The explicit finite-difference method in matrix terms

André Ran
Vrije Universiteit Amsterdam

The explicit finite difference method

Let us put the equation

$$u_n^{m+1} = \alpha u_{n+1}^m + (1-2\alpha)u_n^m + \alpha u_{n-1}^m$$

with its initial and boundary conditions in matrix-vector form.
Put

$$u^m = \begin{pmatrix} u_{-N^-+1}^m \\ \vdots \\ u_{N^+-1}^m \end{pmatrix}$$

This vector contains the u-values on the grid that are at time level $m \cdot \delta\tau$, *but not* on the boundary.
Put $k = N^+ + N^- - 1$. Then $u^m \in \mathbb{R}^k$.

To compute $u^{m+1}_{-N^-+1}$ and $u^{m+1}_{N^+-1}$ we need to use the boundary points, that is, we need the vector

$$b^m = \begin{pmatrix} u^m_{-N^-} \\ 0 \\ \vdots \\ 0 \\ u^m_{N^+} \end{pmatrix}.$$

This should be understood correctly: we want b^m and u^m to be vectors of the same size k!

With these notations

$$u_n^{m+1} = \alpha u_{n+1}^m + (1-2\alpha)u_n^m + \alpha u_{n-1}^m$$

can be rewritten as

$$u^{m+1} = (I - \alpha T)u^m + \alpha b^m$$

where $\alpha > 0$ and T is the symmetric tridiagonal Toeplitz matrix

$$T = \begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

Iteration

We can now iteratively compute, as long as we have u^0, which is the given initial condition.

Thus:

$$
\begin{aligned}
u^1 &= (I - \alpha T)u^0 + b^0, \\
u^2 &= (I - \alpha T)u^1 + b^1 = (I - \alpha T)^2 u^0 + (I - \alpha T)b^0 + b^1, \\
&\vdots \\
u^{m+1} &= (I - \alpha T)^m u^m + \sum_{j=0}^{m} (I - \alpha T)^j b^{m-j}.
\end{aligned}
$$

Error propagation

Assuming we make no errors in b^m, errors we make at time step m, which are denoted by e^m propagate as

$$e^{m+1} = (I - \alpha T)e^m.$$

Iterating this: errors we make at time 0 have at time $m + 1$ the effect

$$e^{m+1} == (I - \alpha T)^{m+1}e^0.$$

Eigenvalues

Since $T = T^\top$, we have that $I - \alpha T = UDU^{-1}$, where U is an orthogonal matrix, D is diagonal with real numbers on the diagonal. Then

$$(I - \alpha T)^{m+1} = UD^{m+1}U^{-1}$$

Now the right hand side will go to zero as $m \to \infty$ if and only if all eigenvalues of $I - \alpha T$ are in the interval $(-1, 1)$.
Note: since T is positive definite (i.e., $\langle Tx, x\rangle > 0$ for all non-zero x) and $\alpha > 0$ all eigenvalues of $I - \alpha T$ are in $(-\infty, 1)$.
Thus we need to analyse for which values of α the eigenvalues are in $(-1, 1)$ and for which values there is an eigenvalue less than -1.

Stability

Recall $k = N^+ + N^- - 1$. Also recall $\alpha = \frac{\delta\tau}{\delta x^2} > 0$.

Theorem

- *For all $\alpha < \frac{1}{2}$ the eigenvalues of $I - \alpha T$ are in $(-1, 1)$.*
- *For k large enough (i.e., δx small enough) and for $\alpha > \frac{1}{2}$ the matrix $I - \alpha T$ has an eigenvalue below -1.*

To show the first part use that

$$4I - T = \begin{pmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 2 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & 2 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & 2 \end{pmatrix}.$$

Like T itself, the latter matrix is positive definite. Thus the eigenvalues of T are in the interval $(0, 4)$. So, the eigenvalues of $I - \alpha T$ are in the interval $(1 - 4\alpha, 1)$. Hence, for $\alpha < \frac{1}{2}$ they are in the interval $(-1, 1)$.

To show the second part we need the following result from linear algebra: If $A = A^\top$ then

$$|\lambda_{\max}(A)| = \max_{x \neq 0} \frac{|\langle Ax, x\rangle|}{\langle x, x\rangle}.$$

We apply this with $A = I - \alpha T$ and for k odd, with x given by

$$x = \begin{pmatrix} 1 \\ -1 \\ 1 \\ \vdots \\ -1 \\ 1 \end{pmatrix}.$$

Thus, we need to show that the quotient on the right for this vector x is above 1 for $\alpha > \frac{1}{2}$ and k large enough.

Then

$$(I-\alpha T)x = \begin{pmatrix} 1-3\alpha \\ 4\alpha-1 \\ 1-4\alpha \\ \vdots \\ 4\alpha-1 \\ 1-3\alpha \end{pmatrix}.$$

So

$$\begin{aligned} \langle (I-\alpha T)x, x\rangle &= 2-6\alpha+(k-2)(1-4\alpha) \\ &= k(1-4\alpha)+2\alpha, \\ \langle x, x\rangle &= k. \end{aligned}$$

Thus

$$\frac{|\langle (I-\alpha T)x, x\rangle|}{\langle x, x\rangle} = \left|1 - 4\alpha + \frac{2\alpha}{k}\right|.$$

For $\alpha > \frac{1}{2}$ this is equal to $4\alpha - 1 - \frac{2\alpha}{k}$. We have to show that for $\alpha > \frac{1}{2}$ this is larger than 1 for k large enough. Since $\alpha > \frac{1}{2}$ we have $4\alpha - 1 > 1$, and hence for k large enough we see that is indeed larger than 1.

Conclusions

This seems not so bad. After all, just keep α below $\frac{1}{2}$ and the algorithm will work wonderfully. However, it is **BAD NEWS**. Remember

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

So, if we want to improve on the calculations by choosing ten times as many data points in the x-direction (which in financial terms is the variable S), the effect of that is to multiply α by 100. In order to compensate for that, to keep α the same, we need to divide $\delta\tau$ by 100. Thus, if we are prepared to do ten times as much work in the x-direction, we end up having to do thousand times as much work.

Examples

In the examples we compute the European put, with $T = 0.5, \sigma = 0.4, r = 0.02, E = 20$. The light blue line is the exact value. The dark blue line is the approximation using the explicit finite difference method. In all figures the left hand graph is the graph of the put, the right hand graph is the graph of its derivative, the Δ. In all cases we compute using an algorithm that solves the heat equation between $x = -2$ and $x = 1.5$.

In the first three graphs we take $\delta\tau = 10^{-4}$, and vary δx. The values for δx are, in this order $\delta x = 0.05$, $\delta x = 0.02$, $\delta x = 0.0141$. The corresponding values for α are, respectively, $\alpha = 0.04$, $\alpha = 0.25$, and $\alpha = 0.503$. We see that in the first two graphs the value of the put is nicely approximated (actually indistinguishable from the exact value), but the approximation of Δ is poor, although it improves with decreasing δx. Once $\alpha > 0.5$, however, the algorithm quickly brakes down, as is seen in the third graph.

Nice approximation, $\alpha < \frac{1}{2}$

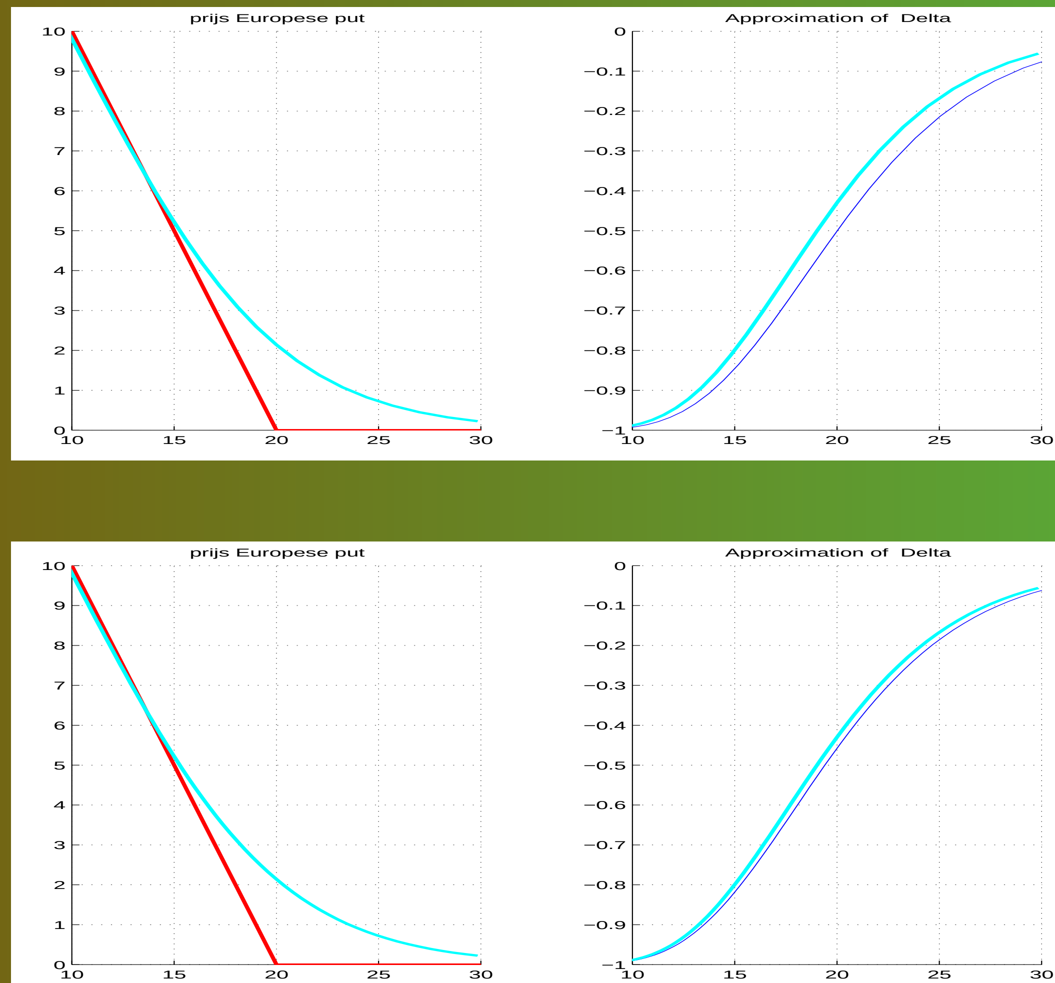

Figure 1: $\delta x = 0.05$ (above), $\delta x = 0.02$ (below), $\delta\tau = 10^{-4}$

Bad approximation, $\alpha > \frac{1}{2}$

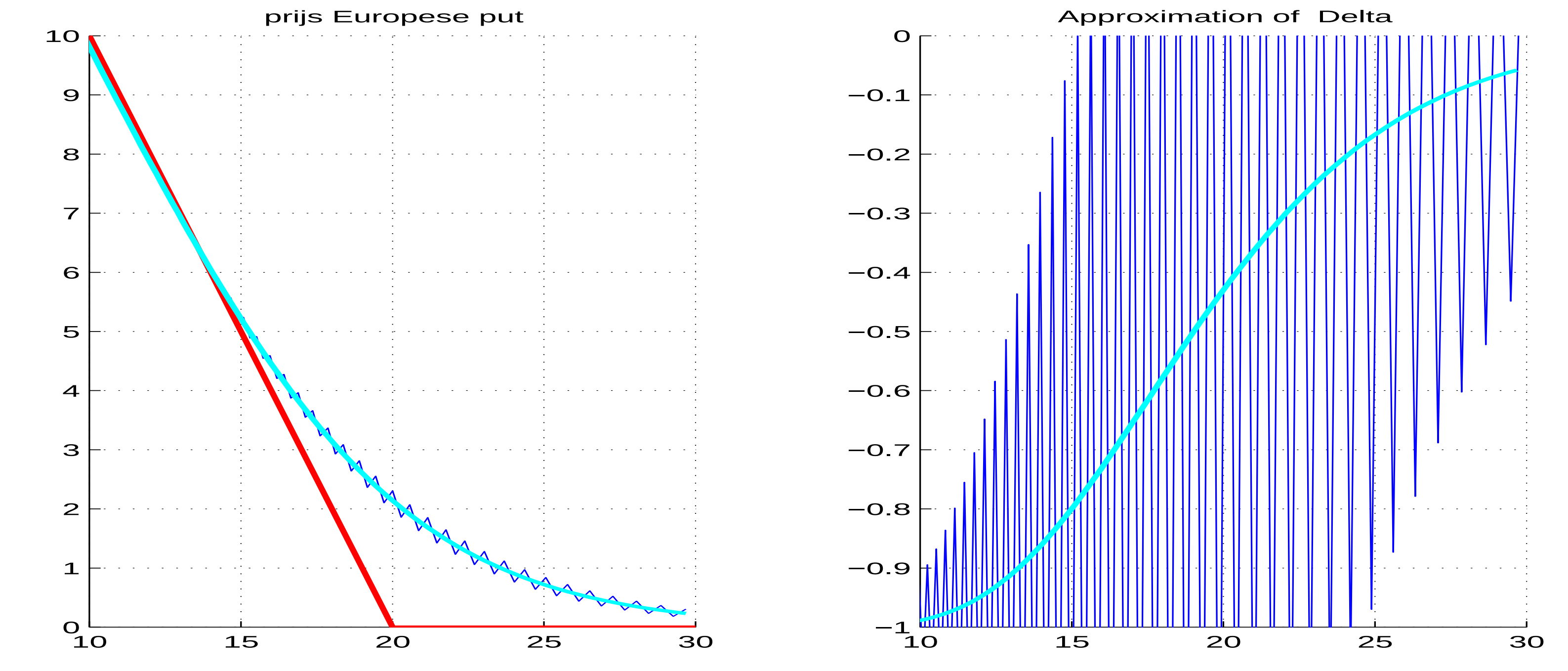

Figure 2: $\delta x = 0.0141$, $\delta \tau = 10^{-4}$

To get a good approximation going down to $\delta x = 0.01$, we now change $\delta\tau$ to $0.49 \cdot 10^{-4}$. This gives the following result.

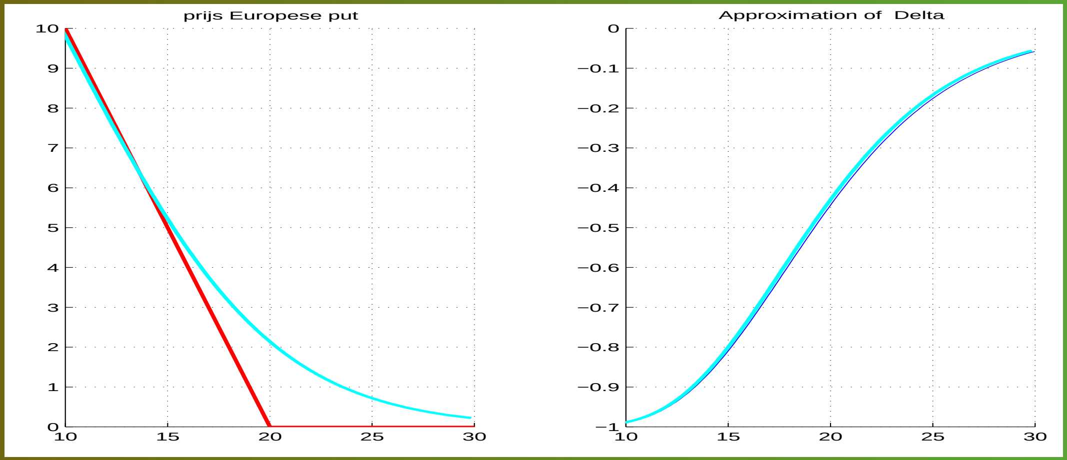

Figure 3: $\delta x = 0.01$