MTH 9821 Numerical Methods for Finance

Fall 2016

Homework 8

Assigned: November 10; Due: November 17

This homework is to be done as a group. Each team will hand in one homework solution, and each member of the team should write at least one problem. On the cover page of the homework, please indicate the members of the team and who wrote each problem.

Finite Differences Solution of the Heat PDE

The goal of this homework is to write a portable finite difference code for solving the heat equation on a bounded domain as follows:

$$u_\tau \;=\; u_{xx}, \quad \forall\, x_{left} \;<\; x \;<\; x_{right}, \quad \forall\, 0 \;<\; \tau \;<\; \tau_{final},$$

with boundary conditions

$$\begin{aligned} u(x,0) &= f(x), \quad \forall\, x_{left} \;\le\; x \;\le\; x_{right}; \\ u(x_{left},\tau) &= g_{left}(\tau), \quad \forall\, 0 \;\le\; \tau \;\le\; \tau_{final}; \\ u(x_{right},\tau) &= g_{right}(\tau), \quad \forall\, 0 \;\le\; \tau \;\le\; \tau_{final}. \end{aligned}$$

Input for the code:

• x_{left}, x_{right}, and τ_{final}, which define the computational domain;

• the boundary conditions functions $f(x)$, $g_{left}(\tau)$, and $g_{right}(\tau)$;

• the initial number of intervals on the τ–axis, denoted by M, and the initial number of intervals on the x–axis, denoted by N;

Specific data for this homework:

$$x_{left} = -2; \quad x_{right} = 2; \quad \tau_{final} = 1.$$

$$\begin{aligned} f(x) &= e^x, \ \forall\, x \in [-2,2]; \\ g_{left}(\tau) &= e^{\tau-2}, \ \forall\, \tau \in [0,1]. \\ g_{right}(2,\tau) &= e^{\tau+2}, \ \forall\, \tau \in [0,1]. \end{aligned}$$

Note: the exact solution is $u_{exact}(x,\tau) = e^{x+\tau}$.

Domain discretization: Given M and N, the initial number of intervals on the τ–axis and on the x–axis, respectively, the fixed Courant constant α is

$$\alpha \;=\; \frac{\delta\tau}{(\delta x)^2} \;=\; \frac{\tau_{final}}{(x_{right}-x_{left})^2}\,\frac{N^2}{M}.$$

The number of intervals will double on the x–axis and quadruple on the τ–axis.

For this homework, use $M = 8$ and $N = 4$, corresponding to $\alpha = 0.125$, $M = 8$ and $N = 8$, corresponding to $\alpha = 0.5$, $M = 8$ and $N = 16$, corresponding to $\alpha = 2$.

Finite difference solvers:

Use Forward Euler, Backward Euler, and Crank-Nicolson with $\alpha \in \{0.125, 0.5, 2\}$ to solve the diffusion equation for $u(x, \tau)$. For solving the implicit methods, use both tridiagonal LU without pivoting and SOR with relaxation parameter $\omega = 1.2$. The stopping criterion for SOR is that the norm of the difference between two consecutive approximations is less than $tol = 10^{-6}$.

Note: there are 15 different approximations, corresponding to:

(1) Forward Euler, $\alpha \in \{0.125, 0.5, 2\}$ (three);
(2) Backward Euler, tridiagonal LU without pivoting, $\alpha \in \{0.125, 0.5, 2\}$ (three);
(3) Backward Euler, SOR, $\omega = 1.2$, $\alpha \in \{0.125, 0.5, 2\}$ (three);
(4) Crank-Nicolson, tridiagonal LU without pivoting, $\alpha \in \{0.125, 0.5, 2\}$ (three);
(5) Crank-Nicolson, SOR, $\omega = 1.2$, $\alpha \in \{0.125, 0.5, 2\}$ (three);

Run each finite difference method for the initial value of M chosen above, and then quadruple the number of points on the τ-axis, i.e., choose $M \in \{4M, 16M, 64M\}$.

To make sure your codes run properly, please report the following: for $\alpha = 0.5$, $M = 8$ and $N = 8$, run Forward Euler, and Backward Euler Crank-Nicolson (with tridiagonal LU without pivoting and with SOR) and record the values of the finite difference approximations at each nodes, including at the boundary nodes. Report 9 decimal digits. Thus, for each of the three methods above, you will have to fill out a table with 9 rows (corresponding to time steps from 0 – boundary conditions, to 8) and 9 columns (including the boundary conditions at x_{left} and x_{right}).

Convergence Analysis of the Finite Difference Solvers

Let $x_k = x_{left} + k\delta x$, $k = 0 : N$, be the nodal points on the x-axis. Denote by $u_{approx}(x_k, \tau_{final})$ the value obtained using a finite difference solver corresponding to the node x_k at time τ_{final}.

Maximum pointwise approximation error and Root-Mean-Squared (RMS) Error:

The pointwise approximation error of the finite difference solver at the node x_k is defined as

$$|u_{approx}(x_k, \tau_{final}) - u_{exact}(x_k, \tau_{final})|,$$

where $u_{exact}(x_k, \tau_{final})$ is the exact value of the solution of the heat equation. (Recall that $u_{exact}(x, \tau) = e^{x+\tau}$.) The maximum pointwise approximation error is defined as follows:

$$max\_pointwise\_error \;=\; \max_{k=1:(N-1)} |u_{approx}(x_k, \tau_{final}) - u_{exact}(x_k, \tau_{final})|.$$

The RMS error $error\_RMS$ is defined as

$$error\_RMS \;=\; \sqrt{\frac{1}{N+1} \sum_{k=0:N} \frac{|u_{approx}(x_k, \tau_{final}) - u_{exact}(x_k, \tau_{final})|^2}{|u_{exact}(x_k, \tau_{final})|^2}}.$$

For each finite difference method, compute and record $max\_pointwise\_error$ and $error\_RMS$, as well as the ratio of the approximation errors from one discretization level to the next.

Pricing European Put Options Using Finite Differences on a Fixed Computational Domain

An underlying asset has lognormal distribution with volatility $\sigma = 0.35$, spot price $S_0 = 41$, and pays dividends continuously at the rate $q = 0.02$. Consider a put option with strike $K = 40$ and maturity $T = 0.75$, i.e., 9 months, The interest rate is assumed to be constant and equal to $r = 0.04$.

We will solve the diffusion equation (the heat equation) on a bounded domain as follows:

$$u_\tau \;=\; u_{xx}, \;\forall\, x_{left} \;<\; x \;<\; x_{right}, \;\forall\, 0 \;<\; \tau \;<\; \tau_{final},$$

with boundary conditions

$$\begin{aligned} u(x,0) \;&=\; f(x), \;\forall\, x_{left} \;\leq\; x \;\leq\; x_{right}; \\ u(x_{left},\tau) \;&=\; g_{left}(\tau), \;\forall\, 0 \;\leq\; \tau \;\leq\; \tau_{final}; \\ u(x_{right},\tau) \;&=\; g_{right}(\tau), \;\forall\, 0 \;\leq\; \tau \;\leq\; \tau_{final}. \end{aligned}$$

1. **Computational domain:**

We will use a fixed computational domain, where the node $x_{compute} = \ln\left(\frac{S_0}{K}\right)$ will not be on the finite difference mesh.

The upper bound τ_{final} for τ is

$$\tau_{final} \;=\; \frac{T\sigma^2}{2}.$$

The computational domain on the x–axis is

$$[x_{left},\; x_{right}] \;=\; \left[\ln\left(\frac{S_0}{K}\right) + \left(r - q - \frac{\sigma^2}{2}\right)T - 3\sigma\sqrt{T},\; \ln\left(\frac{S_0}{K}\right) + \left(r - q - \frac{\sigma^2}{2}\right)T + 3\sigma\sqrt{T}\right].$$

The finite difference discretization is built as follows:

Start with M and α_{temp} given (α will be slightly smaller than α_{temp} in the end). Then,

$$\delta\tau \;=\; \frac{\tau_{final}}{M},$$

and δx will be approximately $\sqrt{\delta\tau/\alpha_{temp}}$. Let

$$N \;=\; \text{floor}\left(\frac{x_{right} - x_{left}}{\sqrt{\delta\tau/\alpha_{temp}}}\right),$$

where floor(y) is the largest integer smaller than or equal to y.

Then,

$$\delta x \;=\; \frac{x_{right} - x_{left}}{N}$$

and α is defined as

$$\alpha \;=\; \frac{\delta\tau}{(\delta x)^2}.$$

Note that $\alpha < \alpha_{temp}$.

2. **Boundary conditions**

Recall that

$$V(S,t) \;=\; \exp(-ax - b\tau)u(x,\tau),$$

with $x = \ln\left(\frac{S}{K}\right)$ and $\tau = \frac{(T-t)\sigma^2}{2}$.

For a put option, the boundary conditions for $u(x,\tau)$ are:

$$\begin{aligned} f(x) &= K\,\exp(ax)\ \max(1-\exp(x),0),\ \forall\, x_{left} \le x \le x_{right}; \\ g_{left}(\tau) &= K\,\exp(ax_{left}+b\tau)\ \left(\exp\left(-\frac{2r\tau}{\sigma^2}\right)-\exp\left(x_{left}-\frac{2q\tau}{\sigma^2}\right)\right),\ \forall\, 0 \le \tau \le \tau_{final}; \\ g_{right}(\tau) &= 0,\ \forall\, 0 \le \tau \le \tau_{final}. \end{aligned}$$

3. **Finite difference schemes**

Use Forward Euler with $\alpha = 0.45$, and Backward Euler and Crank-Nicolson with $\alpha \in \{0.45, 5\}$, to solve the diffusion equation for $u(x,\tau)$. For the implicit methods, use both tridiagonal LU and SOR with relaxation parameter $\omega = 1.2$. The stopping criterion for SOR is that the norm of the difference between two consecutive approximations is less than $tol = 10^{-6}$.

Run each finite difference method for the initial value $M = 4$, and then quadruple the number of points on the τ-axis, i.e., choose $M \in \{4M, 16M, 64M\}$.

To check the numbers, include the following: for Forward Euler with $\alpha = 0.45$, for Backward Euler with $\alpha = 0.45$, and for Crank-Nicolson with $\alpha = 0.45$, let $M = 4$. Run your codes and record the values of the finite difference approximations at each nodes, including at the boundary nodes. For $M = 4$ and $\alpha = 0.45$ the corresponding value of N is $N = 12$. Thus, for each method above, you will have to fill out a table with five rows (corresponding to time steps from time step 0 – the boundary conditions, to time step 4 – corresponding to τ_{final}) and 13 columns (the first and the last column correspond to the boundary conditions at x_{left} and x_{right}, respectively).

4. **Pointwise Convergence**

Identify the interval containing $x_{compute} = \ln\left(\frac{S_0}{K}\right)$, i.e., find i such that

$$x_i \le x_{compute} < x_{i+1}.$$

Let

$$S_i = Ke^{x_i}; \tag{1}$$

$$S_{i+1} = Ke^{x_{i+1}} \tag{2}$$

be the values of S corresponding to the nodes x_i and x_{i+1}. Let

$$\begin{aligned} V_i &= exp(-ax_i - b\tau_{final})U^M(i); \\ V_{i+1} &= exp(-ax_{i+1} - b\tau_{final})U^M(i+1) \end{aligned}$$

be the approximate values of the option at the nodes S_i and S_{i+1}, respectively, where $U^M(i)$ and $U^M(i+1)$ are the finite difference approximations of $u(x_i,\tau_{final})$ and $u(x_{i+1},\tau_{final})$, respectively.

The approximate value of the option at S_0 is computed from V_i and V_{i+1} as follows:

$$V_{approx}(S_0,0) = \frac{(S_{i+1}-S_0)V_i \ + \ (S_0-S_i)V_{i+1}}{S_{i+1}-S_i}.$$

Let $V_{exact}(S_0,0)$ be the Black–Scholes value of the option. The pointwise relative error of the finite difference solution is

$$error\_pointwise = |V_{approx}(S_0,0) - V_{exact}(S_0,0)|.$$

Another way of computing an approximate value for the option would be to use linear interpolation to find the value of $u(x_{compute},\tau_{final})$, and then use the change of variables to obtain $V_{approx,2}(S_0, 0)$, i.e.,

$$u(x_{compute}, \tau_{final}) = \frac{(x_{i+1} - x_{compute})u(x_i, \tau_{final}) + (x_{compute} - x_i)u(x_{i+1}, \tau_{final})}{x_{i+1} - x_i};$$

$$V_{approx,2}(S_0, 0) = exp(-ax_{compute} - b\tau_{final})u(x_{compute}, \tau_{final}).$$

Let

$$error\_pointwise\_2 = |V_{approx}(S_0, 0) - V_{exact}(S_0, 0)|$$

be the pointwise relative error corresponding to this method.

For each finite difference method, compute and record *error_pointwise* and *error_pointwise_2*, as well as the ratio of the approximation errors from one discretization level to the next.

5. **Root-Mean-Squared (RMS) Error**

Let $x_k = x_{left} + k\delta x$, $k = 0 : N$, be a nodal point on the x-axis, and let $S_k = Ke^x_k$ be the corresponding asset value. The approximate value of a call option with spot price S_k obtained from the finite difference scheme is

$$V_{approx}(S_k, 0) = \exp(-ax_k - b\tau_{final})U^M(k),$$

where $U^M(k)$ is the finite difference approximation of $u(x_k, \tau_{final})$.

Let $V_{exact}(S_k, 0)$ be the Black-Scholes value of a call option with spot price S_k.

Denote by N_{RMS} the number of nodes k such that $V_{exact}(S_k, 0) > 0.00001 \cdot S_0$. The RMS error $error\_RMS$ is defined as

$$error\_RMS = \sqrt{\frac{1}{N_{RMS}} \sum_{0 \le k \le N \text{ with } \frac{V_{exact}(S_k,0)}{S_0} > 0.00001} \frac{|V_{approx}(S_k, 0) - V_{exact}(S_k, 0)|^2}{|V_{exact}(S_k, 0)|^2}}$$

For each finite difference method, compute and record $error\_RMS$, as well as the ratio of the RMS errors from one discretization level to the next.

6. **Finite Difference Approximation of Δ, Γ, and Θ:**

Recall that x_i and x_{i+1} are consecutive nodes such that $x_i \le x_{compute} < x_{i+1}$.

Let

$$S_{i-1} = Ke^{x_{i-1}};$$
$$S_{i+2} = Ke^{x_{i+2}}$$

be the values of S corresponding to the nodes x_{i-1} and x_{i+2}, respectively. Let V_{i-1} and V_{i+2} be the approximate values of the option at the nodes S_{i-1} and S_{i+2}, i.e.,

$$V_{i-1} = exp(-ax_{i-1} - b\tau_{final})U^M(i-1);$$
$$V_{i+2} = exp(-ax_{i+2} - b\tau_{final})U^M(i),$$

where $U^M(i-1)$ and $U^M(i)$ are the finite difference approximations of $u(x_{i-1}, \tau_{final})$ and $u(x_{i+2}, \tau_{final})$, respectively.

Compute the following finite different approximations of the Delta and of the Gamma of the option:

$$\Delta_{fd} = \frac{V_{i+1} - V_i}{S_{i+1} - S_i};$$

$$\Gamma_{fd} = \frac{\frac{V_{i+2}-V_{i+1}}{S_{i+2}-S_{i+1}} - \frac{V_i - V_{i-1}}{S_i - S_{i-1}}}{\frac{S_{i+2}+S_{i+1}}{2} - \frac{S_i + S_{i-1}}{2}}.$$

A finite difference approximation for the Theta of the option, i.e., for $\Theta = \frac{\partial V}{\partial t}$, can be obtained as follows:

From the change of variables $\tau = \frac{(T-t)\sigma^2}{2}$, it follows that

$$t \;=\; T - \frac{2\tau}{\sigma^2}. \tag{3}$$

Recall that $\tau_{final} = \frac{T\sigma^2}{2}$. Thus, $T = \frac{2\tau_{final}}{\sigma^2}$, and, from (3), we find that

$$t \;=\; \frac{2(\tau_{final} - \tau)}{\sigma^2}.$$

Then, the next to last time step on the τ–axis, i.e., $\tau_{final} - \delta\tau$, corresponds to time

$$\delta t \;=\; \frac{2(\tau_{final} - (\tau_{final} - \delta\tau))}{\sigma^2} \;=\; \frac{2\delta\tau}{\sigma^2}$$

on the t–axis.

Let

$$\begin{aligned} V_{i,\delta t} &= exp(-ax_i - b(\tau_{final} - \delta\tau))U^{M-1}(i); \\ V_{i+1,\delta t} &= exp(-ax_{i+1} - b(\tau_{final} - \delta\tau))U^{M-1}(i+1), \end{aligned}$$

where $U^{M-1}(i)$ and $U^{M-1}(i+1)$ are the finite difference approximations of $u(x_i, \tau_{final} - \delta\tau)$ and $u(x_{i+1}, \tau_{final} - \delta\tau)$, respectively. Let

$$V_{approx}(S_0, \delta t) \;=\; \frac{(S_{i+1} - S_0)V_{i,\delta t} \;+\; (S_0 - S_i)V_{i+1,\delta t}}{S_{i+1} - S_i}.$$

Compute the following finite difference approximation for the Theta of the option:

$$\Theta_{fd} \;=\; \frac{V_{approx}(S_0, 0) - V_{approx}(S_0, \delta t)}{\delta t}.$$