X. Numerical methods: Pricing European options

Beáta Stehlíková

Financial derivatives

Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava

Transformation to a heat equation

- Transformation

$$V(S,t) = e^{-\alpha x - \beta \tau} u(x,\tau),$$

$$\alpha = \frac{r-q}{\sigma^2} - \frac{1}{2}, \beta = \frac{r+q}{2} + \frac{\sigma^2}{8} + \frac{(r-q)^2}{2\sigma^2}, \tau = T - t, x = \ln(S/E),$$

transforms the Black-Scholes equation to the following heat equation:

$$\frac{\partial u}{\partial \tau} - \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2} = 0$$

for $x \in \mathbb{R}, \tau \in [0, T]$

- Initial condition: $u(x,0) = g(x)$
  - call option: $g(x) = Ee^{\alpha x + \beta \tau} \max(e^x - 1, 0)$
  - put option: $g(x) = Ee^{\alpha x + \beta \tau} \max(1 - e^x, 0)$

Boundary conditions

- For a numerical scheme we also need boundary condition - we need to think of the option value for very small and very large stock prices
- Call option:
  - $V(0,t) = 0$
  - for $S \to \infty$ we have: $V(S,t) \sim Se^{-q(T-t)}$, more precisely: $V(S,t) \sim Se^{-q(T-t)} - Ee^{-r(T-t)}$
- Put option:
  - $V(0,t) = Ee^{-r(T-t)}$
  - $V(S,t) \to 0$ for $S \to \infty$

Approximation of the solution

- Numerical solution on a bounded space interval $x \in [-L, L]$
- Grid points - in time and space:

$$x_i = ih, \quad i = -n, ..., -2, -1, 0, 1, 2, ...n,$$

$$\tau_j = jk, \ j = 0, 1, ..., m.$$

where $h = L/n, k = T/m$

- Approximation of the solution u in the point (x_i, τ_j) will be denoted by

$$u_i^j \approx u(x_i, \tau_j), \qquad g_i^j \approx g(x_i, \tau_j)$$

Approximation of the solution

- Boundary conditions:
  - call option:

$$\begin{aligned} \phi^j &:= u^j_{-N} = 0 \\ \psi^j &:= u^j_N = Ee^{(\alpha+1)Nh+(\beta-q)jk} \end{aligned}$$

  - put option:

$$\begin{aligned} \phi^j &:= u^j_{-N} = Ee^{-\alpha Nh+(\beta-r)jk} \\ \psi^j &:= u^j_N = 0 \end{aligned}$$

Implicit scheme

- Recall from the numerical methods course: explicit and implicit scheme for a heat equation
- Implicit scheme - can be written as:
  $-\gamma u_{i-1}^j + (1+2\gamma)u_i^j - \gamma u_{i+1}^j = u_i^{j-1}$, where $\gamma = \frac{\sigma^2 k}{2h^2}$,
- In a matrix form: $\mathbf{A}u^j = u^{j-1} + b^{j-1}$ for $j = 1, 2, \ldots, m$ where

$$\mathbf{A} = \begin{pmatrix} 1+2\gamma & -\gamma & 0 & \cdots & 0 \\ -\gamma & 1+2\gamma & -\gamma & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & -\gamma & 1+2\gamma & -\gamma \\ 0 & \cdots & 0 & -\gamma & 1+2\gamma \end{pmatrix},$$

$$b^j = (\gamma\phi^{j+1}, 0, \ldots, 0, \gamma\psi^{j+1})^T$$

Solving the linear system

- The system $\mathbf{A}x = b$ with the matrix

$$\mathbf{A} = \begin{pmatrix} 1+2\gamma & -\gamma & 0 & \cdots & 0 \\ -\gamma & 1+2\gamma & -\gamma & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & -\gamma & 1+2\gamma & -\gamma \\ 0 & \cdots & 0 & -\gamma & 1+2\gamma \end{pmatrix}$$

- Firstly - we solve it using Gauss-Seidel method
- Afterwards - its generalization, SOR method (its modification will be used in a scheme for American options)

XI. Numerical methods: System of linear equations arising from the implicit scheme

Beáta Stehlíková

Financial derivatives

Faculty of Mathematics, Physics and Informatics

Comenius University, Bratislava

Gauss-Seidel method: example

- Consider the system

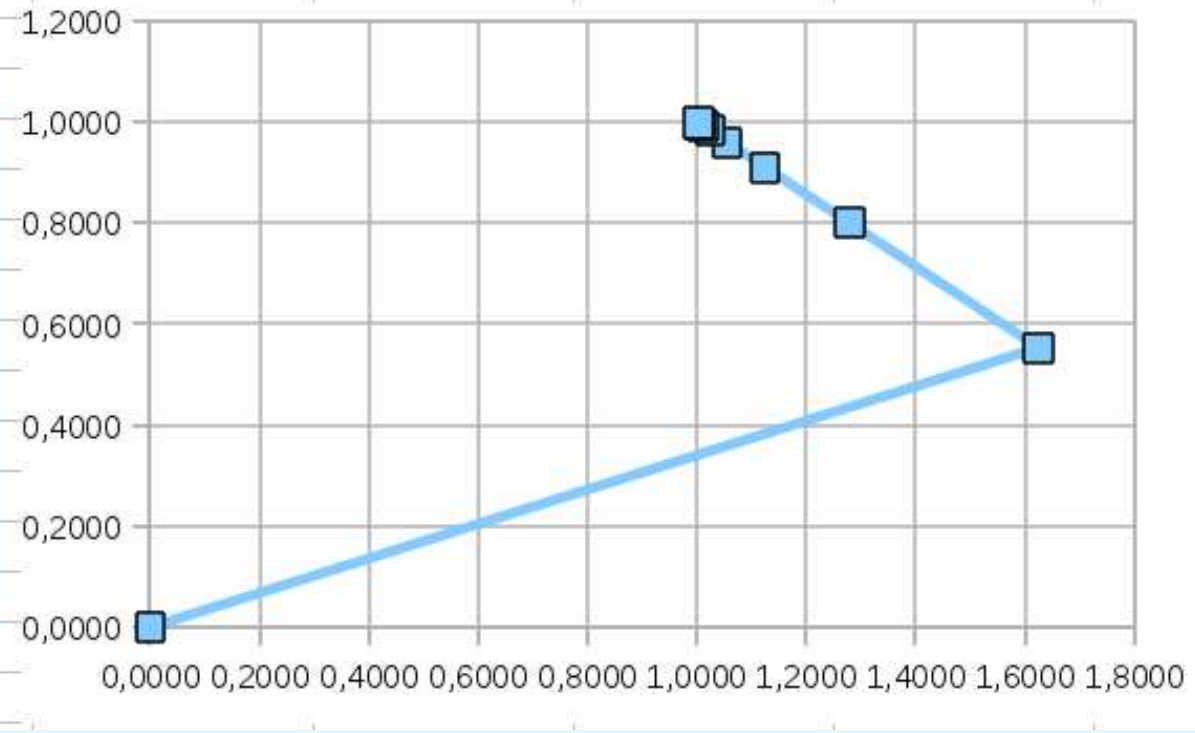
$$\begin{pmatrix} 8 & 5 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 13 \\ 12 \end{pmatrix},$$

  which has the exact solution $(x_1, x_2)' = (1, 1)'$

- We will use Gauss-Seidel method and start from $(0, 0)'$

Gauss-Seidel method: example

matrix A		right hand side b
8,0000	5,0000	13,0000
5,0000	7,0000	12,0000
0,0000	0,0000	
1,6250	0,5536	
1,2790	0,8007	
1,1246	0,9110	
1,0556	0,9603	
1,0248	0,9823	
1,0111	0,9921	
1,0049	0,9965	
1,0022	0,9984	
1,0010	0,9993	
1,0004	0,9997	



OBSERVATION: It converges, but slowly, because it make too short steps, although in the right direction $\Rightarrow$ motivation for a modification of the method

Gauss-Seidel method: example

- Gauss-Seidel method - by coordinates:

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right]$$

- When computing $x_i^{(k)}$, we have
  - current approximation of the solution:

$$\left(x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)} \right)$$

  - residual, i.e., the difference $b - \mathbf{A}x^{ap}$:

$$\mathbf{r}_i^{(k)} = \left(r_{1i}^{(k)}, \ldots, r_{ni}^{(k)} \right)$$

Modification of the Gauss-Seidel method

- Gauss-Seidel computation of $x_i^{(k)}$ can be written as:

$$\begin{aligned} x_i^{(k)} &= \frac{1}{a_{ii}} \left[b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right] \\ &= x_i^{(k-1)} + \frac{1}{a_{ii}} r_{ii}^{(k)} \end{aligned}$$

- Modification of the Gauss-Seidel method according to our earlier observation, so called successive over-relaxation (SOR) method:

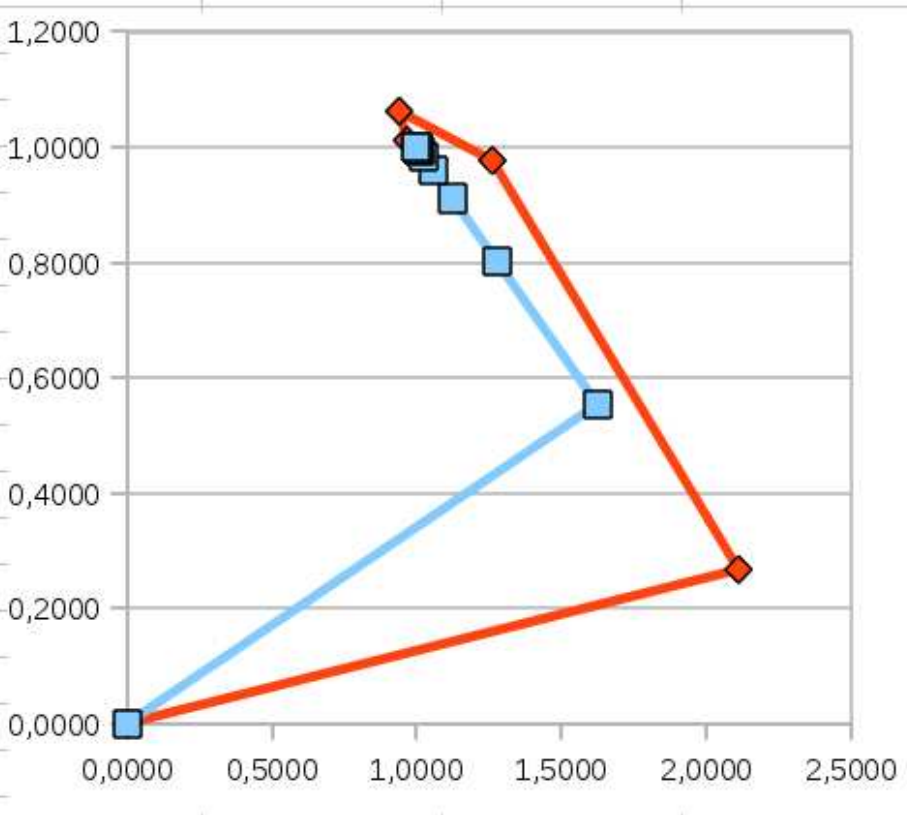
$$x_i^{(k)} = x_i^{(k-1)} + \omega \, \frac{1}{a_{ii}} r_{ii}^{(k)}$$

for $\omega > 1$ (for $0 < \omega < 1$ it is, in fact, under-relaxation; for $\omega = 1$ we have the original Gauss-Seidel method)

SOR method: example

Example for the beginning of the lecture; this time we use SOR with $\omega = 1.3$:

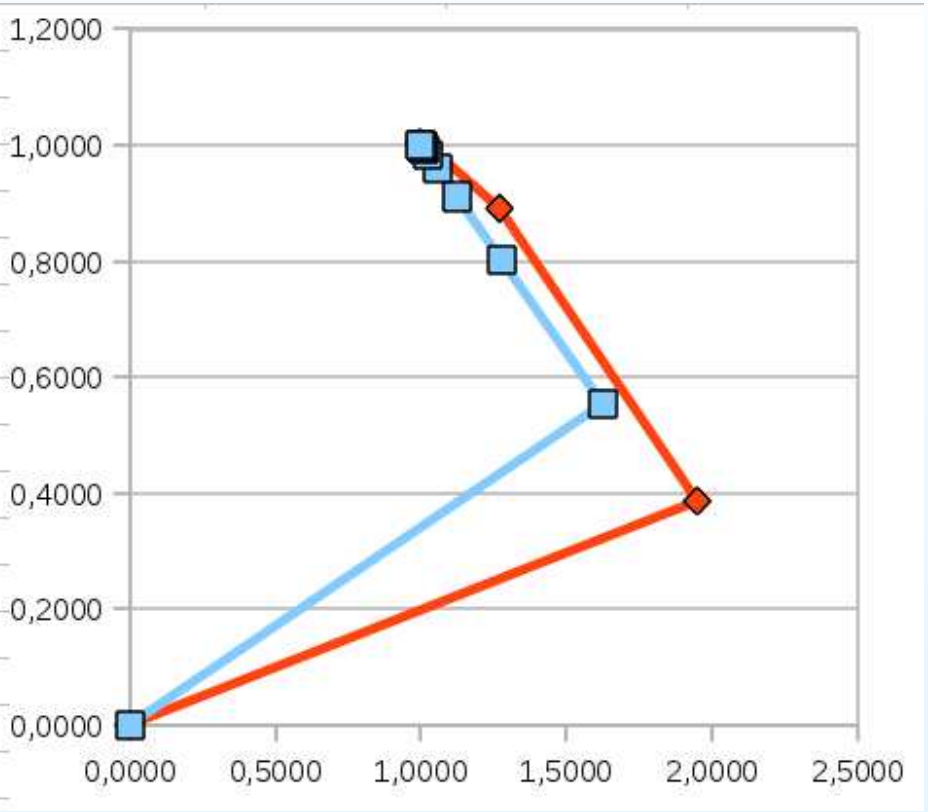
omega	1,3000
0,0000	0,0000
2,1125	0,2670
1,2618	0,9768
0,9403	1,0624
0,9672	1,0117
1,0003	0,9962
1,0030	0,9984
1,0004	1,0001
0,9998	1,0002
0,9999	1,0000
1,0000	1,0000



SOR method: example

Even better convergence for $\omega = 1.2$:

omega	1,2000
0,0000	0,0000
1,9500	0,3857
1,2707	0,8908
1,0277	0,9981
0,9959	1,0039
0,9979	1,0010
0,9997	1,0001
1,0000	1,0000
1,0000	1,0000
1,0000	1,0000
1,0000	1,0000



SOR method: example

Comparison of Gauss-Seidel (left) and SOR method for $\omega = 1.2$ (right):

		omega	1,2000
0,0000	0,0000	0,0000	0,0000
1,6250	0,5536	1,9500	0,3857
1,2790	0,8007	1,2707	0,8908
1,1246	0,9110	1,0277	0,9981
1,0556	0,9603	0,9959	1,0039
1,0248	0,9823	0,9979	1,0010
1,0111	0,9921	0,9997	1,0001
1,0049	0,9965	1,0000	1,0000
1,0022	0,9984	1,0000	1,0000
1,0010	0,9993	1,0000	1,0000
1,0004	0,9997	1,0000	1,0000

QUESTION:
How to choose ω?
What are the factors influencing the speed of convergence?

Iteration schemes: speed of convergenec

- Consider a general iteration scheme $x^{(k+1)} = \mathbf{T}x^{(k)} + g$ with exact solution x^*, to which the scheme converges
- We have:

$$\begin{aligned}
\|x^{(k)} - x^*\| &= \|(\mathbf{T}x^{(k-1)} + g) - (\mathbf{T}x^* + g)\| \\
&= \|\mathbf{T}\left[x^{(k-1)} - x^*\right]\| \\
&= \|\mathbf{T}\left[(\mathbf{T}x^{(k-1)} + g) - (\mathbf{T}x^* + g)\right]\| \\
&= \|\mathbf{T}^2\left[x^{(k-2)} - x^*\right]\| \\
&\dots \\
&= \|\mathbf{T}^k\left[x^{(0)} - x^*\right]\| \leq \|\mathbf{T}^k\|\,\|x^{(0)} - x^*\|
\end{aligned}$$

- We need to estimate the norm $\|\mathbf{T}^k\|$
- We use spectral radius of a matrix and its properties

Spectral radius and matrix norms

- Let $\mathbf{M}$ be a square matrix
- Spectral radius of this matrix:

$$\rho(\mathbf{M}) = \max |\lambda_i| \,,$$

  where λ_i are eigenvalues of $\mathbf{M}$
- Relation of the spectral radius and matrix norms:
  - the followig holds:

$$\lim_{n\to\infty} \|\mathbf{M}^n\|^{1/n} = \rho(\mathbf{M})$$

  - therefore for large n we can use an approximation

$$\|\mathbf{M}^n\|^{1/n} \sim \rho(\mathbf{M}) \Rightarrow \|\mathbf{M}^n\| \sim \rho(\mathbf{M})^n$$

Iteration schemes: speed of convergence

- Therefore we have:

$$\|x^{(k)} - x^*\| \leq \|\mathbf{T}^k\| \, \|x^{(0)} - x^*\| \sim \rho(\mathbf{T})^k \|x^{(0)} - x^*\|$$

- Hence the spectral radius of the matrix $\mathbf{T}$
  - has to be less than 1 - so that the error converges to zero and the method converges
  - should be as small as possible - to have the speed of convergence as high as possible

Iter. schemes: speed of convergence - example

- We had the system:

$$\begin{pmatrix} 8 & 5 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 13 \\ 12 \end{pmatrix},$$

- SOR method in matrix form $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$:

$$x^{(k+1)} = (\mathbf{D} + \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} - \omega\mathbf{U}]x^{(k)} + \omega(\mathbf{D} + \omega\mathbf{L})^{-1}b$$

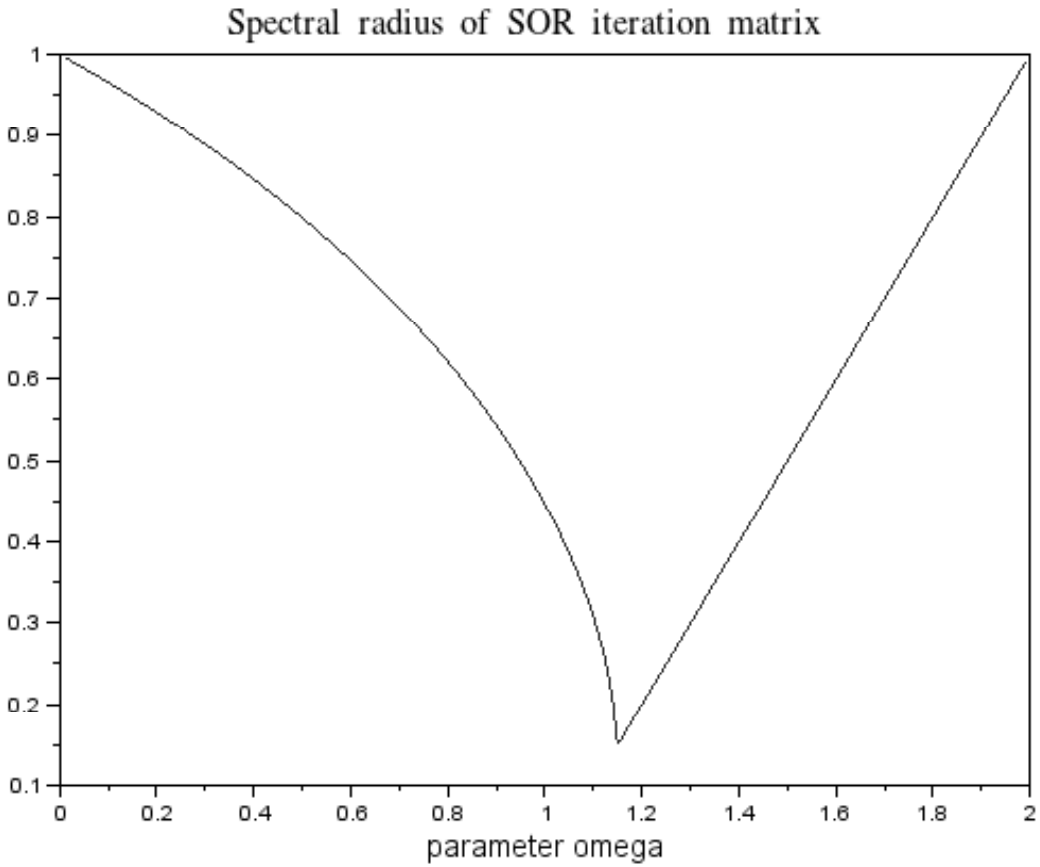
- Hence, in our case the iteration matrix is

$$\mathbf{T} = \begin{pmatrix} 8 & 0 \\ 5\omega & 7 \end{pmatrix}^{-1} \begin{pmatrix} 8(1-\omega) & -5\omega \\ 0 & 7(1-\omega) \end{pmatrix}$$

and we need the dependence of the maximal absolute value of an eigenvalue on ω

Iter. schemes: speed of convergence - example

Result:



We have used: $\omega = 1$ (Gauss-Seidel), $\omega = 1.3$, $\omega = 1.2$

Iteration schemes: speed of convergence

Some useful theorems (without proofs):

1. [Kahan] Let $a_{ii} \neq 0$. Then $\rho(\mathbf{T}_\omega) \geq |\omega - 1|$.
2. [Ostrowski-Reich] Let A be positive definite and let $0 < \omega < 2$. Then the SOR converges for any initial point.
3. Let A be a positive definite tridiagonal matrix. Then $\rho(\mathbf{T}_{gs}) = \rho(\mathbf{T}_j)^2$ and the optimal choice of parameter ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(\mathbf{T}_j)]^2}}.$$

For this choice, we have: $\rho(\mathbf{T}_\omega) = \omega - 1$.

Notation:

$\mathbf{T}_j, \mathbf{T}_{gs}, \mathbf{T}_\omega$ - iteration matrices of Jacobi, Gauss-Seidel and SOR methods.

With the decomposition $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$ we express $\mathbf{T}_j$ as $-\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$

SOR method: speed of convergence

Corollaries:

- From Theorem 1: for $\omega \notin (0, 2)$ we have $\rho(\mathbf{T}_\omega) \geq 1$. Condition $\omega \in (0, 2)$ is therefore a necessary condition for convergence.
- Theorem 2 gives a class of matrices, for which the condition $\omega \in (0, 2)$ is also a sufficient condition for convergence.

Theorem 3 and our example

The matrix of our system is positive definite and tridiagonal:

$$\begin{pmatrix} 8 & 5 \\ 5 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 13 \\ 12 \end{pmatrix}$$

JACOBI METHOD:

- iteration matrix:

$$\mathbf{T}_j = -\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}) = -\begin{pmatrix} 8 & 0 \\ 0 & 7 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 5 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{5}{8} \\ -\frac{5}{8} & 0 \end{pmatrix}$$

- eigenvalues of $\mathbf{T}_j$: $\lambda_1 = \frac{5}{2\sqrt{14}}$, $\lambda_1 = -\frac{5}{2\sqrt{14}}$
- spectral radius: $\rho(\mathbf{T}_j) = \frac{5}{2\sqrt{14}}$

Theorem 3 and our example

GAUSS-SEIDEL METHOD:

- iteration matrix:

$$\mathbf{T}_{gs} = -(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U} = -\begin{pmatrix} 8 & 0 \\ 5 & 7 \end{pmatrix}^{-1}\begin{pmatrix} 0 & 5 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{5}{8} \\ 0 & \frac{25}{56} \end{pmatrix}$$

- eigenvalues of $\mathbf{T}_{gs}$: $\lambda_1 = 0$, $\lambda_1 = \frac{25}{56}$
- spectral radius: $\rho(\mathbf{T}_{gs}) = \frac{25}{56}$

INDEED WE HAVE:

$$\begin{aligned} \rho(\mathbf{T}_{gs}) &= \rho(\mathbf{T}_j)^2 \\ \frac{25}{56} &= \left(\frac{5}{2\sqrt{14}}\right)^2 \end{aligned}$$

Theorem 3 and our example

SOR METHOD:

- Optimal value of parameter ω according to Theorem 3:

$$\omega = \frac{2}{1+\sqrt{1-[\rho(\mathbf{T}_j)]^2}} = \frac{2}{1+\sqrt{1-\frac{25}{56}}} = \frac{2\sqrt{56}}{\sqrt{56}+\sqrt{31}} \approx 1.147$$

- Corresponding value of spectral radius of the iteration matrix according to Theorem 3:

$$\rho(\mathbf{T}_\omega) = \omega - 1 = \frac{\sqrt{56}-\sqrt{31}}{\sqrt{56}+\sqrt{31}} \approx 0.147$$

Theorem 3 and our example

- We compute the spectral radius for the given ω directly:

```
(%i1) omega:2*sqrt(56)/(sqrt(56)+sqrt(31)) $

      m1:invert(matrix([8,0],[5*omega,7])) $
      m2:matrix([8*(1-omega),-5*omega],[0,7*(1-omega)]) $
      t:m1.m2;
```

$$(\%o4)\quad \begin{bmatrix} 1-\frac{4\sqrt{14}}{\sqrt{31}+2\sqrt{14}} & -\frac{5\sqrt{14}}{2\left(\sqrt{31}+2\sqrt{14}\right)} \\ -\frac{40\left(1-\frac{4\sqrt{14}}{\sqrt{31}+2\sqrt{14}}\right)}{\sqrt{14}\left(\sqrt{31}+2\sqrt{14}\right)} & -\frac{4\sqrt{14}}{\sqrt{31}+2\sqrt{14}}+\frac{100}{\left(\sqrt{31}+2\sqrt{14}\right)^2}+1 \end{bmatrix}$$

```
(%i5) eigenvalues(t);
```

$$(\%o5)\quad \left[\left[\frac{25}{4\sqrt{14}\sqrt{31}+87}\right],[2]\right]$$

Hence we obtain: $\rho(\mathbf{T}_\omega) = \frac{25}{4\sqrt{14}\sqrt{31}+87}$

Theorem 3 and our example

- This agrees with the result given in Theorem 3:

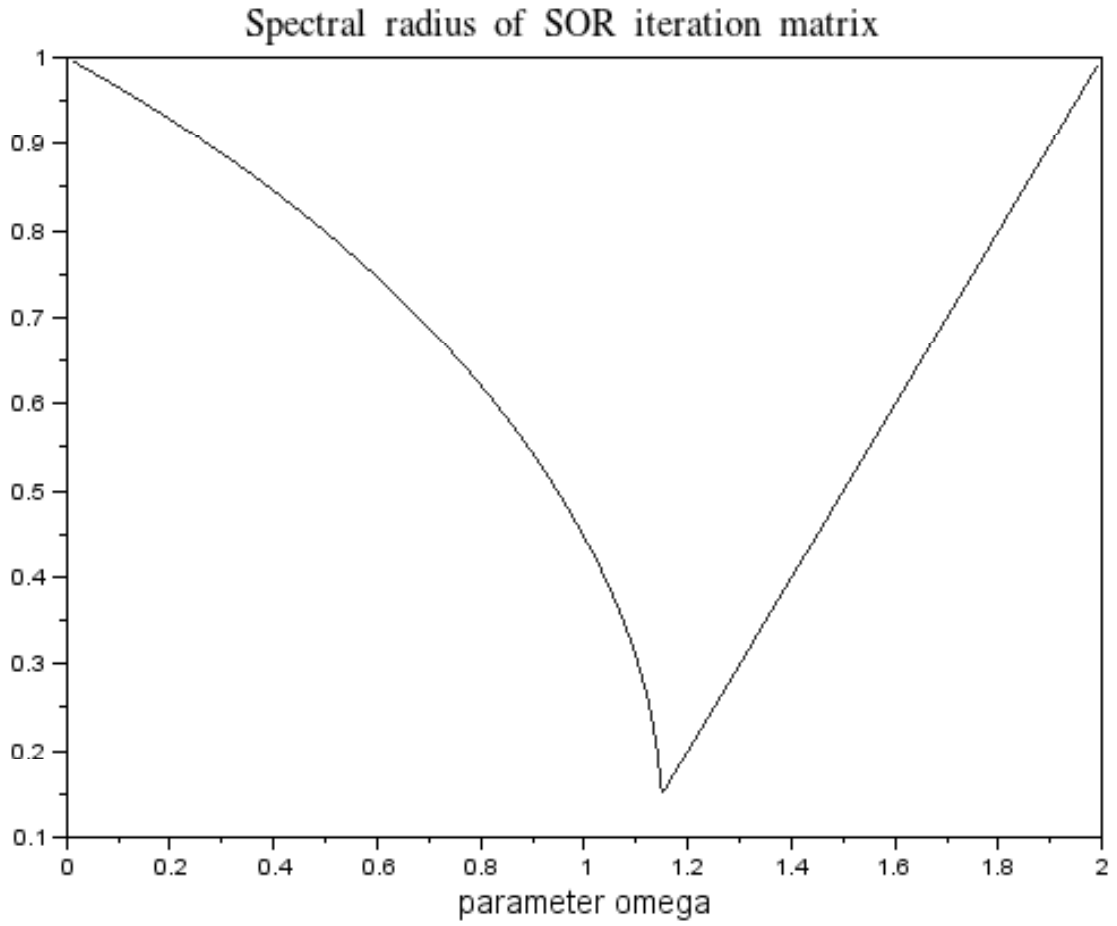
```
rho:25/(4*sqrt(14)*sqrt(31)+87) $
float(rho);
.1467733350400546

ratsimp(rho-((sqrt(56)-sqrt(31))/(sqrt(56)+sqrt(31))));
0
```

(firstly numerically, then equality of two exact numbers)

Remarks on **wxMaxima** commands:
$ at the end: the result of the computation is not printed
`float`: numerical value
`ratsimp`: simplifies the expression
`eigenvalues`: returns eigenvalues and their muliplicity, e.g., `[[1/3,1/2],[1,2]]`
means eigenvalue 1/3 with multiplicity 1 and eigenvalue with multiplicity 2

Theorem 3 and our example

- Comparison of the optimal value $\omega \approx 1.147$ which we obtained and the corresponding value of $\rho(\mathbf{T}_\omega) \approx 0.147$ with the application of the forumla from p. 17:



Exercise

- Repeat for another tridiagonal positive definite matrix:
  - computation of the iterations
  - spectral radius, the optimal choice of the parameter ω

SOR method and pricing derivatives

- EXERCISES SESSION:
  SOR method for solving the system of linear equations arising in the implicit scheme
  - practical implementation of a given ω
  - choice of ω - numerical computation of the Jacobi matrix spectrum, formulation of the theorem about its spectrum and spectral radius
  - optimal ω for different mesh grids - observations
- HOMEWORK 1:
  Show that this matrix is positive definite.
  (Hence the convergence theorems, which we have just seen, apply to our problem.)

Direct methods for solving systems of lin. eq.

- Direct methods for systems of linear equations - Gaussian elimination methos and its modifications, LU decomposition
- Tridiagonal, diagonally dominant matrix $\Rightarrow$ a convenient way of solving the system is the LU decomposition
- HOMEWORK 2 - LU decomposition method
  - What is the LU decomposition? How it can be used to solve a system of linear equations?
  - Show that the uniqueness of the decomposition follow from the matrix being diagonally dominant.
  - Why is this convenient, when solving a system with a tridiagonal matrix? (you do not need to memorize the formulae, but you need to show the principle of the computation on an example and explain why we do not obtain zeros in denumertors of fractions.)

References for HW2: e.g., [Ševčovič, Stehlíková, Mikula]

XII: Numerical methods for pricing American options - PSOR algorithm

Beáta Stehlíková

Financial derivatives

Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava

Numerical solution

- Recall:

$$\left(\frac{\partial u}{\partial \tau} - \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2}\right)(u(x,\tau) - g(x,\tau)) = 0,$$

$$\frac{\partial u}{\partial \tau} - \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} \geq 0, \quad u(x,\tau) - g(x,\tau) \geq 0$$

- Discretization - in the same way as in the implicit scheme for European options:

$$\mathbf{A}u^{j+1} \geq u^j + b^j, \quad u^{j+1} \geq g^{j+1} \quad \text{for } j = 0, 1, ..., m-1,$$
$$(\mathbf{A}u^{j+1} - u^j - b^j)_i (u^{j+1} - g^{j+1})_i = 0 \quad \forall\, i$$

Numerical solution

- Matrix A and vector b remain the same:

$$\mathbf{A} = \begin{pmatrix} 1+2\gamma & -\gamma & 0 & \cdots & 0 \\ -\gamma & 1+2\gamma & -\gamma & & \vdots \\ 0 & \cdot & \cdot & \cdot & 0 \\ \vdots & & -\gamma & 1+2\gamma & -\gamma \\ 0 & \cdots & 0 & -\gamma & 1+2\gamma \end{pmatrix},$$

$$b^j = (\gamma\phi^{j+1}, 0, \ldots, 0, \gamma\psi^{j+1})^T$$

where $\gamma = \frac{\sigma^2 k}{2h^2}$

PSOR method

- On each time level we solve a problem of the form

$$\begin{aligned} \mathbf{A}u \geq b, \quad u &\geq g, \\ (\mathbf{A}u - b)_i(u_i - g_i) &= 0 \quad \forall\, i. \end{aligned}$$

- Define the sequence

$$u^0 = 0, \quad u^{p+1} = \max\left(\mathbf{T}_\omega u^p + c_\omega,\ g\right) \quad \text{for } p = 1, 2, ...,$$

  where T_ω, c_ω come from the classical SOR method and maaximum is taken componentwise

- Projected SOR $\rightarrow$ known as PSOR method or PSOR algorithm

PSOR method

- Components of the approximate solution:

$$u_i^{p+1} = \max\left[\frac{\omega}{A_{ii}}\left(b_i - \sum_{j<i} A_{ij}u_j^{p+1} - \sum_{j>i} A_{ij}u_j^p\right) + (1-\omega)u_i^p,\ g_i\right]$$

Convergence of the algorithm to the solution

- The sequence u^p converges to some limit u - proof uses Banach fixed point theorem [Ševčovič, Stehlíková, Mikula: **Analytical and numerical methods for pricing financial derivatives**, pp. 156-157]
- This limit is a solution:
  - $u_i^{p+1} \geq g_i \ \Rightarrow$ also the limit satisfies $u_i \geq g_i$
  - $u_i^{p+1} \geq \frac{\omega}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j^{p+1} - \sum_{j>i} A_{ij} u_j^p \right)$ $+(1-\omega) u_i^p \ \Rightarrow$ also the limit satisfies $u_i \geq \frac{\omega}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j - \sum_{j>i} A_{ij} u_j^p \right) + (1-\omega) u_i$ we use that $A_{ii} > 0, \omega > 0 \ \rightarrow$ we obtain $(Au)_i \geq b_i$
  - if $u_i > g_i$, then starting with some index p_0 we have $u_i^p > g_i$; for these indices:

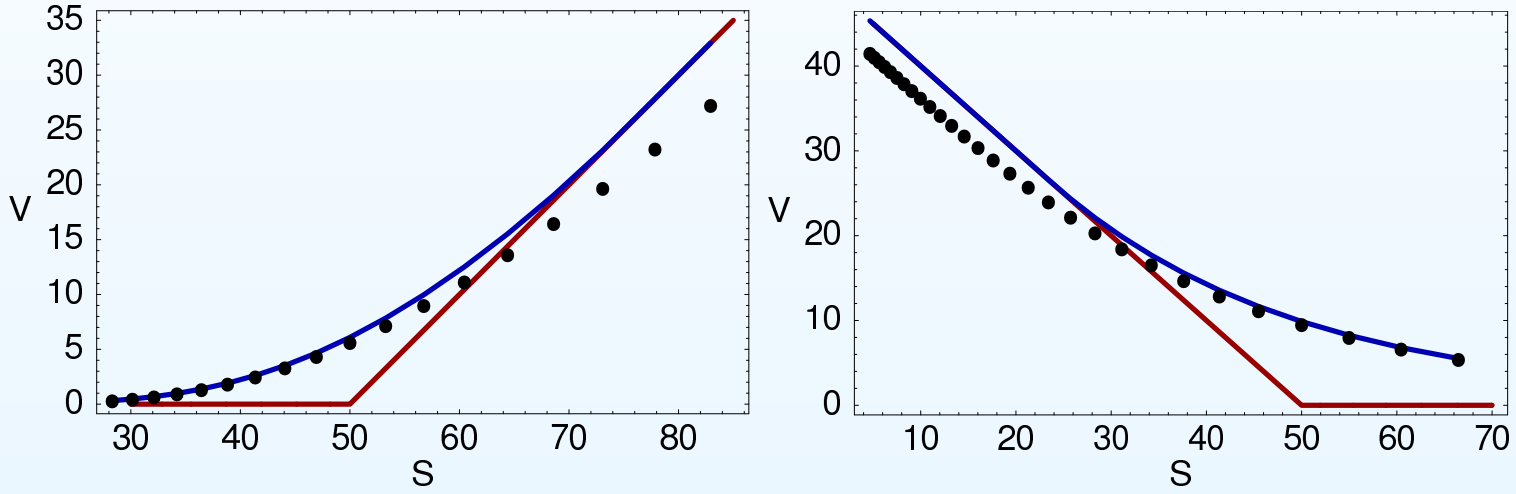
Convergence of the algorithm to the solution

$$u_i^{p+1} = \frac{\omega}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j^{p+1} - \sum_{j>i} A_{ij} u_j^p \right) + (1-\omega) u_i^p,$$

taking limit as $p \to \infty$ we get $(Au)_i = b_i \Rightarrow$ condition $(Au - b)_i (u_i - g_i) = 0$ is satisfied

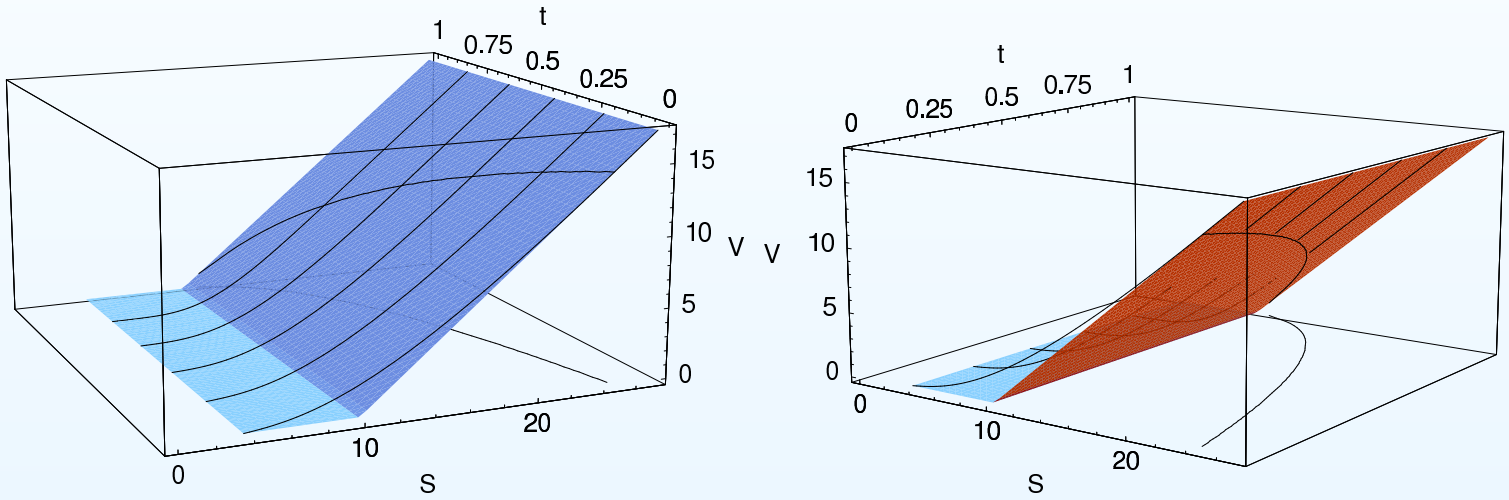
Numerical examples

- Pricing American call and put options (for a comparison: price of a European option - dotted line)



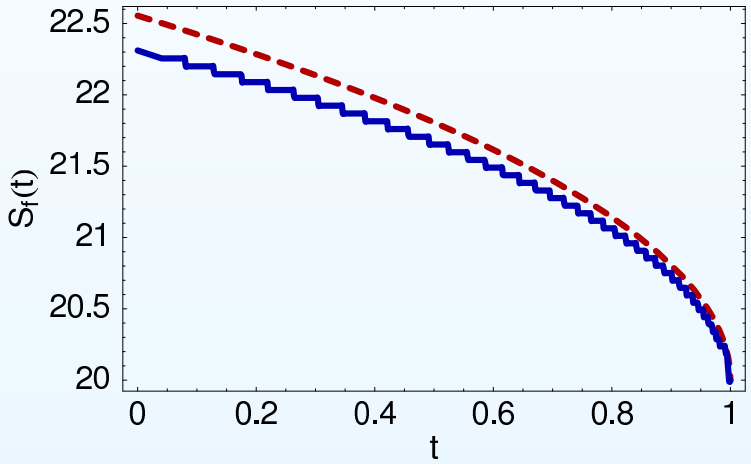
Numerical examples

- 3D graph for a call option, the free is depicted:



Numerical examples

- Numerical computation of the free boundary and its comparison with the "square root approximation formula"



Numerical examples

- M. Lauko, D. Ševčovič: **Comparison of numerical and analytical approximations of the early exercise boundary of American put options**, ANZIAM journal 51, 2010, 430-448.

  Comparison of approximation formulae for put options: