The combination method for multidimensional Black-Scholes PDEs

Ray Wu and Christina C. Christara

University of Toronto
Department of Computer Science

June 13, 2023

Overview

- Motivation, method introduction, theoretical considerations.
- Black-Scholes equation, European options, American options.
- Solution methods, ADI methods, penalty method, etc.
- Smoothing techniques, continuity and smoothness of initial conditions.
- Convergence rates, computational savings of the sparse grid method.
- Numerical examples.
- Conclusions and future work.

Motivation: The Curse of Dimensionality

Why are we interested in solving high-dimensional PDEs?

- Many examples, but we are specifically interested in pricing multi-asset options accurately.
- Some options can have hundreds of underlying assets, each asset giving rise to a spatial dimension.
- Consideration of risk factors also give rise to additional spatial dimensions.

Curse of Dimensionality

The *curse of dimensionality* refers to the problem that the complexity of the numerical method scales exponentially with the dimension. With N gridpoints per dimension, there are N^d unknowns in total. Finite Difference Methods (FDMs) and Finite Element Methods (FEMs) on full grids suffer from this problem.

Dimensions	1	2	3	4
Execution Time	<0.01 sec	0.1 sec	15 sec	28 min
Accuracy	1.57×10^{-2}	3.15×10^{-3}	1.44×10^{-4}	9.24×10^{-4}
Unknowns	64	64^2	64^3	64^4

Table 1: Exponential increase in runtime of an ADI method as dimensions increase

Sparse Grids

Sparse grids, originally introduced by [Smolyak, 1963] are a method of mitigating the curse of dimensionality when discretizing a d-dimensional rectangular domain.

Instead of a discretization on the full grid, the sparse grid method chooses a subset of points on the full grid for spatial discretization.

- the standard sparse grid method alleviates the curse of dimensionality to some extent
- energy-based sparse grid method claims to overcome it entirely

Types of sparse grid PDE methods include

- Hierarchical Sparse Grid Finite Element Method [Balder and Zenger, 1996], concept originally introduced in [Yserentant, 1986].
- Sparse Grid Combination Method [Griebel et al., 1992]

The Combination Method

The combination method avoids the hierarchical discretizations, and instead combines solutions on smaller, anisotropic full grids still based on a tensor product formulation.

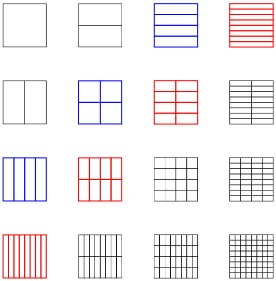

Figure 1: Tabular arrangement of full grids in two dimensions; highlighted grids correspond to the grids selected for the combination method. The number of unknowns on an d dimensional domain is reduced from $\mathcal{O}(n^d)$ to $\mathcal{O}(n(\log n)^{d-1})$.

Computing the solution with the combination method

Let $u_{i,j}$ ($u_{i,j,k}$) denote the numerical solution computed on the grid with level i in the x variable and level j in the y variable (and level k in the z variable).

Each increase in level doubles the number of gridpoints in that dimension.

In two dimensions, the combination method solution is computed by

$$u^c_{q,q} = \sum_{i+j=q+1} u_{i,j} - \sum_{i+j=q} u_{i,j}. \tag{1}$$

Using asymptotic error expansions, it can be shown that the error terms involving x and y on the grids that are largest in the respective dimensions cancel out.

In three dimensions, the combination method solution is computed by

$$u^c_{q,q,q} = \sum_{i+j+k=q+2} u_{i,j,k} - 2 \sum_{i+j+k=q+1} u_{i,j,k} + \sum_{i+j+k=q} u_{i,j,k}, \tag{2}$$

and the error analysis is similar.

Discussion between Hierarchical and Combination Methods

Both the hierarchical and the combination method have similar properties in terms of theoretical efficiency, however, in terms of implementation there are some differences, with the following advantages for the combination method:

- The combination method is based on existing standard PDE solvers (which can be based on a tensor product formulation), which allows us to solve the Black-Scholes equation on the original grid with untransformed variables.
  - Solving the Black-Scholes equation with the hierarchical finite element method typically requires us to transform it and remove derivatives of a certain order, which causes the numerical quality of the solution to decrease.
  - If this transformation is not applied, then the finite element matrix is dense, and would take a long time to construct and solve.
- The combination method does not involve any transformations between a nodal basis and hierarchical basis.
- All subproblems of the combination method can be solved in parallel.

On the other hand, the hierarchical sparse grid method has less smoothness requirements on the solution.

Black-Scholes equation

The Black-Scholes equation for pricing financial options is given by

$$V_\tau = \mathcal{L}V \equiv \frac{1}{2}\sigma^2 S^2 V_{SS} + (r-q)SV_S - rV. \tag{3}$$

Note that subscripts denote partial derivatives, and

- S denotes the stock price,
- τ denotes the reverse time counted from expiry T ($\tau = T - t$, t is forward time),
- σ denotes the volatility of the stock,
- r denotes the risk-free interest rate,
- q denotes the rate of dividend yield,
- V denotes the unknown option price we are solving for.

We are interested in the option values at $\tau = T$. Payoff functions denoted by $V^*(S)$ correspond to initial conditions:

- Call Payoff: $V(0,S) = V^*_{\text{call}}(S) = \max(S-K,0)$,
- Put Payoff: $V(0,S) = V^*_{\text{put}}(S) = \max(K-S,0)$.

Multidimensional Black-Scholes PDEs

In d dimensions, the Black-Scholes PDE is given by

$$V_\tau = \mathcal{L}V \equiv \frac{1}{2}\sum_{i,j=1}^{d} \rho_{i,j}\sigma_i\sigma_j S_i S_j V_{S_i,S_j} + \sum_{i=1}^{d}(r-q_i)S_i V_{S_i} - rV. \qquad (4)$$

where

- S_i, σ_i, q_i denote the price, volatility, and dividend yield of the i-th stock.
- $\rho_{i,j}$ denotes the correlation between stocks S_i and S_j.

Many different payoffs, but we use Geometric Average Put, given by

$$V(0, S_1, S_2, \dots, S_d) = \max(K - \left(\prod_{i=1}^{d} S_i\right)^{1/d}, 0) \qquad (5)$$

It is useful because it is equivalent to a corresponding one-dimensional problem, with r unchanged, and adjusted parameters $\hat{\sigma}$ and $\hat{q}$ given by

$$\hat{\sigma} = \frac{1}{d}\sqrt{\sum_{i=1}^{d} \rho_{i,j}\sigma_i\sigma_j}, \quad \hat{q} = \frac{1}{d}\sum_{i=1}^{d}(q_i + \frac{1}{2}\sigma_i^2) - \frac{1}{2}\hat{\sigma}^2, \qquad (6)$$

which has a known solution formula.

American options

Due to the early exercise right, the American option satisfies a linear complementarity problem (LCP)

$$\begin{gathered} V_\tau - \mathcal{L}V \geq 0 \text{ and } V = V^* \text{ or} \\ V_\tau = \mathcal{L}V \text{ and } V \geq V^* \end{gathered} \tag{7}$$

which is replaced by the nonlinear PDE

$$V_\tau = \mathcal{L}V + \rho \max(V^* - V, 0) \tag{8}$$

with ρ being the reciprocal of the desired accuracy ϵ for solving with a penalty iteration algorithm (i.e. [Forsyth and Vetzal, 2002]).

Numerical methods

We use the following numerical methods:

- Second order accurate methods for discretizing time and space derivatives.
  - centered finite differences on a nonuniform grid for space variables
  - Crank-Nicolson-Rannacher (CNR) or Modified Craig-Sneyd (MCS) ADI [Wyns et al., 2016] for timestepping.
- In the numerical solution of multi-dimensional PDEs, a lexicographical ordering of the unknowns leads to a banded matrix. Solving these matrices with a direct method will lead to substantial fill-in.
  - biconjugate gradient method with iLU preconditioner for CNR.
- ADI methods avoid banded matrices by rearranging the entries in each dimension and solving d subproblems, each being a tridiagonal matrix that can be solved in linear time.
- MCS ADI method requires $2d$ solves of tridiagonal matrices.

Convergence Rate for Sparse Grid methods

For full grid methods, the rate of convergence is computed based on a refinement of the step size $2h$ to h, i.e.

$$c = \log_2 \left(\frac{\text{error}_{2h}}{\text{error}_h} \right) = \log_2 \left(\frac{\text{error}_{N^d}}{\text{error}_{(2N)^d}} \right) = d \frac{\log(\text{error}_{N^d} / \text{error}_{(2N)^d})}{\log((2N)^d / N^d)}. \quad (9)$$

For sparse grids, no such uniform h exists, due to the multilevel nature of the method. Instead, we define the rate of convergence based on the number of unknowns.

To keep consistency with existing definitions, on a full grid we generalize this to M_1 and M_2 entries, where M_1 and M_2 do not necessarily share a common factor on different levels, we have

$$c = d \frac{\log(\text{error}_{M_1} / \text{error}_{M_2})}{\log(M_2 / M_1)}. \quad (10)$$

Theoretical efficiency of the Sparse Grid method

The sparse grid method allows us to use fewer degrees of freedom and can attain comparable error with the full grid method. This allows us to extrapolate to a finer resolution and outperform the full grid method.

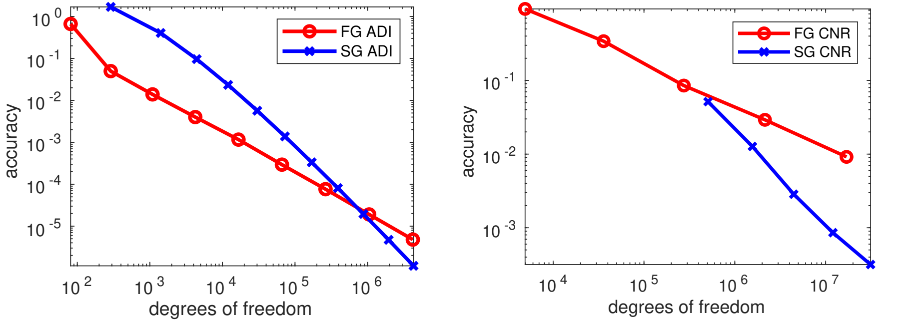

Figure 2: Plot of accuracy vs degrees of freedom for various option pricing problems. Left: Two-dim European geometric average put. Right: Three-dim European power put. The full grid method has a rate of convergence of 2.

Smoothing

The initial conditions for geometric average put are non-smooth, as there is a cusp at the curve where $(\prod S_i)^{1/d} = K$, hence, smoothing techniques are required to maintain the order of convergence of the discretization method, especially when we solve with the combination method.

We implement the smoothing techniques developed in [Kreiss et al., 1970] in multiple dimensions. These smoothing techniques restore the order of convergence, but require us to use uniform grids.

A complete discussion is beyond the scope of this talk; some derivations can be found in [Christara and Leung, 2018] and a multidimensional example which we followed can be found in [Düring and Heuer, 2015].

Parameters for Black-Scholes equations

We use the following table of parameters. For some problems, we adjust the value of the strike K to ensure that the computed values of different problems are on a roughly similar scale. This only affects the scale and not the accuracy of the solution.

Description	Symbol	value
Volatility of S_i	σ_i	0.4
Correlation of S_i with S_j	$\rho_{i,j}$	0.2
truncation of domain	$S_{\max}$	$8K$
interest rate	r	0.10
time of expiry	T	1.0
strike	K	100.0

Table 2: Table of parameters for option pricing problems with multiple underlyings.

European options – ADI method full grid

q	n_S	n_τ	value	error	rate	time (s)
1	81	10	7.956965184	-6.66e-01	—	1.33e-02
2	289	18	8.572781256	-4.99e-02	3.74	1.77e-02
3	1089	34	8.608731714	-1.39e-02	1.84	2.94e-02
4	4225	66	8.618665573	-4.00e-03	1.80	5.25e-02
5	16641	130	8.621504479	-1.16e-03	1.78	2.51e-01
6	66049	258	8.622372493	-2.93e-04	1.99	1.55e+00
7	263169	514	8.622589224	-7.62e-05	1.94	1.04e+01
8	1050625	1026	8.622646406	-1.90e-05	2.00	6.57e+01
9	4198401	2050	8.622660611	-4.78e-06	1.99	7.48e+02

Table 3: Two-dimensional European geometric put option; MCS ADI; full grid; nonuniform grid.

European options – ADI method sparse grid – no smoothing

q	n_S	n_τ	value	error	rate	time (s)
2	289	18	8.565056906	-5.76e-02	—	7.23e-01
3	1411	102	8.623991582	1.33e-03	4.76	5.26e-01
4	4421	330	8.611445333	-1.12e-02	-3.74	6.20e-01
5	11975	910	8.626333795	3.67e-03	2.24	6.42e-01
6	30153	2322	8.621898509	-7.67e-04	3.39	9.63e-01
7	72651	5654	8.622190480	-4.75e-04	1.09	1.33e+00
8	169933	13338	8.622676246	1.09e-05	8.89	2.97e+00
9	389071	30750	8.622686045	2.07e-05	-1.55	9.15e+00
10	876497	69666	8.622686950	2.16e-05	-0.11	3.37e+01
11	1949651	155686	8.622667262	1.87e-06	6.11	1.42e+02
12	4292565	344106	8.622670664	5.28e-06	-2.62	6.12e+02

Table 4: Two-dimensional European geometric put option; MCS ADI; sparse grid; nonuniform grid; no smoothing; parallel computation of subproblems.

European options – ADI method sparse grid – smoothing

q	n_S	n_T	value	error	rate	time (s)
2	289	18	10.319867703	1.70e+00	—	1.31e-01
3	1411	102	9.028073043	4.05e-01	1.81	1.64e-01
4	4421	330	8.719565958	9.69e-02	2.51	2.53e-01
5	11975	910	8.646111475	2.34e-02	2.85	4.06e-01
6	30153	2322	8.628351788	5.69e-03	3.07	7.75e-01
7	72651	5654	8.624043473	1.38e-03	3.22	1.68e+00
8	169933	13338	8.622998872	3.33e-04	3.34	4.24e+00
9	389071	30750	8.622746163	8.08e-05	3.42	1.21e+01
10	876497	69666	8.622684888	1.95e-05	3.50	3.96e+01
11	1949651	155686	8.622670093	4.71e-06	3.56	1.47e+02
12	4292565	344106	8.622666516	1.13e-06	3.62	6.14e+02

Table 5: Two-dimensional European geometric put option; MCS ADI; sparse grid; uniform grid; smoothing from [Kreiss et al., 1970]; parallel computation of subproblems.

Comparison of methods – two-dim European geometric put

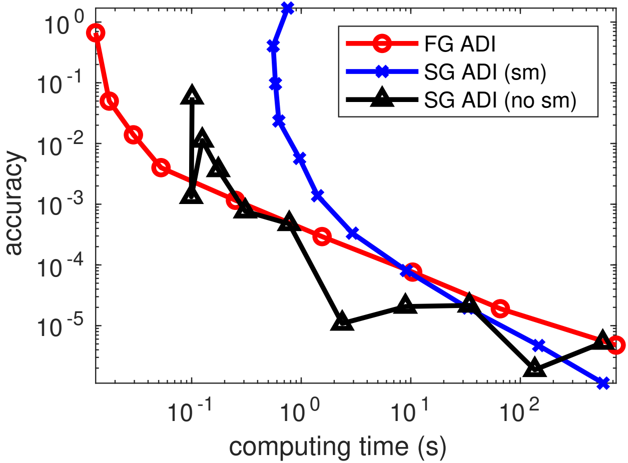

Figure 3: Two-dimensional European geometric put option: Comparison of accuracy vs computational time for MCS ADI, on full and sparse grids.

American options - preface

- To compute the numerical solution of American options, we used the penalty method introduced in [Forsyth and Vetzal, 2002].
- A straightforward approach of applying the penalty method (with uniform timesteps) leads to a deteriorated order of convergence. Although the order of accuracy in space remains $\mathcal{O}(\Delta x^2)$, the order of accuracy in time is $\mathcal{O}(\Delta \tau^{3/2})$.
- Some ways of correcting this issue:
  - [Forsyth and Vetzal, 2002] variable timestepping algorithm
  - [Reisinger and Whitley, 2014] quadratic transformation of the time points (i.e. $\tilde{t}_i = t_i^2$ on $t = [0, 1]$).
- We have used the quadratic transformation with CNR timestepping for American options.

Smooth Initial condition – Power put options

The power put option is defined with the payoff function

$$V^*(S_1, S_2, \ldots, S_d) = \max(K - \sum S_i/d, 0)^p \tag{11}$$

where $p = 1, 2, 3, \ldots$.

The payoff function V^* is continuous in C^{p-1}.

Power put initial conditions may not need smoothing; appropriate choices of p can be used to test smoothness requirements in PDE solvers.

We solve the European power put option with $p = 2$ and American power put option with $p = 3$. These are the minimum choices of p for the combination method to work ideally (i.e. with monotone convergence).

Unlike geometric put options, power put options do not have analytical solutions; hence, we test for self-convergence with successive differences. We note that the sparse grid method and the full grid methods are in agreement.

Two-dimensional European power put option

q	n_S	n_τ	value	difference	rate	time (s)
1	81	10	1.232509678	—	—	1.55e-03
2	289	18	1.270141301	3.76e-02	—	2.75e-03
3	1089	34	1.273231529	3.09e-03	3.61	6.31e-03
4	4225	66	1.273652425	4.21e-04	2.88	2.27e-02
5	16641	130	1.273746086	9.37e-05	2.17	1.78e-01
6	66049	258	1.273772945	2.69e-05	1.80	1.36e+00
7	263169	514	1.273779639	6.69e-06	2.00	9.65e+00
8	1050625	1026	1.273781411	1.77e-06	1.92	6.51e+01
9	4198401	2050	1.273781853	4.42e-07	2.00	7.85e+02

Table 6: Two-dimensional European power put option ($p = 2$); MCS ADI; full grid; nonuniform grid.

Two-dimensional European power put option

q	n_S	n_T	value	difference	rate	time (s)
2	289	18	1.270141301	—	—	4.77e-02
3	1411	102	1.273102768	2.96e-03	—	4.59e-02
4	4421	330	1.273577276	4.75e-04	2.97	1.10e-01
5	11975	910	1.273714911	1.38e-04	2.39	1.19e-01
6	30153	2322	1.273762184	4.73e-05	2.27	2.00e-01
7	72651	5654	1.273776321	1.41e-05	2.71	5.01e-01
8	169933	13338	1.273780426	4.11e-06	2.87	1.71e+00
9	389071	30750	1.273781570	1.14e-06	3.07	6.56e+00
10	876497	69666	1.273781885	3.15e-07	3.15	2.59e+01
11	1949651	155686	1.273781971	8.59e-08	3.23	1.10e+02

Table 7: Two-dimensional European power put option ($p = 2$); MCS ADI; sparse grid; nonuniform grid; parallel computation of subproblems.

Comparison of methods – two-dim European Power Put

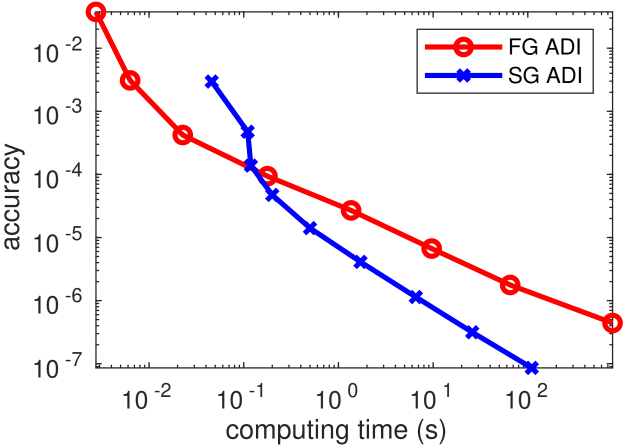

Figure 4: Two-dimensional European power put option ($p = 2$). Comparison of accuracy vs computational time for MCS ADI on full and sparse grids.

Two-dimensional American power put option

q	n_S	n_τ	pen. its	value	difference	rate	time (s)
1	81	10	13	4.626237994	—	—	2.60e-03
2	289	18	21	3.730721263	-8.96e-01	—	1.25e-02
3	1089	34	48	3.623769742	-1.07e-01	3.07	1.98e-01
4	4225	66	114	3.600040823	-2.37e-02	2.17	2.14e+00
5	16641	130	247	3.593702710	-6.34e-03	1.90	2.20e+01
6	66049	258	515	3.591970948	-1.73e-03	1.87	2.56e+02
7	263169	514	1027	3.591537960	-4.33e-04	2.00	2.48e+03

Table 8: Two-dimensional American power put option ($p = 3$); CNR; full grid; nonuniform grid.

Two-dimensional American power put option

q	n_S	n_τ	pen. its	value	difference	rate	time (s)
2	289	18	21	3.730721263	—	—	7.24e-02
3	1411	102	120	3.617977520	-1.13e-01	—	1.23e-01
4	4421	330	402	3.600797572	-1.72e-02	3.05	2.78e-01
5	11975	910	1165	3.593501819	-7.30e-03	1.66	8.76e-01
6	30153	2322	3042	3.592030868	-1.47e-03	3.40	3.51e+00
7	72651	5654	7526	3.591583572	-4.47e-04	2.67	1.57e+01
8	169933	13338	17861	3.591447883	-1.36e-04	2.77	8.11e+01
9	389071	30750	41490	3.591409706	-3.82e-05	3.04	4.26e+02
10	876497	69666	94928	3.591396103	-1.36e-05	2.53	2.58e+03

Table 9: Two-dimensional American power put option ($p = 3$); CNR; sparse grid; nonuniform grid; parallel computation of subproblems.

Comparison of methods – two-dim American power put

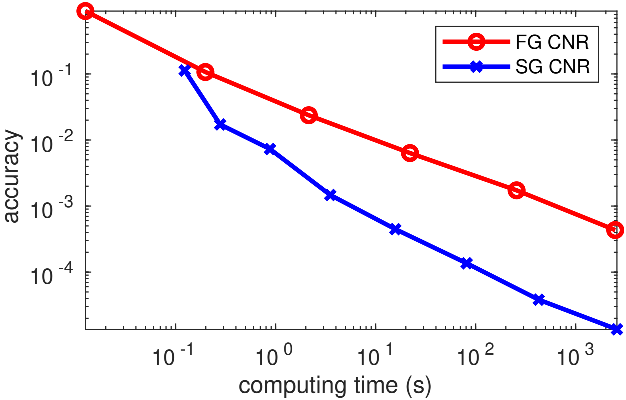

Figure 5: Two-dimensional American power put option with $p = 3$. Comparison of accuracy vs computational time for CNR on full and sparse grids.

Three dimensional European power put

q	n_S	n_T	biCG its	value	difference	rate	time (s)
1	729	10	24	19.035216628	—	—	9.08e-01
2	4913	18	62	18.100412803	-9.35e-01	—	3.34e-01
3	35937	34	160	17.759704363	-3.41e-01	1.46	3.64e+00
4	274625	66	580	17.674501714	-8.52e-02	2.00	1.42e+02
5	2146689	130	2001	17.645251968	-2.92e-02	1.54	2.69e+03
6	16974593	258	5859	17.636077651	-9.17e-03	1.67	1.87e+04

Table 10: Three-dim European power put option ($p = 4$); CNR method; full grid.

q	n_S	n_T	biCG its	value	difference	rate	time (s)
4	145418	660	6278	17.7009716	—	—	2.00e+00
5	507075	2470	20609	17.6495116	-5.15e-02	—	1.06e+01
6	1563327	7998	58686	17.6368118	-1.27e-02	3.64	7.02e+01
7	4456254	23644	150443	17.6339621	-2.85e-03	4.20	4.11e+02
8	12032192	65664	345579	17.6331050	-8.57e-04	3.58	2.11e+03
9	31222853	174250	738635	17.6327880	-3.17e-04	3.09	1.04e+04

Table 11: Three-dim European power put option ($p = 4$); CNR method; sparse grid; parallel computation of subproblems.

Comparison of methods – three-dim European power put

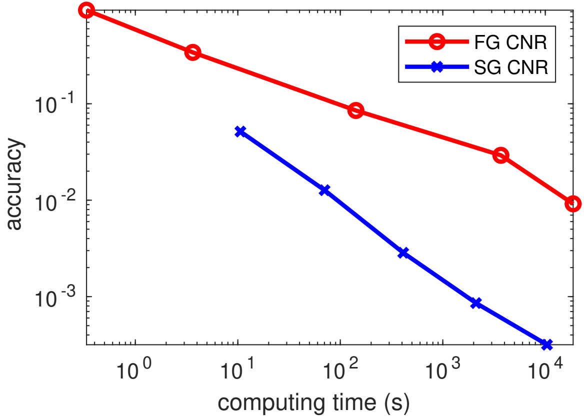

Figure 6: Three dimensional European power put option ($p = 4$), solved by CNR on full and sparse grids.

Conclusions & Future work

The sparse grid method

- Is a powerful extrapolation-like technique for the numerical solution of multi-dimensional PDEs.
- It allows us to compute solutions accurate to a degree that would be prohibited by the full grid method due to memory and time limitations.
- The combination formulation is applicable to variable coefficient PDEs with all derivative terms, unlike the hierarchical finite element method.
- The combination formulation allows straightforward parallelization of the subproblems.
- Using smoothing techniques derived in [Kreiss et al., 1970], we can ensure consistency of the order of convergence. However, this restricts us to use uniform grids, which is not ideal for many option pricing problems.

We are currently working on smoothing techniques for nonuniform grids.

References I

Balder, R. and Zenger, C. (1996).
The solution of multidimensional real Helmholtz equations on sparse grids.
SIAM Journal on Scientific Computing, 17(3):631–646.

Christara, C. and Leung, N. C.-H. (2018).
Analysis of quantization error in financial pricing via finite difference methods.
SIAM Journal on Numerical Analysis, 56(3):1731–1757.

Düring, B. and Heuer, C. (2015).
High-order compact schemes for parabolic problems with mixed derivatives in multiple space dimensions.
SIAM Journal on Numerical Analysis, 53(5):2113–2134.

Forsyth, P. and Vetzal, K. (2002).
Quadratic convergence for valuing American options using a penalty method.
SIAM Journal on Scientific Computing, 23(6):2095–2122.

References II

Griebel, M., Schneider, M., and Zenger, C. (1992).
A combination technique for the solution of sparse grid problems.
Proceedings of the IMACS International Symposium on Iterative Methods in Linear Algebra, pages 263–292.

Kreiss, H.-O., Thomée, V., and Widlund, O. (1970).
Smoothing of initial data and rates of convergence for parabolic difference equations.
Communications on Pure and Applied Mathematics, 23(2):241–259.

Reisinger, C. and Whitley, A. (2014).
The impact of a natural time change on the convergence of the Crank–Nicolson scheme.
IMA Journal of Numerical Analysis, 34(3):1156–1192.

References III

Smolyak, S. A. (1963).
Quadrature and interpolation formulas for tensor products of certain classes of functions.
In *Doklady Akademii Nauk*, volume 148, pages 1042–1045. Russian Academy of Sciences.

Wyns, M. et al. (2016).
Convergence of the modified Craig–Sneyd scheme for two-dimensional convection–diffusion equations with mixed derivative term.
Journal of Computational and Applied Mathematics, 296:170–180.

Yserentant, H. (1986).
On the multi-level splitting of finite element spaces.
Numerische Mathematik, 49(4):379–412.