Exotic option pricing

Binary options

- These options function as bets on the level of the asset price. The payoff functions are of the form $\Lambda(S) = \chi_A(S)$ for some set A.
- For example, if $A = [E, \infty)$, then

$$\Lambda(S) = \begin{cases} 1 & \text{if} \quad S \geq E \\ 0 & \text{otherwise.} \end{cases}$$

- Clearly any binary option on a reasonable set A can be made up of a combination of options of the above form.

Black-Scholes for a binary option.

If $A = [E, \infty)$, and $C_A(S,t)$ denotes the value of a binary option on A with expiry T on a log-normal asset with volatility σ, then

$$\boxed{C_A(S,t) = \mathrm{e}^{-r(T-t)} N(d_-)},$$

where d_- has the usual definition.

Barrier options

- Barrier options are primed either to self-destruct (become worthless), or to spring into existence if the asset price crosses a barrier *at any point during the life of the contract.*
- The advantage of such a contract is that it offers much the same facilities as the corresponding vanilla option, but will be cheaper. Of course the holder runs the risk of the contract evaporating before his or her eyes, but if the asset is unlikely to pass the barrier, this is correspondingly unlikely to happen.
- Basic barrier options can be formulated as

$$\left\{ \begin{matrix} \text{up} \\ \text{down} \end{matrix} \right\} \text{ - and - } \left\{ \begin{matrix} \text{in} \\ \text{out} \end{matrix} \right\} \left\{ \begin{matrix} \text{put} \\ \text{call} \end{matrix} \right\} \text{ options.}$$

  Of course, many other flavours than these eight can be and have been devised.
- Sometimes *rebate payments* are made when the barrier is breached, to compensate for the loss in the value of the option. For example, an up-and-out barrier option could trigger a payment of $B - K$ if the asset reaches the barrier value B. This correspond to a policy of forced early exercise at this point.

Black-Scholes for barriers

For the basic types of barrier options Black-Scholes-type formulae can be derived. Two examples—for a down-and-out and up-and-out call barrier at $S = B$:

- down-and-out call:

$$C(S,t) - (S/B)^{1-2r/\sigma^2} C(B^2/S,t), \qquad S \geq B.$$

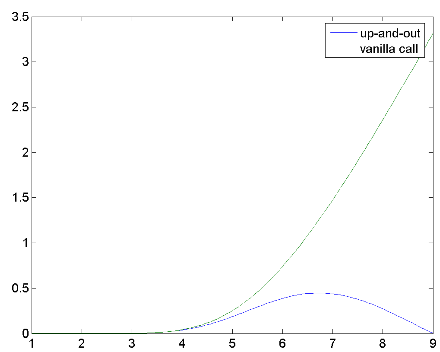
- up-and-out call:

$$S\left[N\left(d_+\left(\tfrac{S}{K}\right)\right) - N\left(d_+\left(\tfrac{S}{B}\right)\right) - \left(\tfrac{B}{S}\right)^{1+\frac{2r}{\sigma^2}}\left(N\left(-d_+\left(\tfrac{B}{S}\right)\right) - N\left(-d_+\left(\tfrac{B^2}{SK}\right)\right)\right)\right]$$
$$- K\mathrm{e}^{-r(T-t)}\left[N\left(d_-\left(\tfrac{S}{K}\right)\right) - N\left(d_-\left(\tfrac{S}{B}\right)\right) - \left(\tfrac{B}{S}\right)^{-1+\frac{2r}{\sigma^2}}\left(N\left(d_-\left(\tfrac{B}{S}\right)\right) - N\left(-d_-\left(\tfrac{B^2}{SK}\right)\right)\right)\right],$$

  where

$$d_\pm(u) = \frac{\log u + (r \pm \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}.$$

Up-and-out call option values



Binomial for barriers

- We may price a barrier option using a binomial tree if we wish. All we need to do is to assign a zero option value at any of the nodes of the tree that lie beyond the barrier(s).
- In this way we can price variations on the barrier theme:
  - double barriers.
  - partial barriers.
- But we need to be careful: the answers are not great, unless we can locate the barrier accurately on our grid.

Bermudan options

- Bermudan options are somewhere between American and European options. The holder can exercise only at a discrete set of times. (In the european case, this is also true - the set of times includes only the expiry time.)
- No explicit option formula exists except in the most trival cases.
- If we are able to have nodes of our tree at the exercise times, then we can approximate the Bermudan option value using the binomial method.

Shout options

- The holder of a shout option has the chance to lock in the payoff to be at least the payoff at the current asset price at a single time of his/her choosing during the life of the contract.
- It will only be rational to do this at time τ if $S(\tau) > E$, where E is the default strike price specified in the contract.
- The value of the payoff at the final time if the shout occurs at time τ is

$$\max(S(T) - E, S(\tau) - E) = \max(S(T) - S(\tau), 0) + S(\tau) - E.$$

  The value of this payoff at time τ is $S(\tau) - E$ plus the value of a european option with expiry T, strike $S(\tau)$ and the initial asset price of $S(\tau)$.
- We can use the binomial approach to compute the value of such an option.

Lookback options

- Lookback options are path-dependent options in that the payoff depends on some property of the asset path over the life of the option. Specifically, the term 'lookback' is used to describe options that depend on the maximum or minimum value attained by the asset. Define

$$S^{\max} := \max_{[0,T]} S(t) \quad \text{and} \quad S^{\min} := \min_{[0,T]} S(t).$$

- There are two main varieties, distinguished by the way in which the payoff depends on $S^{\max / \min}$.
  - Fixed-strike lookback options have payoffs of the form

$$\Lambda(S([0,T])) = (S^{\max} - E)_+ \quad \text{(for example).} \quad .$$

  - An example of a floating-strike lookback option payoff might be

$$\Lambda(S([0,T])) = (S - S^{\min})_+ = S - S^{\min}.$$

    Floating–strike options will either always be exercised or never be exercised.

Binomial for lookbacks

- It is possible to define binomial schemes for lookback options. One defines auxiliary variables that monitor the maximum or mimimum value—there will be several of these at most nodes—and defines the option payoff in terms of the values of these variables.
- The disadvantage is the increase in the amount of storage and work. At the final time step, all possible values of $S^{\max}$ at each node will need to be computed (and stored), each having its own associated option value. If the tree is defined so that $ud = 1$ this may be a manageable number. Otherwise, the cost grows exponentially with the number of timesteps.

Asian options

- Asian options are another type of path-dependent option. In this case the option depends on an average value of the asset, discretely or continuously sampled.
- With continuous sampling, we can define

$$A(t_1,t_2) = \begin{cases} \frac{1}{t_2-t_1}\int_{t_1}^{t_2} S(t)\mathrm{d}t & \text{(arithmetic)} \\ \exp\left(\frac{1}{t_2-t_1}\int_{t_1}^{t_2} \ln S(t)\mathrm{d}t\right) & \text{(geometric).} \end{cases}$$

- With discrete sampling, the integrals are replaced by averages over samples at times $\{\tau_i\}$ falling in the interval (t_1, t_2):

$$A(t_1,t_2) = \begin{cases} \frac{1}{\#\{\tau_i\in[t_1,t_2]\}}\sum_{\tau_i\in[t_1,t_2]} S(\tau_i)) & \text{(arithmetic)} \\ \exp\left(\frac{1}{\#\{\tau_i\in[t_1,t_2]\}}\sum_{\tau_i\in[t_1,t_2]} \ln S(\tau_i)\right) & \text{(geometric).} \end{cases}$$

- An average-strike call option has payoff $(S - A)_+$, whereas an average-price version has payoff $(A - E)_+$.

Asian options—geometric averaging

- For the *arithmetic average* options, no explicit formula exists. However, for the *geometric average*, it turns out that we can derive formulae for European options.
- The payoff of a geometric average Asian option might be

$$\max\left(\exp\left(\frac{1}{N}\sum_{n=1}^{N}\log S_{T_n}\right) - K, 0\right).$$

- If the log-returns of the asset are independent, and normally distributed, and if the sample times are $t_j = jh$, with $h = T/N$, then we can price these options using the Black-Scholes formula, with modified volatility and dividend yield terms:

$$\sigma^* = \sigma\sqrt{\frac{(N+1)(2N+1)}{6N^2}}$$
$$\delta^* = r\left(\frac{N-1}{2N}\right) + \delta\left(\frac{N+1}{2N}\right) + \frac{\sigma^2}{2}\left(\frac{N^2-1}{6N^2}\right).$$

- For average strike versions, we need to use formulae for exchange options, covered below.

Exchange options

- These give the holder the right to exchange one asset for another. (Ordinary puts and calls do the same thing with cash and stock.)
- If S_t^1 and S_t^2 are the prices of the two assets, we write their dividend yields as δ_i and their volatilities as σ_i $(i = 1, 2)$. If the instantaneous correlation between the log-returns of the two assets is ρ, then the formula for the price of an option to receive the first asset in exchange for the second at time T is

$$S_1 \mathrm{e}^{-\delta_1 T} N(d_+) - S_2 \mathrm{e}^{-\delta_2 T} N(d_-),$$

where

$$d_\pm = \frac{\log \frac{S_1 \mathrm{e}^{-\delta_1 T}}{S_2 \mathrm{e}^{-\delta_2 T}} \pm \frac{\sigma^2}{2} T}{\sigma \sqrt{T}},$$

and

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2.$$

This formula was first published by Margrabe in 1978.
- Notice that we recover the Black-Scholes formula for a call if $S_2 \equiv K$ and $\delta_2 = r$, and for a put if $S_1 \equiv K$ and $\delta_1 = r$.

Monte-Carlo for path-dependence

- The Monte-Carlo method can be used for path-dependent options. Instead of directly sampling the asset prices at expiry, it is necessary to sample complete paths (although we cannot do this).
- But what we can do is simulate a discrete path. We set down sample times $0 = t_0 < t_1 < \cdots < t_M = T$, define $\Delta t_m = t_{m+1} - t_m$, and we generate a sample Brownian motion via

$$W_0 = 0; \qquad W_{m+1} = W_m + \Delta W_m,$$

where $\Delta W_m \sim N(0, \Delta t_m)$. This is an example of an Euler-Maruyama discretisation.
- If we want to simulate a solution to the SDE

$$\mathrm{d}X(t) = a(X(t), t)\mathrm{d}t + b(X(t), t)\mathrm{d}W(t)$$

then the Euler-Maruyama discretisation takes the form

$$\Delta X_m = a_m(X_m)\Delta t_m + b_m(X_m)\Delta W_m,$$

where $a_m(x) = a(x, t_m)$, etc., and $\Delta X_m = X_{m+1} - X_m$.

Monte-Carlo for path-dependence
Strong and weak convergence

- The discrete paths produced by the approximation will differ from (samples of) the paths produced by the original SDE. This source of error is in addition to the sampling error that comes as standard with Monte-Carlo methods.
- *Strong convergence* relates to this discrepancy. But we are more interested in computing expectations of payoff functions evaluated using the approximate paths. The accuracy of these computations will be related to the *weak convergence* properties of the discretisation.
- The Milstein discretisation gives an improved rate of strong convergence:

$$\Delta X_m = a_m(X_m)\Delta t_m + b_m(X_m)\Delta W_m + \frac{1}{2} b_m(X_m) b_m'(X_m) \left((\Delta W_m)^2 - \Delta t_m\right).$$

Finite difference methods

Objectives

- See how the Black-Scholes equation can be transformed into the heat equation on a line.
- Learn about explicit finite difference methods for the heat equation and understand the concepts of stability and convergence.

The Black-Scholes PDE

The value $V(S,t)$ at time t of a european-style derivative contract written on an underlying asset S, with payoff $\Lambda(S)$ at expiry T, and with a risk-free interest rate of r, satisfies, in $(0,\infty) \times [0,T)$,

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} = rV$$
$$V(S,T) = \Lambda(S).$$

In order to complete the specification we need to supply boundary conditions. For example, for a call option with strike price K:

$$\lim_{S\to 0} V(S,t) = 0, \qquad V(S,t) \sim S - K\mathrm{e}^{-r(T-t)} \quad \text{as} \quad S \to \infty.$$

For a put option with the same strike:

$$\lim_{S\to 0} V(S,t) = K\mathrm{e}^{-r(T-t)}, \qquad \lim_{S\to\infty} V(S,t) = 0.$$

Transformation

We have already seen one change of variable that reduces the Black-Scholes PDE to the heat equation. Here is another. Set

$$u(x,t) = e^{\hat{r}_- x + \hat{r}_+^2 t} V\left(Ke^x, T - \frac{2t}{\sigma^2}\right),$$

where

$$\hat{r}_\pm = \frac{r}{\sigma^2} \pm \frac{1}{2},$$

and where K is the strike price (or some other parameter associated with the payoff).
We write $\lambda(x) = e^{\hat{r}_- x}\Lambda(S)$. Then $u(x,0) = \lambda(x)$. For the boundary conditions we have, for a call option,

$$\lim_{x\to-\infty} u(x,t) = 0 \quad \text{and} \quad u(x,t) \sim Ke^{\hat{r}_- x + \hat{r}_+^2 t}(e^x - e^{-\frac{2rt}{\sigma^2}})$$

as $x \to \infty$.

The heat equation

We want to solve the equation

$$\begin{aligned} u_t &= u_{xx}, \quad x \in \mathbb{R}, \quad t > 0,\\ u(x,0) &= \lambda(x), \quad x \in \mathbb{R},\\ u(x,t) &\sim \alpha(x,t) \quad \text{as} \quad x \to -\infty, \quad 0 < t,\\ u(x,t) &\sim \beta(x,t) \quad \text{as} \quad x \to \infty, \quad 0 < t \end{aligned}$$

for some functions λ, α and β.
In order to approximate the solution to this numerically, we will attack the truncated problem:

$$\begin{aligned} u_t &= u_{xx}, \quad x \in (x_L, x_R), \quad 0 < t \le t_{\max},\\ u(x,0) &= \lambda(x), \quad x \in [x_L, x_R],\\ u(x_L,t) &= \alpha(x_L,t), \quad 0 < t \le t_{\max},\\ u(x_R,t) &= \beta(x_R,t)), \quad 0 < t \le t_{\max}. \end{aligned}$$

The grid

- The first step in the finite-difference method is to lay down a *grid* of points in the (x,t) plane, covering the region $[x_L, x_R] \times [0, t_{\max}]$. It is convenient at this stage to assume that the grid is equally-spaced and cartesian, so we get points (x_j, t_n), $j = 0, \ldots, J$, $n = 0, \ldots, N$, with $x_j = x_L + j\Delta x$, $j = 0, \ldots, J$, $\Delta x = (x_R - x_L)/J$ and with $t_n = n\Delta t$, $\Delta t = t_{\max}/N$.
- We are going to compute numbers U_j^n associated with these nodes that are going to be our approximations to the true values $u_j^n = u(x_j, t_n)$. We are going to do this by establishing *relationships* between the values at neighbouring nodes that approximately mirror the balance satisfied by the true solution. We do this be replacing *derivatives* in the heat equation by *finite differences*.

Finite differences

- For example, we can approximate the value $\frac{\partial u}{\partial t}$ at the node (x_j, t_n) (denoted $\partial_t u_j^n$) by

$$\partial_t u_j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t} =: \delta_t^+ u_j^n.$$

  A Taylor expansion will show that the error in this approximation is roughly $\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}(x_j, t_n)$ (any extra terms involve higher powers of Δt, and higher derivatives of u). Thus the error is small when Δt is small.
- The second derivative $\frac{\partial^2 u}{\partial x^2}$, denoted $\partial_x^2 u_j^n$, can be approximated similarly:

$$\partial_x^2 u_j^n \approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} =: \delta_x^+ \delta_x^- u_j^n.$$

  This time the difference between these two expressions is roughly $\frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(x_j, t_n)$.

Discrete balance

- If we replace the balance represented by the heat equation by a discrete balance between our values U_j^n at the nodes of our grid, we obtain, for $0 < j < J$ and $0 \le n < N$,

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}.$$

  Rearranging this gives

$$U_j^{n+1} = \nu U_{j+1}^n + (1 - 2\nu)U_j^n + \nu U_{j-1}^n,$$

  where $\nu = \frac{\Delta t}{\Delta x^2}$.
- At $n = 0$, $j = 0$ and $j = J$ we impose boundary conditions:

$$U_j^0 = \lambda(x_j), \quad 0 \le j \le J,$$
$$U_0^n = \alpha(x_L, t_n), \quad 0 < n \le N,$$
$$U_J^n = \beta(x_R, t_n), \quad 0 < n \le N.$$

More about boundary conditions

- We have just given Dirichlet boundary conditions for u: the value of U at the boundary is specified. There are alternatives. For example, Neumann boundary conditions specify a value for the derivative of u (i.e. u_x)at the boundary.
- A discrete implementation of the Neumann condition $u_x(x_R, t) = 0$ might look like:

$$U_J^n = U_{J-1}^n.$$

- Another alternative might be to specify that $u_{xx} = 0$ at the boundary, i.e. that the solution is linear. Discretely, this might look like

$$U_{J-1}^n = 2U_J^n - U_{J+1}^n.$$

  Here we have introduced a fictitious point beyond the boundary (whose value is now given in terms of U_J^n and U_{J-1}^n). Now the discrete equation is applied for $j = J$ too.

Experiments

The file `expt1.m` will run `fdexp.m` and `showsol.m`. These programs implement an explicit finite difference method for solving the constant-coefficient diffusion equation.

- Try changing the value of J to see what the effect is. I suggest the following values: 70, 50, 30, 10, 110,...
- Go back to J=90 and try varying N - increasing it and decreasing it.

When the program worked you were watching the process of *diffusion* taking place. When things went wrong, the problems you observed were those of *instability*.

Stability

If we take a closer look at the discretised heat equation we can try to figure out when things are going to go wrong. We can say that

$$\begin{aligned}\left|U_j^{n+1}\right| &\leq \nu\left|U_{j-1}^n\right| + |1-2\nu|\left|U_j^n\right| + \nu\left|U_{j+1}^n\right| \\ &\leq (2\nu + |1-2\nu|)\,\|U^n\|_\infty\,,\end{aligned}$$

where for a vector v, $\|v\|_\infty = \max_{0\leq j\leq J}\left|v_j\right|$. Then we can say that

$$\max_{0<j<J}\left|U_j^{n+1}\right| \leq (2\nu + |1-2\nu|)\,\|U^n\|_\infty\,.$$

The factor $(2\nu + |1-2\nu|)$ will be ≤ 1 if $0 \leq \nu \leq \frac{1}{2}$. It will be > 1 otherwise.

If $\nu \leq \frac{1}{2}$ then the solution is bounded by the data.

The stability condition

The stability condition $\nu \leq \frac{1}{2}$ is equivalent to

$$2\delta t \leq \delta x^2.$$

In terms of the grid dimensions N and J this means

$$N \geq \frac{2T}{(x_R - x_L)^2}J^2.$$

The implication is that if we are close to the stability limit, and we increase J by a factor of two (say), then we will have to increase N by a factor of four.

This means an eight-fold increase in the amount of work and storage.

Is there good reason to increase J?

Error tracking

Write $e_j^n = u_j^n - U_j^n$. Then

$$e_j^{n+1} = \nu e_{j-1}^n + (1-2\nu)e_j^n + \nu e_{j+1}^n + \Delta t T_j^n,$$

where

$$\begin{aligned}
T_j^n &= \frac{1}{\Delta t}\left[u_j^{n+1} - \nu u_{j-1}^n - (1-2\nu)u_j^n - \nu u_{j+1}^n\right] \\
&= \delta_t^+ u_j^n - \delta_x^+ \delta_x^- u_j^n \\
&= \frac{\partial u}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}(x_j, \tau_n) - \frac{\partial^2 u}{\partial x^2} - \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_j, t_n) \\
&= \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}(x_j, \tau_n) - \frac{\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_j, t_n),
\end{aligned}$$

where $\xi_j \in (x_{j-1}, x_{j+1})$ and $\tau_n \in (t_n, t_{n+1})$.

Error bound

So if $\left|\frac{\partial^2 u}{\partial t^2}\right|$ and $\left|\frac{\partial^4 u}{\partial x^4}\right|$ are bounded in our region of interest, and if $2\Delta t \leq \Delta x^2$, we can say that

$$\left|T_j^n\right| \leq C\Delta x^2$$

for some constant C.
If we use Dirichlet boundary conditions, then $e_0^n = e_J^n = 0$. It follows then that

$$\left\|E^{n+1}\right\|_\infty \leq \|E^n\|_\infty + C\Delta t\Delta x^2,$$

so that

$$\max_{0\leq n\leq N} \|E^n\|_\infty \leq CT\Delta x^2.$$

Reducing Δx will reduce the bound on the error.

Explicit finite differences for option pricing

- Transform the payoff and the boundary conditions.
- Solve the heat equation.
- Transform the solution.

If $\nu = \frac{1}{2}$ we recover a binomial method!

More experiments

- Use the above prescription to price a European call option. Compare the values obtained with those produced by the Black-Scholes formula, as computed by `bs.m`.
- Examine the effect of the size of Δt and Δx, and the positions of x_{min} and x_{max} on the accuracy of your solutions.
- Produce a graph that shows the evolution of the error through time.
- Experiment with different boundary conditions.

More on stability

We can write discrete equations for the interior points in matrix-vector form

$$U^{n+1} = (I - \nu A)U^n + b,$$

where

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix}$$

and b incorporates the boundary conditions.

More on stability

Now, as we have seen, we can describe the evolution of the error using an equation of the form

$$E^{n+1} = (I - \nu A)E^n + \Delta t T^n,$$

where T^n is the truncation error. In practice, there is a contribution from rounding errors too on the right hand side, but these are unimportant *if the method is stable.* In this case, the stability is all about the effect of multiplication by $(I - \nu A)$, and this can be described (at least asymptotically) by looking at the *eigenvalues* of $(I - \nu A)$.

Eigenvalues of a tridiagonal matrix

Lemma *The eigenvalues of the $N \times N$ tridiagonal matrix tri(α, β, γ), with α on the diagonal, and β and γ on the super- and sub-diagonals respectively, are*

$$\mu_k = \alpha + 2\beta\sqrt{\frac{\gamma}{\beta}}\cos\frac{k\pi}{N+1}, \quad k = 1, \ldots, N,$$

and the associated eigenvectors have entries

$$\psi_j = \left(\sqrt{\frac{\gamma}{\beta}}\right)^j \sin\frac{jk\pi}{N+1},$$

for $j = 1, \ldots, N$.

Eigenvalues of A

In our case we have a $(J-1) \times (J-1)$ matrix, with $\beta = \gamma = -1$, and $\alpha = 2$. Then the eigenvalues are

$$\mu_j = 2 - 2\cos\frac{j\pi}{J}, \quad j = 1, \ldots, J-1,$$

which can be written

$$\mu_j = 4\sin^2\frac{j\pi}{2J},$$

and the eigenvectors ψ_j have entries

$$(\psi_j)_k = \sin\frac{jk\pi}{J}, \qquad j, k = 1, \ldots, J-1.$$

The eigenvalues of $(I - \nu A)$lie in the interval $(1 - 4\nu, 1)$, with the lowest corresponding to $j = J - 1$. The condition $\nu \leq \frac{1}{2}$ appears here if we want the eigenvalues to all lie in $(-1, 1)$. The eigenvector corresponding to $j = J - 1$ has entries $\sin\frac{(J-1)k\pi}{J}$. These are almost perfectly oscillatory in k.

An implicit method

- Implicit methods are so-called because we cannot just read off the value of U_j^{n+1} from the values of U_j^n. We have to do some work—solve an equation, sometimes only approximately—in order to get at the values of U_j^{n+1}.
- The gain for the pain is simple to state: stability.
- The simplest example is derived by using a backward difference approximation to the time derivative at t^{n+1}:

$$\partial_t u_j^n \approx \delta_t^- u_j^{n+1} := \frac{u_j^{n+1} - u_j^n}{\Delta t}.$$

  The accuracy of the approximation is exactly as before.
- The resulting discrete equations are:

$$-\nu U_{j+1}^{n+1} + (1 + 2\nu)U_j^{n+1} - \nu U_{j-1}^{n+1} = U_j^n, \qquad 0 < j < J.$$

  The equations are completed by imposing boundary conditions at x_L and x_R.

An implicit method: analysis

- If Dirichlet conditions are used, $e_0^{n+1} = e_J^{n+1} = 0$ and so
$$-\nu e_{j+1}^{n+1} + (1+2\nu)e_j^{n+1} - \nu e_{j-1}^{n+1} = e_j^n + \Delta t T_j^{n+1},$$
where
$$T_j^{n+1} = \delta_t^- u_j^{n+1} - \delta_x^- \delta_x^- u_j^{n+1}.$$
We can write this as
$$(I + \nu A)E^{n+1} = E^n + \Delta t T^{n+1}.$$
- The eigenvalues of $(I + \nu A)$ are all in the interval $(1, 1+4\nu)$ and so are all > 1. We have
$$E^{n+1} = (I + \nu A)^{-1}\left[E^n + \Delta t T^{n+1}\right].$$
If we use the scaled l_2-norm defined by $\|v\|_2^2 := \frac{1}{J}\sum_0^J |v_j|^2$, we have
$$\left\|E^{n+1}\right\|_2 \le \|E^n\|_2 + \Delta t \left\|T^{n+1}\right\|_2.$$

Implicit method: convergence

- Notice that the cumulative bound on the error is unconditional. There is no stability-related restriction on the size of the timestep.
- As in the explicit case, we obtain an error estimate of the form
$$\max_{0\le n\le N} \|E^n\|_2 \le C_1 T\Delta t + C_2 T\Delta x^2,$$
for constants C_1 and C_2 involving bounds on u_{tt} and u_{xxxx} respectively. This time the error is in the l_2-norm.

Maximum principle

It is possible to obtain error bounds in the l_∞-norm, using the maximum principle. Essentially, this says that the solution to the discrete error equation is bounded above and below by the data.

Lemma *Suppose that the vectors f and v satisfy*
$$-\alpha v_{j-1} + (1+\alpha+\beta)v_j - \beta v_{j+1} = f_j,$$
for $0 < j < J$, with $\alpha > 0$ and $\beta > 0$, then
$$\min_j v_j \le \min\left(\min_j f_j, v_0, v_J\right)$$
and
$$\max_j v_j \le \max\left(\max_j f_j, v_0, v_J\right).$$

Maximum principle: proof

- We suppose that $\max_j v_j = v_i$, and that $v_i > \max\left(\max_j f_j, v_0, v_J\right)$. Then v_i is a local maximum, and so we must have $v_i \geq v_{i-1}$ and $v_i \geq v_{i-1}$, with inequality in at least one case (by which we mean that if equality holds then a similar statement can be made for i replaced by $i \pm 1$, and equality cannot always hold because eventually we'll reach the boundary).
- But then $f_i = v_i + \alpha(v_i - v_{i-1}) + \beta(v_i - v_{i+1}) > v_i$, so that $v_i < f_i \leq \max_j f_j$. This is a contradiction, and so our initial supposition must be incorrect.
- Similarly, we can suppose that $v_i = \min_j v_j < \min\left(\min_j f_j, v_0, v_J\right)$. We obtain a contradiction in this case in exactly the same way.

Maximum principle: application

- In the case of our error equation, we have $e_0 = e_J = 0$, and so we can conclude from the lemma that

$$\left\|E^{n+1}\right\|_\infty \leq \left\|E^n + \Delta t T^{n+1}\right\|_\infty \leq \|E^n\|_\infty + \Delta t \left\|T^{n+1}\right\|_\infty .$$

- Then

$$\|E^n\|_\infty \leq \left\|E^0\right\|_\infty + \Delta t \sum_{i=1}^{n} \left\|T^i\right\|_\infty .$$

Again, we can say that the truncation error norms are $O(\Delta t + \Delta x^2)$, and so this is the order of the error. Convergence, as Δt and $\Delta x \to 0$, is established. Note that there is now no stability-induced restriction on the relationship between Δt and Δx.

The θ-method

$$\theta\nu U_{j+1}^{n+1} + (1 - 2\theta\nu)U_j^{n+1} + \theta\nu U_{j-1}^{n+1} = (1-\theta)\nu U_{j+1}^n + (1 - 2(1-\theta)\nu)U_j^n + (1-\theta)\nu U_{j-1}^n.$$

- If $\theta = 0$ this is the explicit method. If $\theta > 0$, it is implicit, and we solve the tridiagional system on the left-hand side in order to complete the timestep from U^n to U^{n+1}.
- The method is stable if $\theta \geq \frac{1}{2} - \frac{1}{4\nu}$.
- If $\theta = \frac{1}{2}$ the stability does not depend on ν, and the trunctation error is $O(\Delta t^2 + \Delta x^2)$. The Crank-Nicholson method corresponds to $\theta = \frac{1}{2}$, and so is stable whatever the value of ν.

Implementation

With implicit methods, equations must be solved in order to get U_j^{n+1} from $\{U_j^n\}$. Typically, this is a tridiagonal system, and it is possible to invert tridiagonal systems particularly simply. The Thomas algorithm is a version of Gaussian elimination: suppose that $c_i v_{i-1} + a_i v_i + b_i v_{i+1} = f_i, \quad 1 \le i \le J$, and that v_0 and v_{J+1} are known. Then we perform the following:

$f_1 \leftarrow f_1 - a_1 v_0$
$f_J \leftarrow f_J - a_J v_{J+1}$
for $j = 1$ ***to*** $J - 1$
  $a_{j+1} \leftarrow a_{j+1} - b_j c_{j+1}/a_j$
  $f_{j+1} \leftarrow f_{j+1} - f_j c_{j+1}/a_j$
end
$v_J \leftarrow f_J/a_J$
for $j = J - 1$ ***to*** 1
  $v_j \leftarrow (f_j - b_j v_{j+1})/a_j$
end

American option pricing

Objectives

- Write the pricing equations for an american option as a linear complimentarity problem.
- Apply implicit finite difference methods to solve the problem.
- Understand how projected successive over-relaxation works.

American options

- The american option differs from its european counterpart in that the holder has the freedom to exercise the option at any time prior to the expiry. On exercise, the holder receives a cashflow equivalent to the value of the payoff function $\Lambda(S)$ at that point in time. Thus the option value always satisfies

$$V_{\text{am}}(S,t) \ge \Lambda(S).$$

- Another inequality is suggested by the fact that the holder can always wait until expiry to exercise. This strategy effectively reduces the american option to a european one, and so

$$V_{\text{am}}(S,t) \ge V_{\text{eur}}(S,t).$$

American options

- Consider the case of a put option. At some point, the value of the european put dips below the payoff function, but this is not true of the american version. In this case there will be a first contact point $S_f(t)$ such that

$$V_{\text{am}}(S,t) > K - S \quad \text{for } S > S_f(t),$$

and

$$V_{\text{am}}(S,t) = K - S \quad \text{for } S \leq S_f(t).$$

We will find that $V_{\text{am}}(S,t)$ must satisfy the *smooth-pasting* condition at such a boundary:

$$\frac{\partial V_{\text{am}}}{\partial S}(S_f(t),t) = \frac{\mathrm{d}\Lambda}{\mathrm{d}S}(S_f(t)).$$

American options

- By mirroring the derivation of the Black-Scholes equation, taking a portfolio $\Pi = V - \Delta S$, we find that that holder of the option can make a guaranteed profit *in excess* of the risk-free rate of return unless

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \leq 0.$$

- Arbitrage considerations mean that this inequality must be satisfied. We have a linear complimentarity problem: the optimal solution satisfies the above inequalities, and also either:
  - the option is equal to the payoff.
  - the option satisfies the Black-Scholes equation.

American options

- We'll consider this problem in the transformed setting:

$$u(x,t) = \exp\left(\hat{r}_- x + \hat{r}_+^2 t\right) V\left(S, T - 2t/\sigma^2\right),$$

where

$$S = Ke^x, \quad \text{and} \quad \hat{r}_\pm = \frac{r}{\sigma^2} \pm \frac{1}{2},$$

and K is the strike price (or some other parameter associated with the payoff).

- Then if $g(x,t) = \exp\left(\hat{r}_- x + \hat{r}_+^2 t\right)\Lambda(S)$, we have, at the point (x,t) with $t > 0$, either

$$u_t - u_{xx} \geq 0 \quad \text{and} \quad u = g$$

or

$$u_t - u_{xx} = 0 \quad \text{and} \quad u \geq g.$$

American options

- These can be combined in the form:

$$(u-g)(u_t-u_{xx})=0$$
$$u_t-u_{xx}\ge 0, \qquad u-g\ge 0.$$

- If we consider a fully-implicit ($\theta = 1$) finite-difference approximation to the partial derivatives, we have

$$(U_j^{n+1}-g_j^{n+1})(U_j^{n+1}-U_j^n-\nu(U_{j+1}^{n+1}-2U_j^{n+1}+U_{j-1}^{n+1}))=0$$
$$U_j^{n+1}-\nu(U_{j+1}^{n+1}-2U_j^{n+1}+U_{j-1}^{n+1})\ge U_j^n, \quad U_j^{n+1}-g_j^{n+1}\ge 0.$$

- When the boundary conditions are incorporated, either in the vector b which is based on U^n or in the matrix A, we can write the above system in the vectorized form: find U^{n+1} satisfying

$$(U^{n+1}-g^{n+1})^T(AU^{n+1}-b)=0$$
$$AU^{n+1}-b\ge 0, \qquad U^{n+1}-g^{n+1}\ge 0.$$

American options

- We can't just this system by solving $AU^{n+1}-b=0$, because the second inequality may or may not be satisfied. We have to solve the whole problem at once, and to do this an iterative method is most appropriate.
- Write $U^{n+1}=g^{n+1}+x$, so that we are looking for a vector x satisfying

$$x^T(Ax-\hat{b})=0$$
$$Ax-\hat{b}\ge 0, \qquad x\ge 0,$$

where $\hat{b}=b-Ag^{n+1}$.
- We rewrite this by splitting A into its lower, diagonal and upper-triangular parts $A=D+L+U$, and writing $(D+L)x\ge\hat{b}-Ux$ so that

$$(D+L)x\ge Dx+[\hat{b}-(D+U)x].$$

American options

- We turn this into an iteration. Given a current 'guess' $x^{(k)}$, we can generate $x^{(k+1)}$ by solving

$$Dx^{(k+1)}\ge Dx^{(k)}+[\hat{b}-(D+U)x^{(k)}-Lx^{(k+1)}].$$

- We can ensure the condition $x^{(k+1)}\ge 0$ as well by setting

$$x^{(k+1)}=\max\left(0, x^{(k)}+D^{-1}[\hat{b}-(D+U)x^{(k)}-Lx^{(k+1)}]\right).$$

- Notice that $x^{(k+1)}$ appears on the right-hand side. If we solve this equation by sweeping from left to right, then the $(k+1)$ values on the right-hand side are values we know.
- If we freeze our iteration at any point, perhaps mid-way through a sweep, our current solution vector will be of the form

$$(x_1^{(k+1)},\dots,x_j^{(k+1)},x_{j+1}^{(k)},\dots,x_{J-1}^{(k)}).$$

American options

- Then we have
$$x_j^{(k+1)} = \max\left(0, x_j^{(k)} + A_{jj}^{-1}[\hat{b}_j - A_{jj}x_j^{(k)} - A_{jj+1}x_{j+1}^{(k)} - A_{jj-1}x_{j-1}^{(k+1)}]\right).$$
- *Overrelaxation* involves a similar iteration:
$$x_j^{(k+1)} = \max\left(0, x_j^{(k)} + \omega A_{jj}^{-1}[\hat{b}_j - A_{jj}x_j^{(k)} - A_{jj+1}x_{j+1}^{(k)} - A_{jj-1}x_{j-1}^{(k+1)}]\right).$$
- What is the effect of this replacement? We can rewrite the problem we are solving in the following way. Find a vector $x \geq 0$ satisfying $Ax - \hat{b} \geq 0$, that minimises the quadratic function
$$G = \frac{1}{2}x^T Ax - \hat{b}^T x.$$
- It turns out that G is reduced at each stage of the iteration, so long as $\omega \in (0, 2)$.