Solving the Black Scholes PDE with Holger's reproducing kernel. In this example, MQ kernel is used. The two-dimensional PDE is approximated with collecting of order-one-functions.

For d dimensions, $\vec{x} \in [0\ 1]^d$.

Define the desired evaluation points $X_E = \{\vec{x_1}, \vec{x_2}, \vec{x_3}, ..., \vec{x_{N_e}}\}$ and the set of anchored sample points $X_N = \{(x_1, a_y), (x_2, a_y), ..., (x_n, a_y), (a_x, a_y), (a_x, y_1), ..., (a_x, y_n)\}$ with $0 < x_i, y_i, < 1$. PDE, equation 1.

$$
\begin{aligned}
&\frac{\partial u}{\partial t} + \frac{1}{2}\sum_{i,j}^{d} \mathbf{C}_{i,j} s_i s_j \frac{\partial^2 u}{\partial s_i \partial s_j} + \sum_{i}^{d} r s_i \frac{\partial u}{\partial s_i} - ru = 0 && \vec{s} \in [0,1]^d, t \in [0,T] \\
&u(\vec{s},t) = \phi(\vec{s},t) = \max(\frac{1}{d}\sum^{d} s_i - Ke^{T-t}, 0) && ||\vec{s}|| \geqslant 1, t \in [0,T] \\
&u(\vec{s},t) = 0 && ||\vec{s}|| = 0, t \in [0,T] \\
&u(\vec{s},t) = \phi(\vec{s},T) && t = T
\end{aligned} \tag{1}
$$

We have different sets of values.

- Evaluation points $X_{eval} = \{(x_i, y_i)\}_i^{N_e}$
- Anchored center points $X = \{(x_i, y_i)\}_i^{N}$
- Inner points $X_{inner} \subseteq X, \{(x_i, y_i)\}_i^{N_inner}$
- Far and close points, where boundary condition apply $X_{far}, X_{close} \subseteq X, \{(x_i, y_i)\}_i^{N_{far}}, \{(x_i, y_i)\}_i^{N_{close}}$

We get that $N = N_{inner} + N_{far} + N_{close}$

Reproducing kernel

The reproducing kernel is for the 2D case defines as follows.

$$
\begin{aligned}
R(\vec{x}^1, \vec{x}^2) &= 1 + k(x_1^1, x_1^2) + k(x_2^1, x_2^2) \\
k(x_1, x_2) &= \sqrt{1 + \varepsilon^2 ||x_1 - x_2||_2^2}
\end{aligned} \tag{2}
$$

From equation 2 we can define the following matrix.

$$
\mathbf{A_0} = \left[R(\vec{x}^j, \vec{x}^k)\right]_{\vec{x}_j, \vec{x}_k \in X} \tag{3}
$$

Followingly, the derived reproducing kernels with associated matrix may be built.

$$
\begin{aligned}
\partial_{x_i} R(\vec{x}^j, \vec{x}^k) &= \frac{\varepsilon^2 (x_i^j - x_i^k)}{\sqrt{1 + \varepsilon^2 ||x_i^j - x_i^k||^2}} \Rightarrow \mathbf{A_1}, \mathbf{A_2} \\
\partial^2_{x_i x_i} R(\vec{x}^j, \vec{x}^k) &= \frac{\varepsilon^2}{(1 + \varepsilon^2 ||x_i^j - x_i^k||^2)^{3/2}} \Rightarrow \mathbf{A_{11}}, \mathbf{A_{22}} \\
\partial^2_{x_i x_j} R(\vec{x}^j, \vec{x}^k) &= 0 \Rightarrow \mathbf{A_{12}}
\end{aligned} \tag{4}
$$

The projected approximation of a function $f(\vec{x})$ at a arbitrary set of evaluation points (may be the same as center points).

$$
\underbrace{\hat{f}(X_{eval})}_{R^{Ne \times 1}} = \underbrace{\left[R(\vec{x}_i, \vec{x}_j)\right]_{\vec{x}_i \in X_{eval}, \vec{x}_j \in X}}_{R^{Ne \times N}} \cdot \underbrace{\mathbf{A_0}^{-1}}_{R^{N \times N}} \underbrace{f(X)}_{R^{N \times 1}} \tag{5}
$$

From equation 5, the way to approximate the 2D Black Scholes operator becomes.

$$
\begin{aligned}
\mathcal{O}_{BS} =&(rX_1^1\mathbf{B_1} + rX_2^2\mathbf{B_2}\\
&+ \frac{1}{2}\sigma_1^2(X_1^1)^2\mathbf{B_{11}} + \frac{1}{2}\sigma_2^2(X_2^2)^2\mathbf{B_{22}}\\
&+ \rho\sigma_1\sigma_2X_1^1X_2^2\mathbf{B_{12}}\\
&- r\mathbf{B_0})\mathbf{A_0}^{-1}
\end{aligned}
\tag{6}
$$

$$
\begin{cases}
\mathbf{B_0} = [\mathbf{A_0}]_{i\in N_i, j\in N}\\
\mathbf{B_1} = [\mathbf{A_1}]_{i\in N_i, j\in N}\\
\mathbf{B_2} = [\mathbf{A_2}]_{i\in N_i, j\in N}\\
\mathbf{B_{11}} = [\mathbf{A_{11}}]_{i\in N_i, j\in N}\\
\mathbf{B_{22}} = [\mathbf{A_{22}}]_{i\in N_i, j\in N}\\
\mathbf{B_{12}} = [\mathbf{A_{12}}]_{i\in N_i, j\in N}
\end{cases}
\tag{7}
$$

Coordinate Transformation

We want to rotate the coordinate system and solve the problem for v_1, v_2 instead of x_1, x_2. The transformation is linear and equation 9 & 10 is only applicable in the linear case.

$$
\vec{v} = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\vec{s}
\tag{8}
$$

$$
\frac{\partial^2 u}{\partial s_i \partial s_j} = \sum_{k,l=1}^{d}\left(\frac{\partial^2 u}{\partial v_k \partial v_l}\frac{\partial v_l}{\partial s_j}\frac{\partial v_k}{\partial s_i}\right)
\tag{9}
$$

$$
\frac{\partial u}{\partial s_i} = \sum_{k}^{d}\left(\frac{\partial u}{\partial v_k}\frac{\partial v_k}{\partial s_i}\right)
\tag{10}
$$

With this, we can rewrite equation 1 in this new system.

$$
\begin{aligned}
\mathcal{O}_{BS}^{\mathcal{T}} =&\Big(r(V_1^1 + V_2^2 - \frac{1}{2})\mathbf{B_1} + r(V_1^1 - V_2^2 + \frac{1}{2})\mathbf{B_2}\\
&+ \frac{1}{2}\sigma_1^2(V_1^1 + V_2^2 - \frac{1}{2})^2\mathbf{B_{11}} + \frac{1}{2}\sigma_2^2(V_1^1 - V_2^2 + \frac{1}{2})^2\mathbf{B_{22}}\\
&+ \rho\sigma_1\sigma_2(V_1^1 + V_2^2 - \frac{1}{2})(V_1^1 - V_2^2 + \frac{1}{2})\mathbf{B_{12}}\\
&- r\mathbf{B_0}\Big)\mathbf{A_0}^{-1}
\end{aligned}
\tag{11}
$$

$$
\begin{cases}
X_1^1 = (V_1^1 + V_2^2 - \frac{1}{2})\\
X_2^2 = (V_1^1 - V_2^2 + \frac{1}{2})\\
\mathbf{B_0} = [\mathbf{A_0}]_{i\in N_i, j\in N}\\
\mathbf{B_1} = [\frac{1}{2}(\mathbf{A_1} + \mathbf{A_2})]_{i\in N_i, j\in N}\\
\mathbf{B_1} = [\frac{1}{2}(\mathbf{A_1} - \mathbf{A_2})]_{i\in N_i, j\in N}\\
\mathbf{B_{11}} = [\frac{1}{4}(\mathbf{A_{11}} + 2\mathbf{A_{12}} + \mathbf{A_{22}})]_{i\in N_i, j\in N}\\
\mathbf{B_{22}} = [\frac{1}{4}(\mathbf{A_{11}} - 2\mathbf{A_{12}} + \mathbf{A_{22}})]_{i\in N_i, j\in N}\\
\mathbf{B_{12}} = [\frac{1}{2}(\mathbf{A_{11}} - \mathbf{A_{22}})]_{i\in N_i, j\in N}
\end{cases}
\tag{12}
$$

Time Solver (BDF2)

From earlier equations, we can define the system that should be solved at each timestep. Note that the set each point belongs to (Inner, Far) is still defined in terms of x_1, x_2. Reversing time $\tau = T - t$ gives:

$$\begin{aligned}
&U = \begin{bmatrix} u(\vec{x_i}, t) \\ \vdots \end{bmatrix}_{\vec{x_i} \in X} \\
&\frac{\partial U}{\partial \tau} = \mathcal{O}_{BS}^{\mathcal{T}} U && x_i \in X_{in} \\
&U = \phi(\vec{x_i}, \tau) && x_i \in X_{far} \\
&U = 0 && x_i \in X_{close}
\end{aligned} \tag{13}$$

Discretize the time $\vec{\tau} = \{\tau_m\}_{m=1}^{M}$ with time step $k_m = \tau_{m+1} - \tau_m$. The PDE is then expanded as:

$$U^{m+2} - \frac{4}{3}U^{m+1} + \frac{1}{3}U^m = \Delta\tau_m \mathcal{O}_{BS}^{\mathcal{T}} U^{m+2} \qquad x_i \in X_{in} \tag{14}$$

$$\Leftrightarrow (I - \Delta\tau_m \mathcal{O}_{BS}^{\mathcal{T}})U^{m+2} = \frac{4}{3}U^{m+1} - \frac{1}{3}U^m \qquad x_i \in X_{in} \tag{15}$$

$$\Leftrightarrow U^{m+2} = (I - \Delta\tau_m \mathcal{O}_{BS}^{\mathcal{T}})^{-1}(\frac{4}{3}U^{m+1} - \frac{1}{3}U^m) \qquad x_i \in X_{in} \tag{16}$$

$$\tag{17}$$

To eliminate the boundary conditions, the matrices are expanded to size $N \times N$. This new matrix, C, has identical rows to $(I - \Delta\tau_m \mathcal{O}_{BS}^{\mathcal{T}})$ for the indices corresponding with the interior points while being the identity matrix for the boundary rows. The boundary condition is then enforced by applying the respective condition to the boundary indices of the right-hand side. $U_{RHS} = (\frac{4}{3}U^{m+1} - \frac{1}{3}U^m)$. This is done before solving the linear system. The final system is then given by.

$$\begin{aligned}
&U^{m+2} = C^{-1}U_{RHS} \qquad \vec{x_i} \in X \\
&\begin{cases} U_{RHS} = \frac{4}{3}U^{m+1} - \frac{1}{3}U^m, & \vec{x_i} \in X_{in} \\ U_{RHS} = 0, & \vec{x_i} \in X_{close} \\ U_{RHS} = \phi(\vec{x_i}, \tau_{m+1}), & \vec{x_i} \in X_{far} \end{cases}
\end{aligned} \tag{18}$$

Algorithm

Algorithm 1 Pseudo code for solving BS

Scale X_{eval}, K to and define problem on $[0,1]^d]$
Generate center points in transformed coordinates
Obtain center points in standard system
Find points corresponding to close and far boundary.
Define **A** matrices, equation (3) - (4).
With Matrices A, build the rotated local matrices B. Equation 12.
Build the transform Black-Scholes operator $\mathcal{O}_{BS}$. Equation (11).
Extended $\mathcal{O}_{BS}$ to form matrix C, eliminating BC. $U_{m+1} = U_m = u_0$
Apply initial condition
for τ_m; m++ **do**
    $U_{temp} = C^{-1} U_{RHS}$
    $U_{RHS} = \frac{4}{3} * U_{m+1} - \frac{1}{3} * U_m$
    Apply boundary conditions:
        $U_{RHS}(\vec{x} \in X_{far}, \tau_{m+1}) = \phi(\vec{x}, \tau_m)$
        $U_{RHS}(\vec{x} \in X_{close}, \tau_{m+1}) = 0$
    Move solution along:
        $U_m = U_{m+1}$
        $U_{m+1} = U_{m+2}$
        $U_{m+2} = U_{temp}$
end for
Build evaluation matrix and evaluate solution in X_{eval}. Equation (5)
Rescale problem

Generalization

Repruducing Kernal

1 Resultat

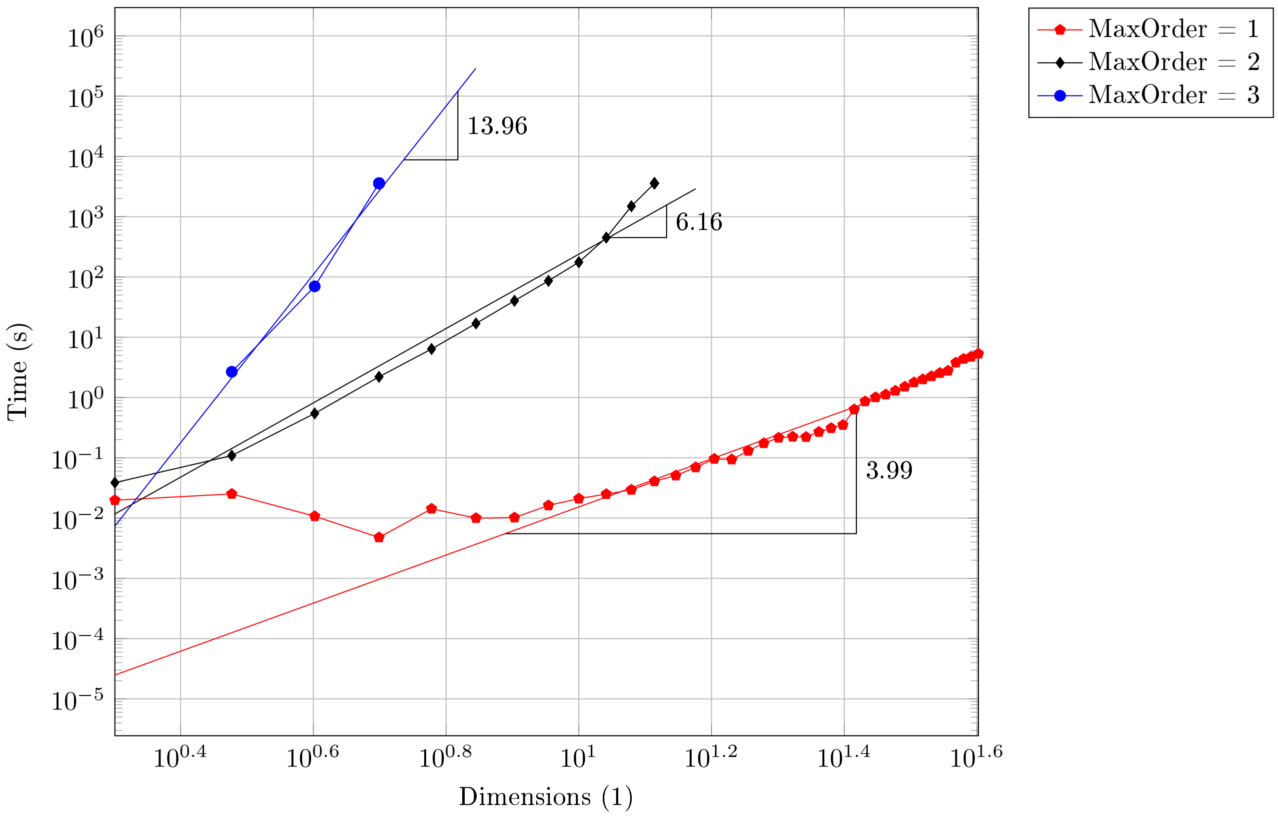
Tabell 1: Comparison between solutions generated with different maxOrder. $U_{d,maxOrder}$

$\mathbf{d}$	$\mathbf{\Omega(X_{eval})}$	$\mathbf{N_e}$	$\|\|\mathbf{U_{d,1} - U_{d,2}}\|\|_\infty$	$\|\|\mathbf{U_{d,2} - U_{d,3}}\|\|_\infty$	$\|\|\mathbf{U_{d,1} - U_{d,3}}\|\|_\infty$	$\|\|\mathbf{U_{d,1}}\|\|_\infty$
2	$[\frac{1}{3}K, \frac{5}{3}K]^d$	121	0.0919	N/A	N/A	13.4848
3	$[\frac{1}{3}K, \frac{5}{3}K]^d$	1331	0.0905	0.0727	0.1449	13.4103
4	$[\frac{1}{3}K, \frac{5}{3}K]^d$	625	0.0660	0.0622	0.1210	13.2346
4	$[0, 2K]^d$	625	0.1906	0.1073	0.2581	20.1911
5	$[\frac{1}{3}K, \frac{5}{3}K]^d$	3125	0.0630	0.0532	0.1031	13.0483
6	$[\frac{1}{3}K, \frac{5}{3}K]^d$	15625	0.0598	TBD?	TBD?	13.4752

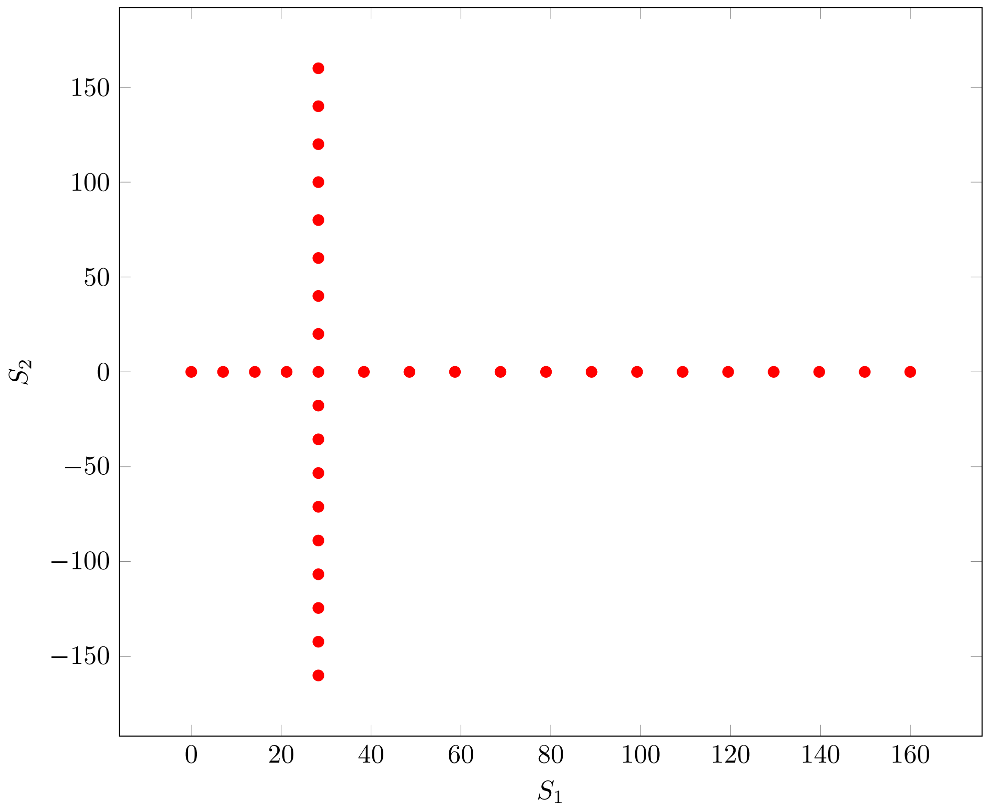
$$N = \sum_{i=0}^{maxOrder} \binom{d}{i} n^i \tag{19}$$

Tabell 2: Run times for different dimensional problems and max order

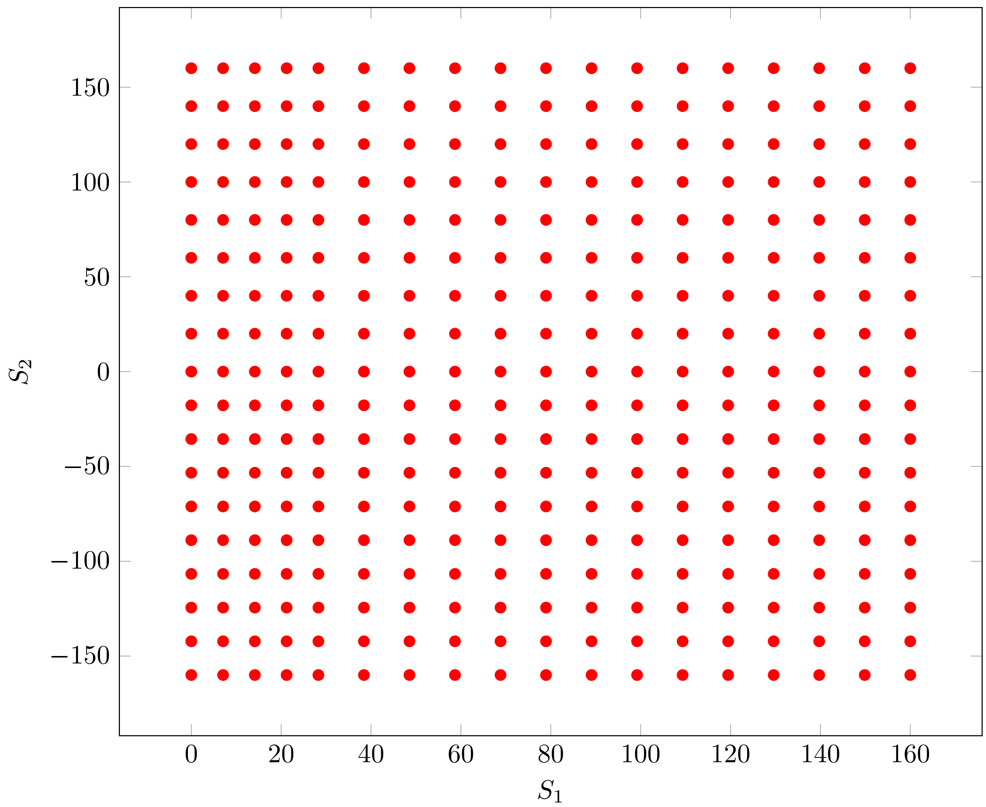
			Run Time [s]		
n	$\mathbf{N_e}$	**d**	MaxOrder = 1	MaxOrder = 2	MaxOrder = 3
10	1	2	0.0197794	0.038682	-
10	1	3	0.0251483	0.10886	2.6737
10	1	4	0.0107194	0.54458	69.8605
10	1	5	0.0047781	2.2029	3579.4775
10	1	6	0.0142638	6.3872	-
10	1	7	0.0099809	16.9169	-
10	1	8	0.0101846	40.2869	-
10	1	9	0.0161873	86.2708	-
10	1	10	0.0210623	175.614	-
10	1	11	0.0249659	421.7462	-
10	1	12	0.0293936	1491.1291	-
10	1	13	0.0406836	3580.1538	-
10	1	14	0.0507523	-	-
10	1	15	0.0693409	-	-
10	1	16	0.0961076	-	-
10	1	17	0.0936543	-	-
10	1	18	0.1303859	-	-
10	1	19	0.2148575	-	-
10	1	20	0.2233701	-	-
10	1	21	0.2683903	-	-
10	1	22	0.3102177	-	-
10	1	23	0.2683903	-	-
10	1	24	0.3102177	-	-
10	1	25	0.3495040	-	-



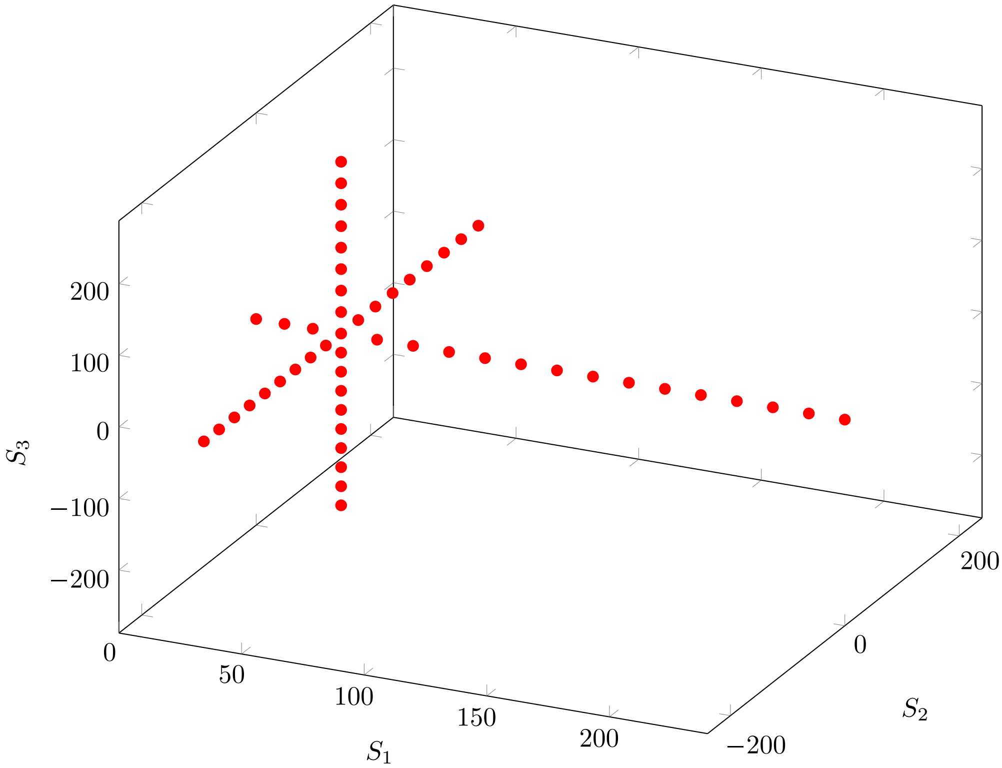
Figur 1: Loglog time with fitted linjes



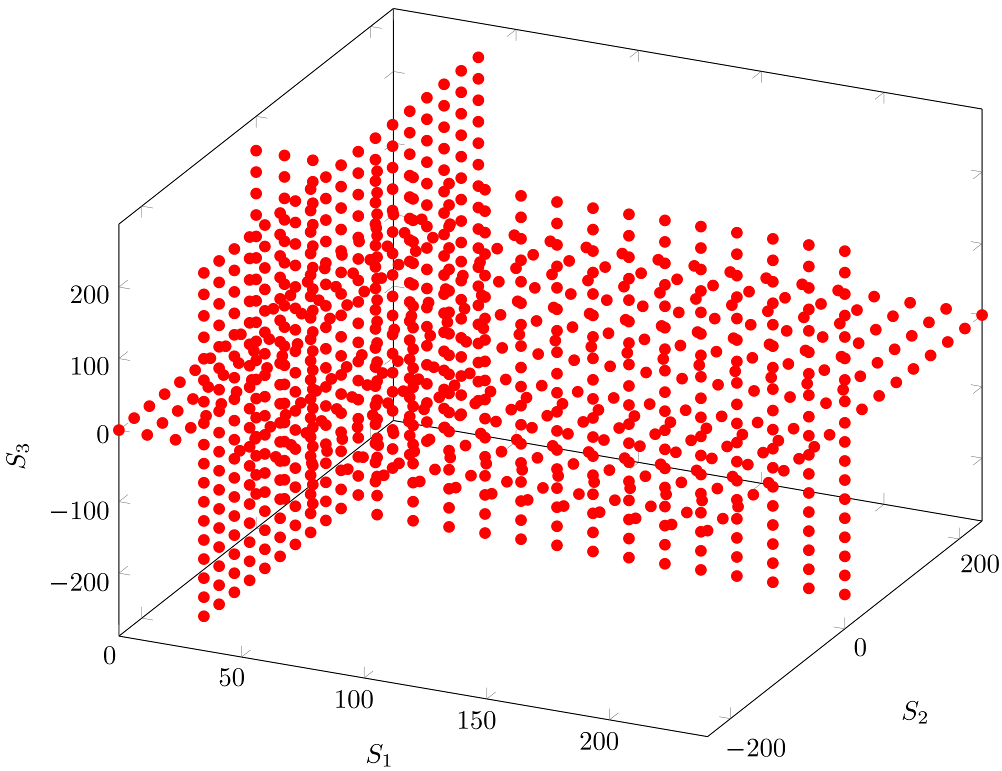
Figur 2: Dim2 max1



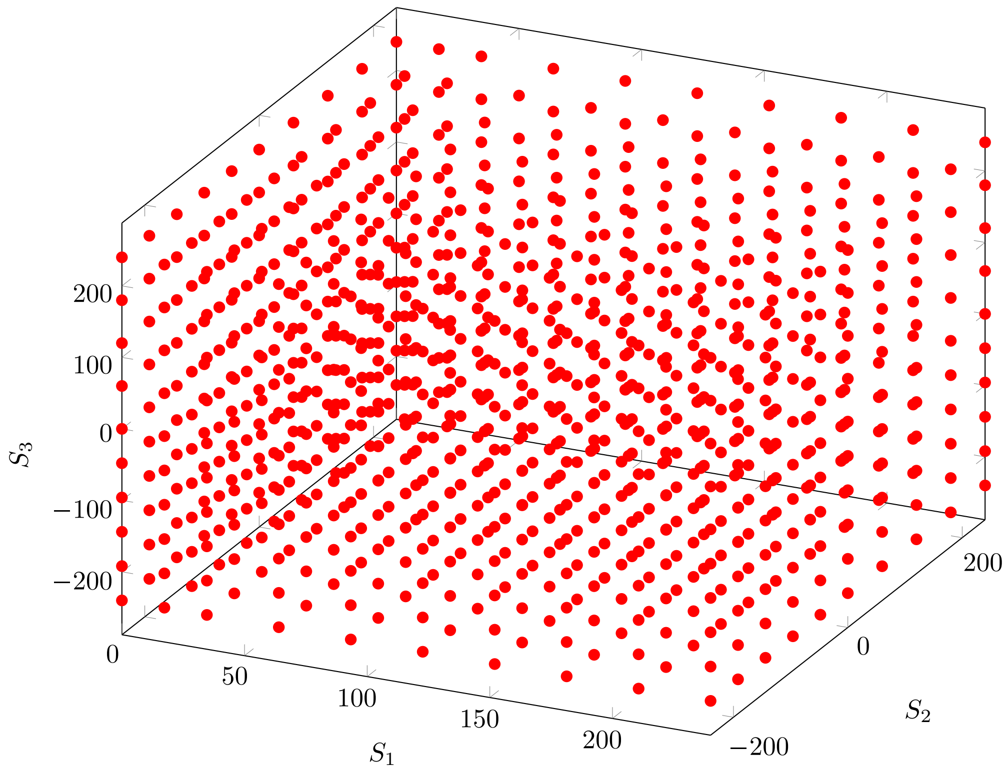
Figur 3: Dim2 max1



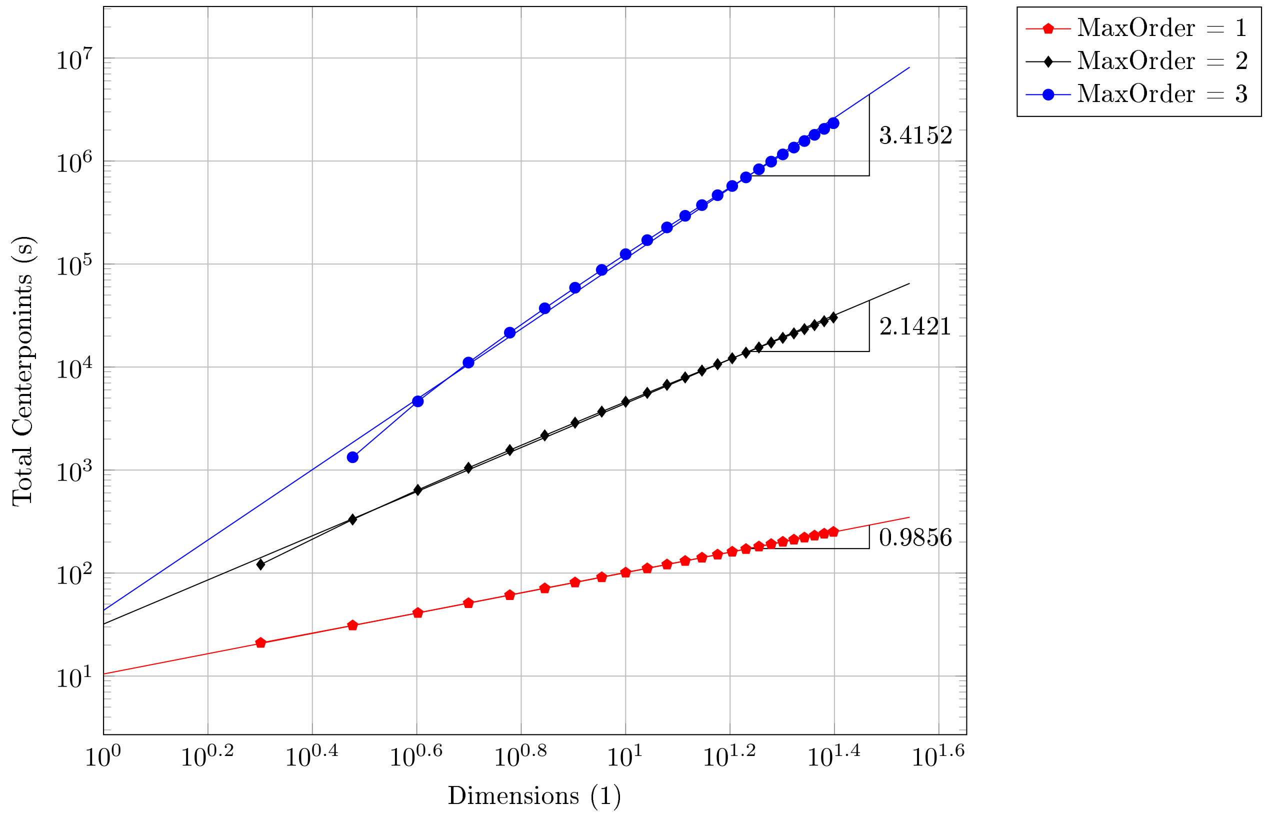
Figur 4: Dim2 max1



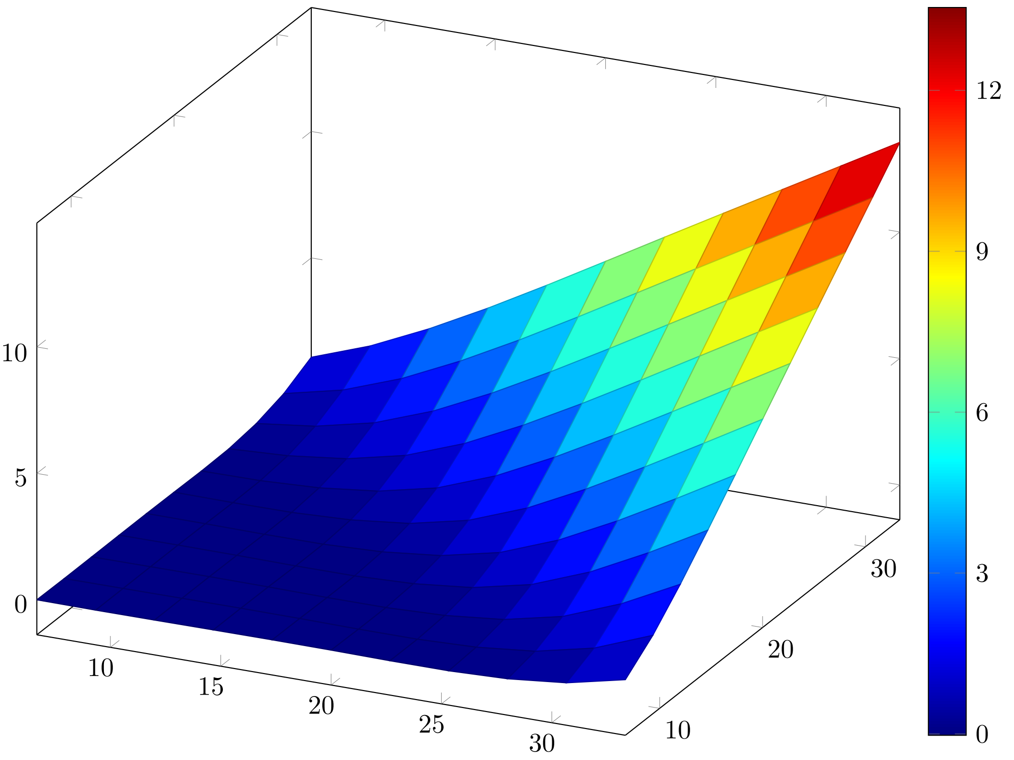
Figur 5: Dim3 max2



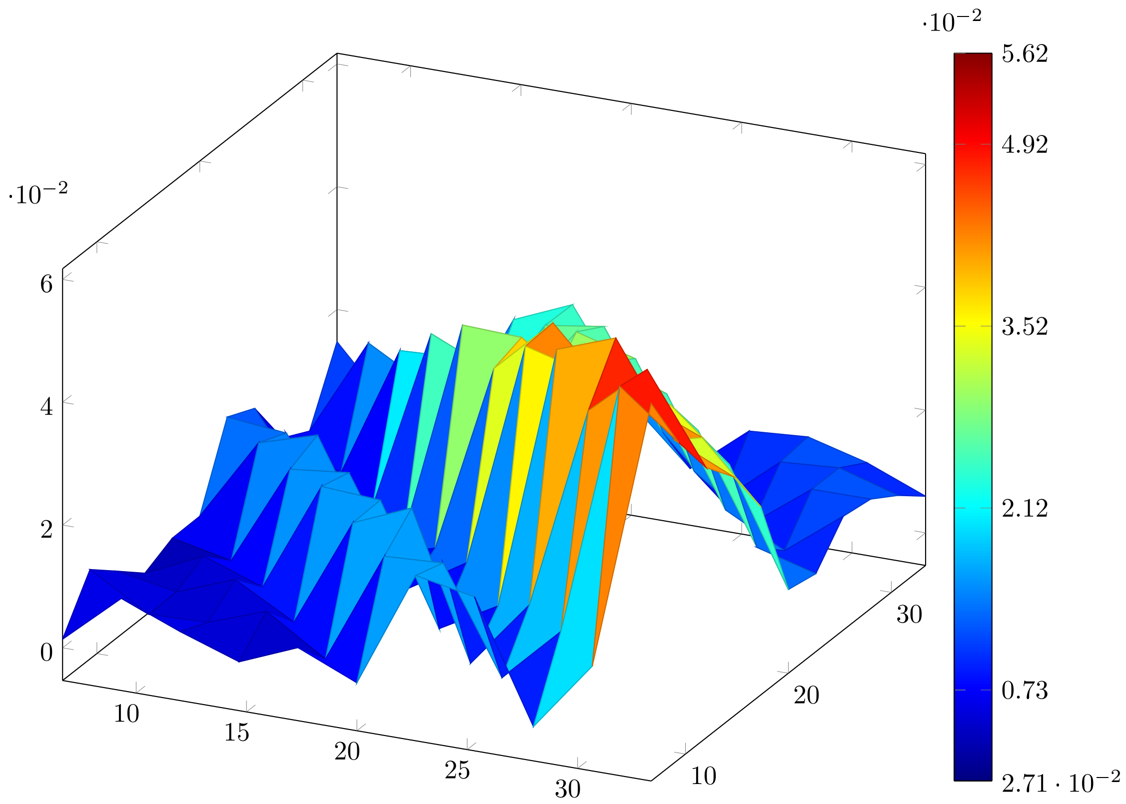
Figur 6: Dim3 max3



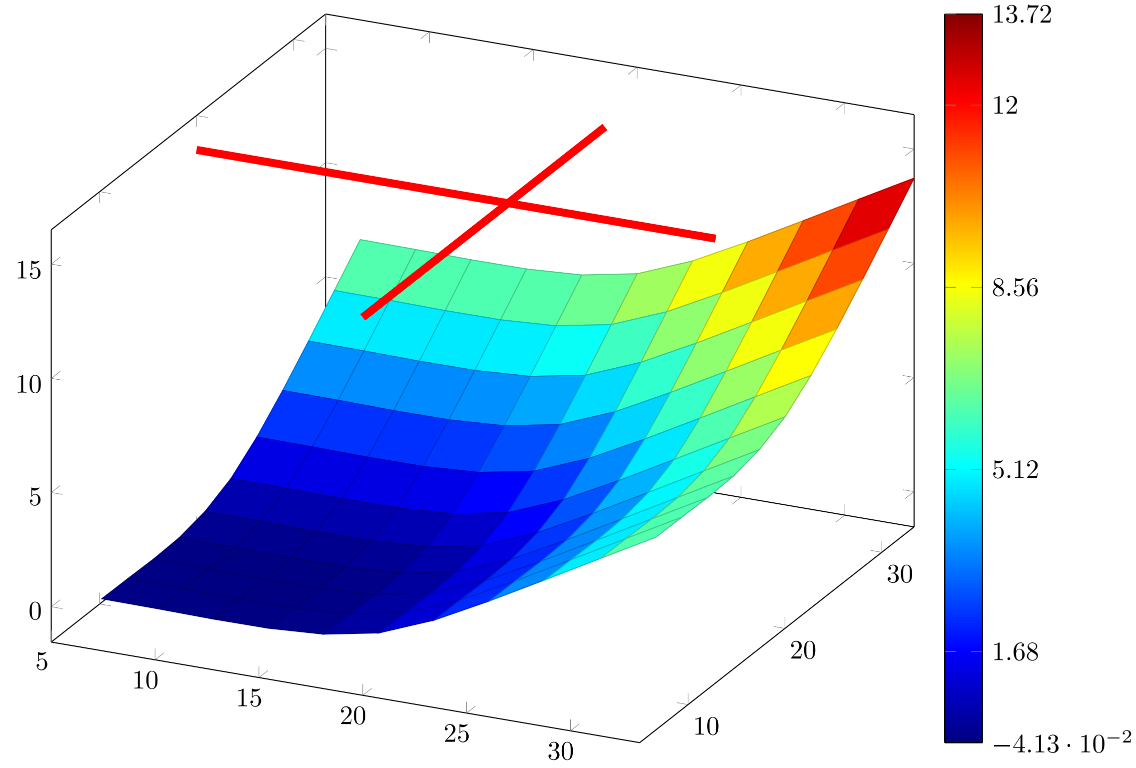
Figur 7: Total number of center points (N) for n = 10



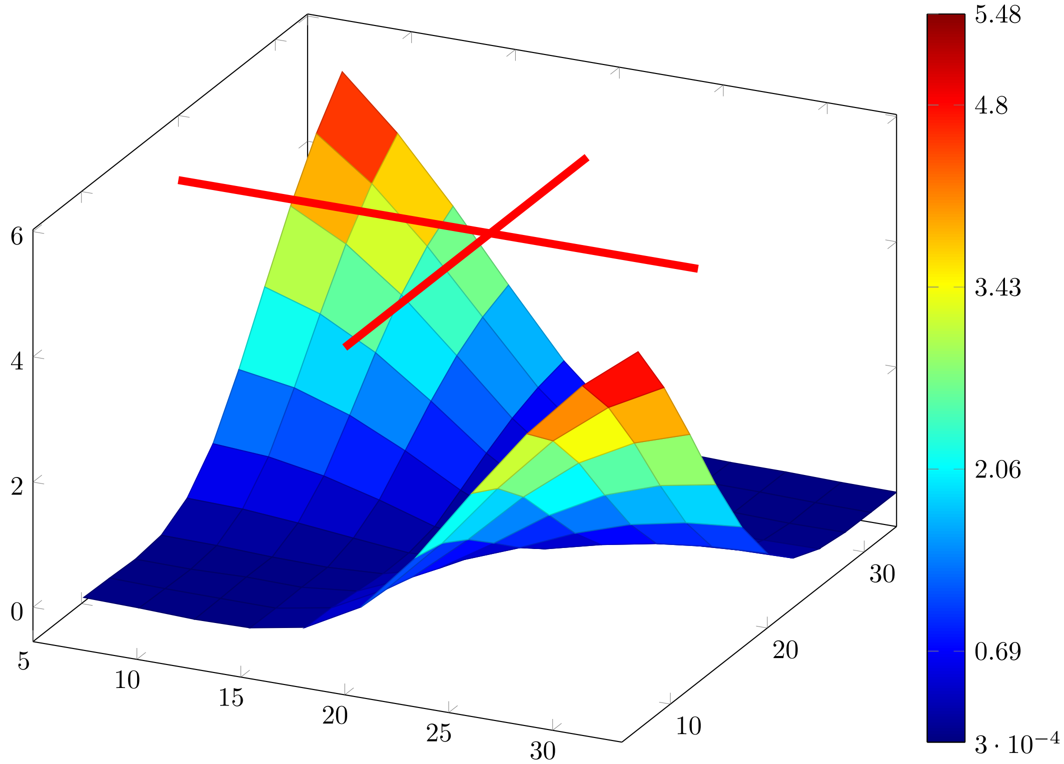
Figur 8: Computed solution around strike. Two dimensions and Max order 2



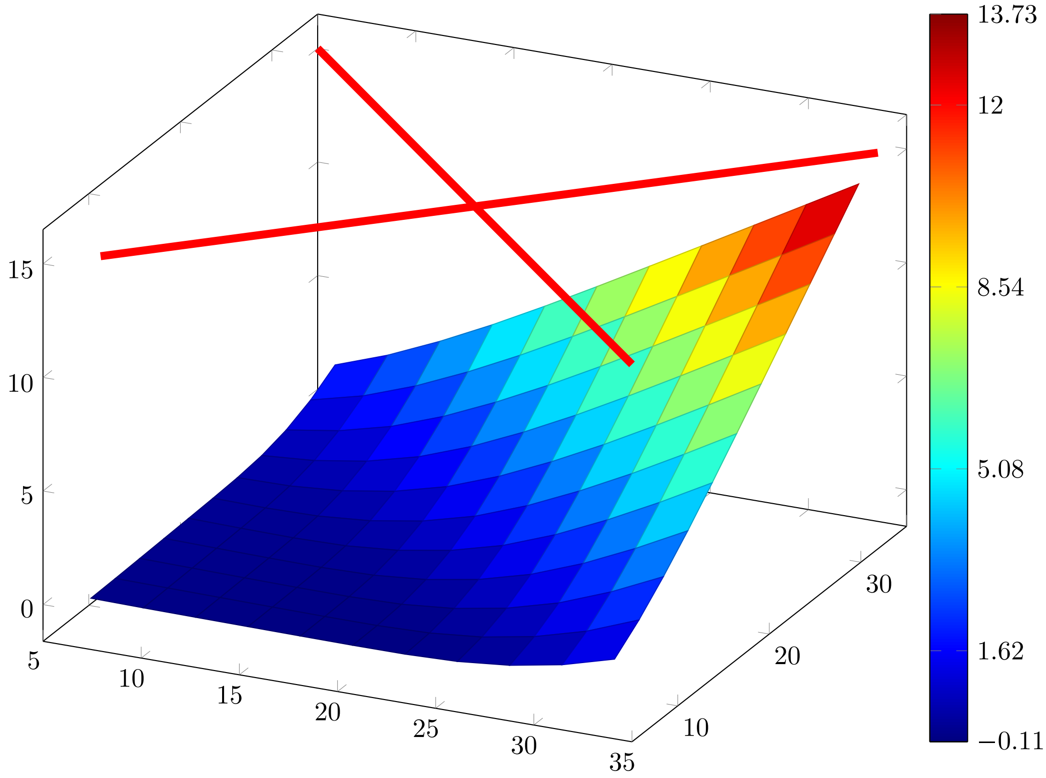
Figur 9: Error around strike from solution compared with established solver. Two dimensions and Max order 2



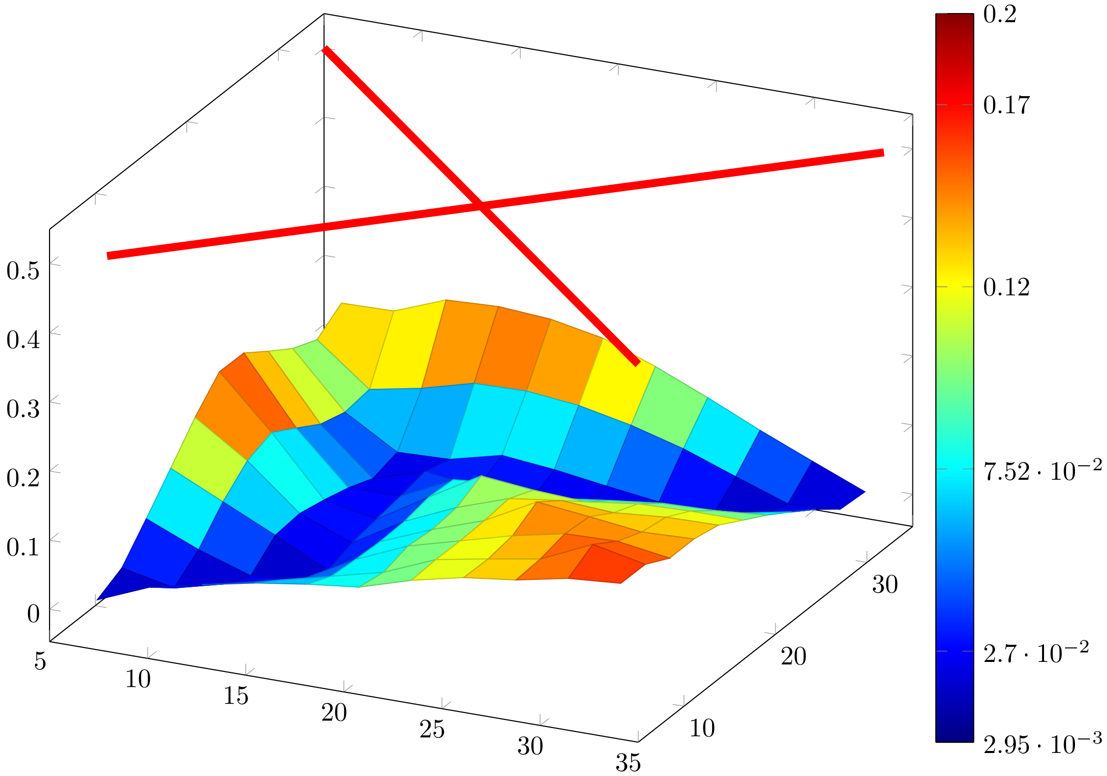
Figur 10: Solution around strike from solution. No rotation D2M1 T = 1



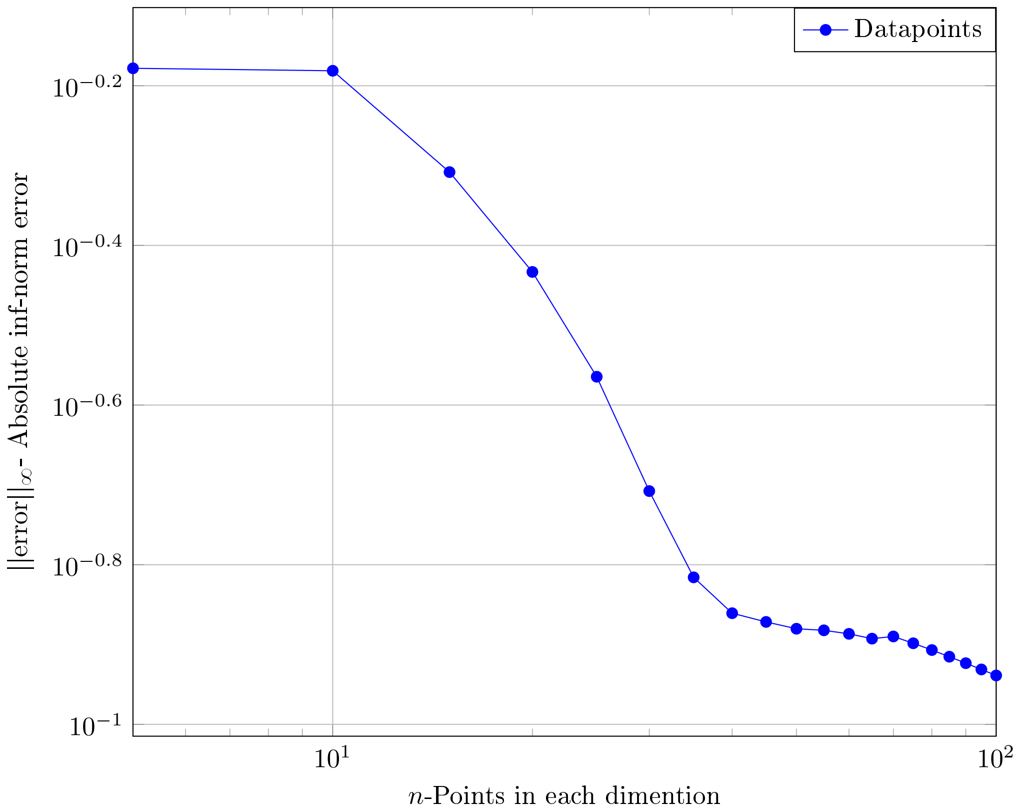
Figur 11: Error around strike from solution. No rotation D2M1 T = 1



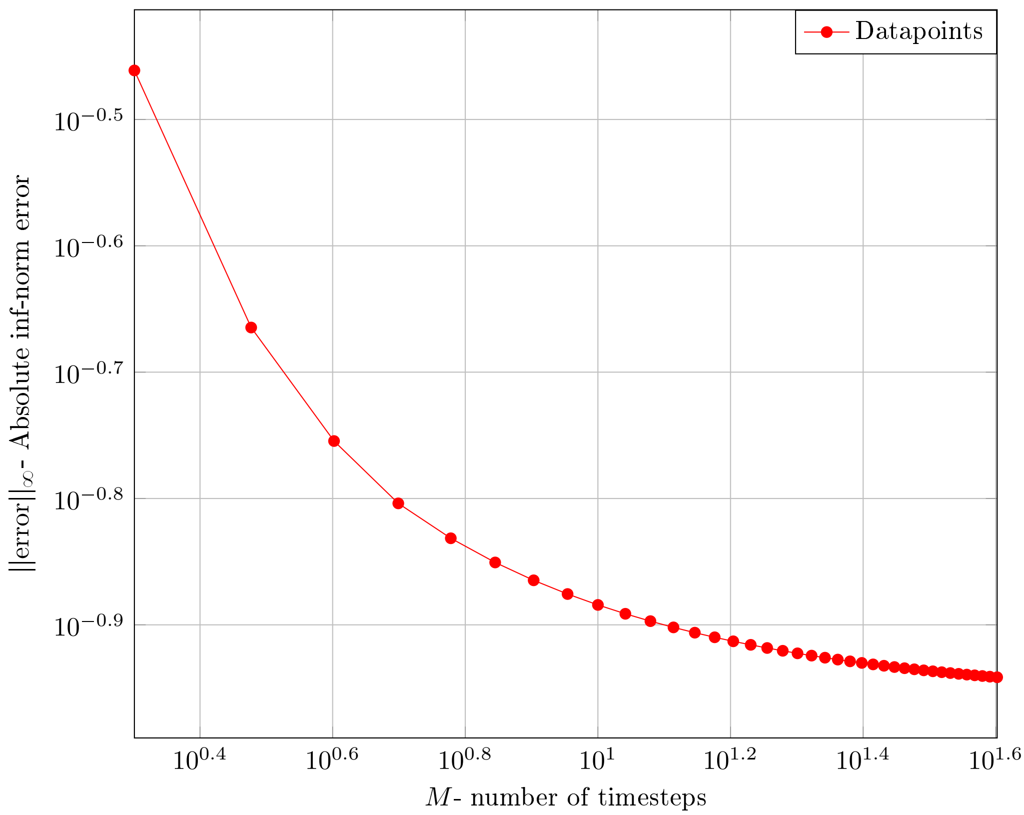
Figur 12: Solution around strike from solution. Rotation D2M1 T = 1



Figur 13: Solution around strike from solution. Rotation D2M1 T = 1



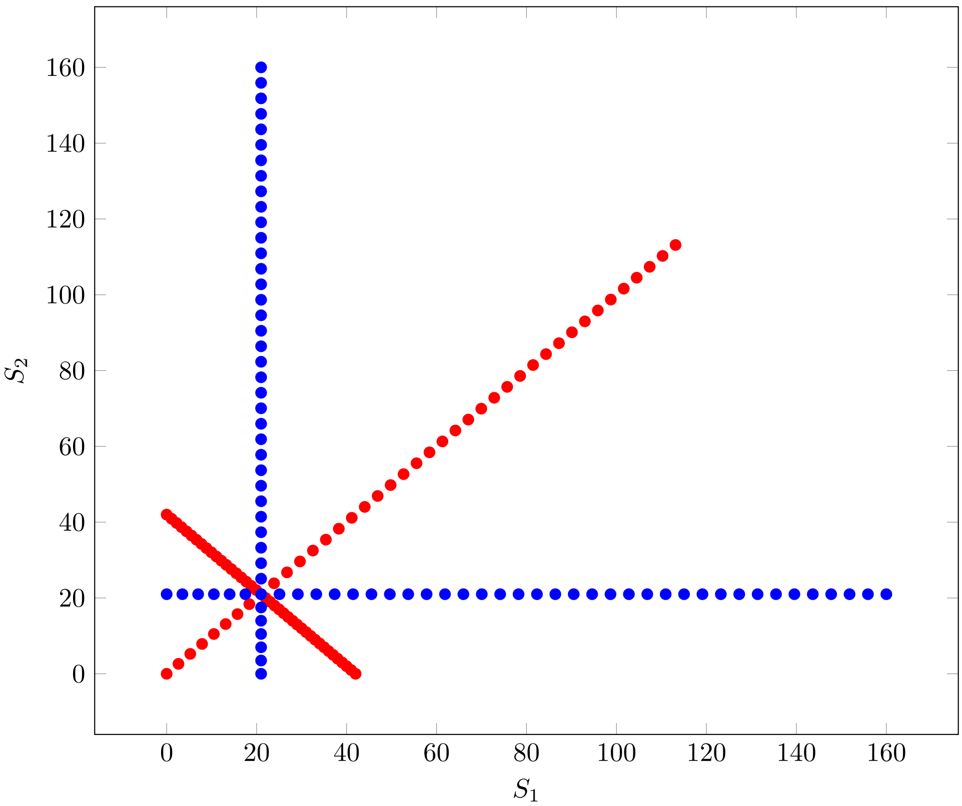
Time Convergence in loglog graph

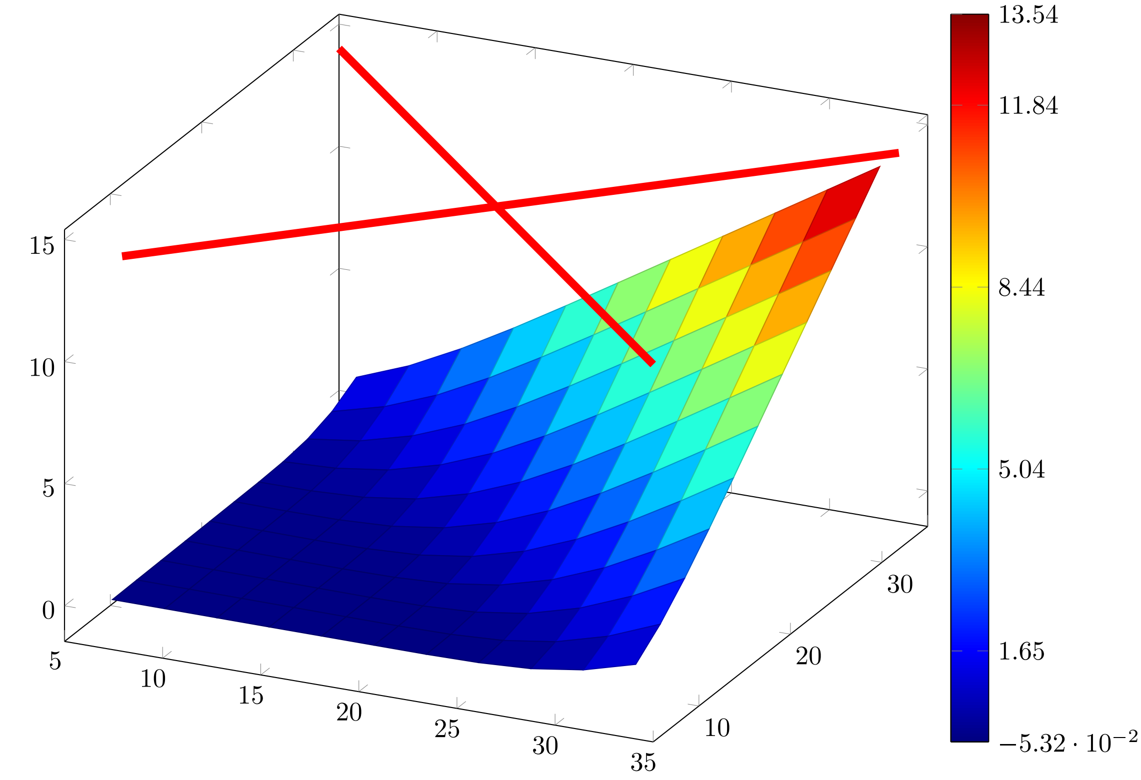


$$K = 20,\ T = 0.5,\ r = 0.02,$$

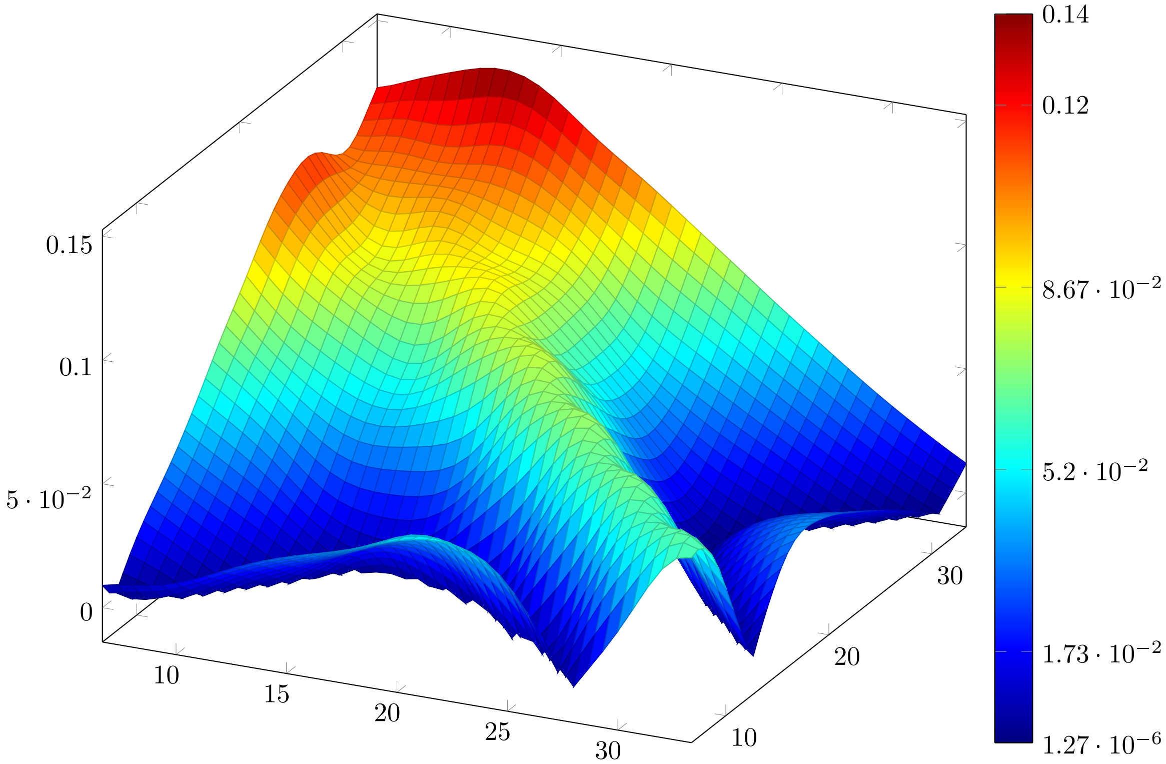
$$C = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\ \rho_{12}\sigma_2\sigma_1 & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} 0.0225 & 0.0150 \\ 0.0150 & 0.0400 \end{pmatrix}$$

$$n = 40,\ M = 15,\ \epsilon = 50$$
$$\text{anchor} = 21$$





Figur 14: Solution around strike from solution. Rotation D2M1 T = 1



Figur 15: Error final