N = 0$, m<M.

1.3.2 Solving the system

Overview

- The tridiagonal matrix $(\mathrm{Id} - \delta t \cdot A^m_{\delta x})$ is not necessarily invertible. However, it is known that a “diagonally dominant" matrix is invertible. *Under what conditions the matrix is diagonally dominant?*

- There are two popular methods to find $\mathbf{u}^m$ in (8)-(10): LU decomposition and iterative methods.

- The LU method is an “exact" method. It consists of bringing the matrix $(\mathrm{Id} - \delta t \cdot A^m_{\delta x})$ into a lower triangular matrix (using elementary operations) and then solve using substitution.

- Iterative methods construct a sequence $\mathbf{u}^{m,k}$, $k = 0, \dots,$ such that

$$\lim_{k\to\infty} \mathbf{u}^{m,k}_n = \mathbf{u}^m_n, \quad m < M, \quad n = 1, \dots, N-1.$$

- All these method exploits heavily the tridiagonal structure of the matrix $(\mathrm{Id} - \delta t \cdot A^m_{\delta x})$.

Iterative solution methods

- The key idea comes from the following manipulation of (7), where α_n^m, β_n^m, and γ_n^m are as in (9):

  $$u_n^m = \frac{1}{1+(\delta t)\beta_n^m}\left\{u_n^{m+1} + \alpha_n^m(\delta t)u_{n-1}^m + \gamma_n^m(\delta t)u_{n+1}^m\right\}, \quad n = 1, \dots, N-1.$$

  The above can be seen as an equation of the form $\mathbf{u}^m = H(\mathbf{u}^m)$, motivating the use of a fixed point type method: $\mathbf{u}^{m,k+1} = H(\mathbf{u}^{m,k})$.

- *Jacobi*: Starting with $u_n^{m,0} = u_n^{m+1}$, compute iteratively

  $$u_n^{m,k+1} = \frac{1}{1+(\delta t)\beta_n^m}\left\{u_n^{m+1} + \alpha_n^m(\delta t)u_{n-1}^{m,k} + \gamma_n^m(\delta t)u_{n+1}^{m,k}\right\}.$$

- *Gauss-Seidel*: Starting with $u_n^{m,0} = u_n^{m+1}$, compute iteratively

  $$u_n^{m,k+1} = \frac{1}{1+(\delta t)\beta_n^m}\left\{u_n^{m+1} + \alpha_n^m(\delta t)u_{n-1}^{m,k+1} + \gamma_n^m(\delta t)u_{n+1}^{m,k}\right\},$$

  for $n = 1, \dots, N-1$, in that order.

- *SOR (Successive Over-Relaxation)*: Starting with $u_n^{m,0} = u_n^{m+1}$, compute iteratively for $n = 1, \dots, N-1$, in that order,

  $$y_n^{m,k+1} = \frac{1}{1+(\delta t)\beta_n^m}\left\{u_n^{m+1} + \alpha_n^m(\delta t)u_{n-1}^{m,k+1} + \gamma_n^m(\delta t)u_{n+1}^{m,k}\right\},$$
  $$u_n^{m,k+1} = u_n^{m,k} + \omega\left(y_n^{m,k+1} - u_n^{m,k}\right),$$

  where $1 < \omega < 2$ (the over-relaxation parameter). There is an optimal $\omega^* \in (1,2)$, that make the method to converge the fastest.

1.3.3 Convergence

Convergence of implicit methods.

Theorem 1 (See Lamberton and Lapeyre for references and details)**.** *Let* $u(t,x)$ *be the solution of (6) and let* $\mathbf{u}^m = [u_1^m, \dots, u_{N-1}^m]'$ *be the solution of the system (10). Define* $\widetilde{u}(t,x) := \widetilde{u}^{\delta t, \delta x}(t,x)$ *as follows:*

$$\widetilde{u}(t,x) := \begin{cases} u_n^m & \text{if } x_n - \frac{\delta x}{2} < x \le x_n + \frac{\delta x}{2}, \quad (m-1)\delta t < t \le m\delta t \\ 0 & o.w. \end{cases}$$
$$\partial_x \widetilde{u}(t,x) = \frac{1}{\delta x}\left\{\widetilde{u}(t, x+\delta x/2) - \widetilde{u}(t, x-\delta x/2\right\}$$

Then,

- *When* $1/2 \le \theta \le 1$*,*

  $$\lim_{\delta t, \delta x \to 0} \widetilde{u}(t,x) = u(t,x), \qquad \lim_{\delta t, \delta x \to 0} \partial_x \widetilde{u}(t,x) = \partial_x u(t,x),$$

- *When* $0 \le \theta < 1/2$*, the limit above holds provided that* $\alpha := \delta t/(\delta x)^2 \to 0$ *as* $\delta t, \delta x \to 0$.