

An efficient Sparse Grid Method for the high dimensional Fokker Planck Equation

Christian Feuersänger
Institute for Numerical Simulation
University of Bonn, Germany

HDA07



Introduction: the Fokker Planck Equation

- is d -dimensional (parabolic) partial differential equation

$$\partial_t p(x, t) - \frac{1}{2} \sum_{i,j=1}^d \partial_{x_i} \partial_{x_j} \left(\{ B(x, t) \cdot B(x, t)^T p(x, t) \}_{i,j} \right) + \sum_{i=1}^d \partial_{x_i} \left(\{ A(x, t) p(x, t) \}_i \right) = 0$$

- with $x \in \mathbb{R}^d$, $p(x, 0) = p_0(x)$, $\int_{\mathbb{R}^d} p(x, t) dx = 1$, $p(x, t) \geq 0$

Fokker Planck Equation *equivalent to* \longrightarrow stochastic differential equation



Introduction: the Fokker Planck Equation

- is d -dimensional (parabolic) partial differential equation

$$\partial_t p(x, t) - \frac{1}{2} \sum_{i,j=1}^d \partial_{x_i} \partial_{x_j} \left(\{ B(x, t) \cdot B(x, t)^T p(x, t) \}_{i,j} \right) + \sum_{i=1}^d \partial_{x_i} \left(\{ A(x, t) p(x, t) \}_i \right) = 0$$

- with $x \in \mathbb{R}^d$, $p(x, 0) = p_0(x)$, $\int_{\mathbb{R}^d} p(x, t) dx = 1$, $p(x, t) \geq 0$

Fokker Planck Equation *equivalent to* \longrightarrow stochastic differential equation

- describes transition probabilities
 $P(X(t) \in dx | X(0) = X_0) = p(x, t) dx$
- $p(x, t)$: probability density function
- $X(t)$: random variable



Introduction: the Fokker Planck Equation

- is d -dimensional (parabolic) partial differential equation

$$\partial_t p(x, t) - \frac{1}{2} \sum_{i,j=1}^d \partial_{x_i} \partial_{x_j} \left(\{ B(x, t) \cdot B(x, t)^T p(x, t) \}_{i,j} \right) + \sum_{i=1}^d \partial_{x_i} \left(\{ A(x, t) p(x, t) \}_i \right) = 0$$

- with $x \in \mathbb{R}^d$, $p(x, 0) = p_0(x)$, $\int_{\mathbb{R}^d} p(x, t) dx = 1$, $p(x, t) \geq 0$

Fokker Planck Equation *equivalent to* \longrightarrow

stochastic differential equation

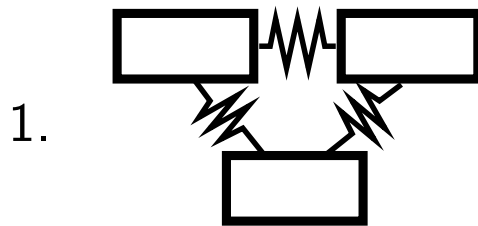
- describes transition probabilities
 $P(X(t) \in dx | X(0) = X_0) = p(x, t) dx$
- $p(x, t)$: probability density function
- $X(t)$: random variable

- $dX(t) = A(X, t) dt + B(X, t) dW_t$
- read as $\dot{X} = A + B \frac{dW_t}{dt}$
- " $\frac{dW_t}{dt}$ " := white noise

\rightsquigarrow involved analysis (Itô Lemma)



Application examples



movable objects, coupled with springs, excited by environmental load. # of oscillating objects \rightsquigarrow scales d

2. pricing of Credit Derivatives if historical values are used to estimate future performance

→ number of involved firms scales d

- Both yield $dX = A dt + BdW_t \in \mathbb{R}^d$
- FPE and SDE formulation **equivalent** because ...



Values of Interest are

- **mean value** of result **at time T**

$$\langle X_T \rangle = \left(\int_{\mathbb{R}^d} p(x, T) x_i dx \right)_{i=1, \dots, d}$$

- **covariance matrix** of result at time T

$$\langle (X_T - \langle X_T \rangle) \cdot (X_T - \langle X_T \rangle)^T \rangle = \left(\int_{\mathbb{R}^d} p(x, T) x_i x_j dx - \langle X_T \rangle_i \langle X_T \rangle_j \right)_{i, j=1, \dots, d}$$

→ moments of $X_T \equiv$ integrals of $p(x, T)$

↔ solution of SDE \Leftrightarrow solution of FPE



Numerical Methods

- choice between stochastic formulation or Fokker Planck Equation
 - SDE: Monte Carlo sampling methods
 - slow convergence
 - efficiency does not degenerate with increasing dimension
 - FPE: finite elements/differences
 - fast convergence
 - for $d \leq 3$: **more efficient** than Monte Carlo
 - $d > 3$ untractable: exponential cost in d (“curse of dimension”)
- In the following: develop solver for **FPE in higher dimensions**
- task: reduce “curse of dimension”, maintain efficiency



Numerical Solution of FPE - Discretisation aspects

$$\partial_t p(x, t) - \frac{1}{2} \sum_{i,j=1}^d \partial_{x_i} \partial_{x_j} \left(\{ B(x, t) \cdot B(x, t)^T p(x, t) \}_{i,j} \right) + \sum_{i=1}^d \partial_{x_i} \left(\{ A(x, t) p(x, t) \}_i \right) = 0$$

1. Cut domain: $\mathbb{R}^d \rightsquigarrow \Omega := [a, b]^d$, $p(x, t) = 0$ on $\partial\Omega$
2. time discretisation: adaptive timestepping method
 - goal: get solution $p(x, T)$ at **end** time T
3. space discretisation: sparse grid galerkin method
 - accuracy depends on (1.)
 - discretise initial value (often: $p(x, 0) =$ narrow normal distribution)
4. 1.-3. precise “enough” $\Rightarrow \int_{\Omega} p(x, t) dx \approx 1$, $p(x, t) \geq 0$.
 → precision check: $|\int_{\Omega} p(x, t) dx - 1| \gg 0 \Rightarrow$ 1.-3. **not** precise enough

Time discretisation: Implicit extrapolation method

- by Bader/Deuflhard '83
- employs extrapolation to get
 - accurate solutions
 - efficient and reliable **adaptive control** of δ_t
- solves up to error $\leq \epsilon$
- for smooth solution:
 - δt can be large, **comparable with p-adaptivity**



Space Discretisation - galerkin method

- after time discretisation: **elliptic problems** of type

$$-\sum_{i,j=1}^d \partial_{x_i} \left(a_{i,j}(x) \partial_{x_j} p(x) \right) + \sum_{i=1}^d b_i(x) \partial_{x_i} p(x) + c(x)p(x) = f(x)$$

- we assume **product-type** coefficients $C(x) = C_1(x_1) \cdots C_d(x_d)$ for $a_{i,j}$, b_i and c
- galerkin method $\forall v \in V_h : a(u, v) = \int_{\Omega} f v \, dx$ yields

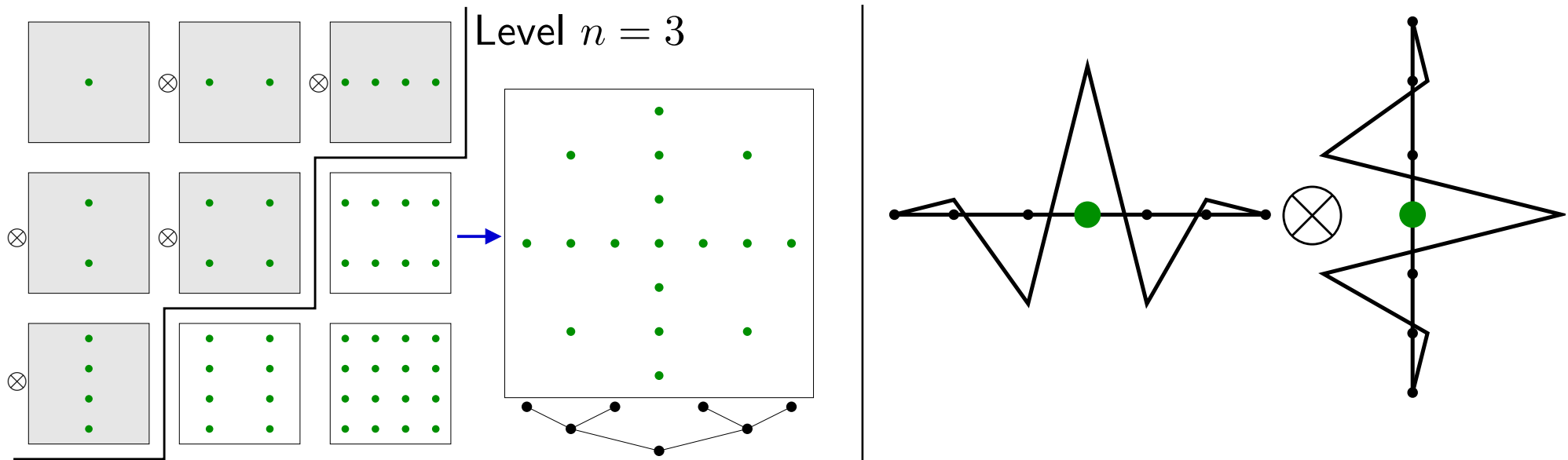
- linear equation system

$$A\bar{u} = \bar{f}$$



Choose V_h : Sparse Grid space

- tensor product, hierarchical splitting with prewavelet basis

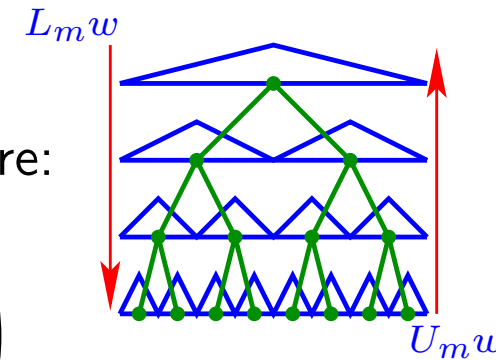


- $N := |V_h| = O(2^n n^{d-1})$ (instead of $O(2^{dn})$)
- finest mesh width 2^{-n} . Each dir: binary tree



Solving the linear system

- multilevel basis \rightsquigarrow matrix assembly $\notin O(N) \rightsquigarrow$ fast Aw required
- product structure allows splitting $Aw = (\prod_{m=1}^d A_m)w$ in 1d Operators
- key idea (Balder/Bungartz): **split 1d part** $A_m w := L_m w + U_m w$:
 - nested basis functions, 1d binary tree structure allow:
 - $L_m w, U_m w$ each in time $O(N)$
 - d -dim: combine **recursively**; respect sparse grid structure:



$$Aw = \left(\prod_{m=1}^d A_m w \right) = \left(\prod_{m=1}^{d-1} A_m \right) U_d w + L_d \left(\prod_{m=1}^{d-1} A_m w \right)$$

- eval. time $O(2^{d+1}N)$; memory $O(dN)$. Laplacian: $O(d2^{d+1}N)$ time!



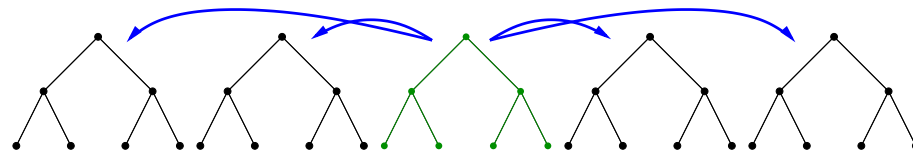
Eliminating the 2^{d+1}

- Idea: prewavelets are **semiorthogonal** with respect to $\langle \cdot, \cdot \rangle_{L_2}$

→ in case $A_m = 1d$ mass part $\int_0^1 uv \, dx$: A_m (almost) diagonal

→ expensive recursion not necessary in this direction

prewavelets nested: ✓,
 but large support. Also
 necessary: neighbors



- one can show: recursion still ok, if neighbors exist
 - regular sparse grid: ✓ ; adaptive: with grid modification ✓
 - laplacian, convection, low dim coeffs $c(x) = c_1(x_{q_1}) \cdots c_r(x_{q_r})$:
 → time $\mathbf{O(dN)}$, mem $\mathbf{O(N)}$

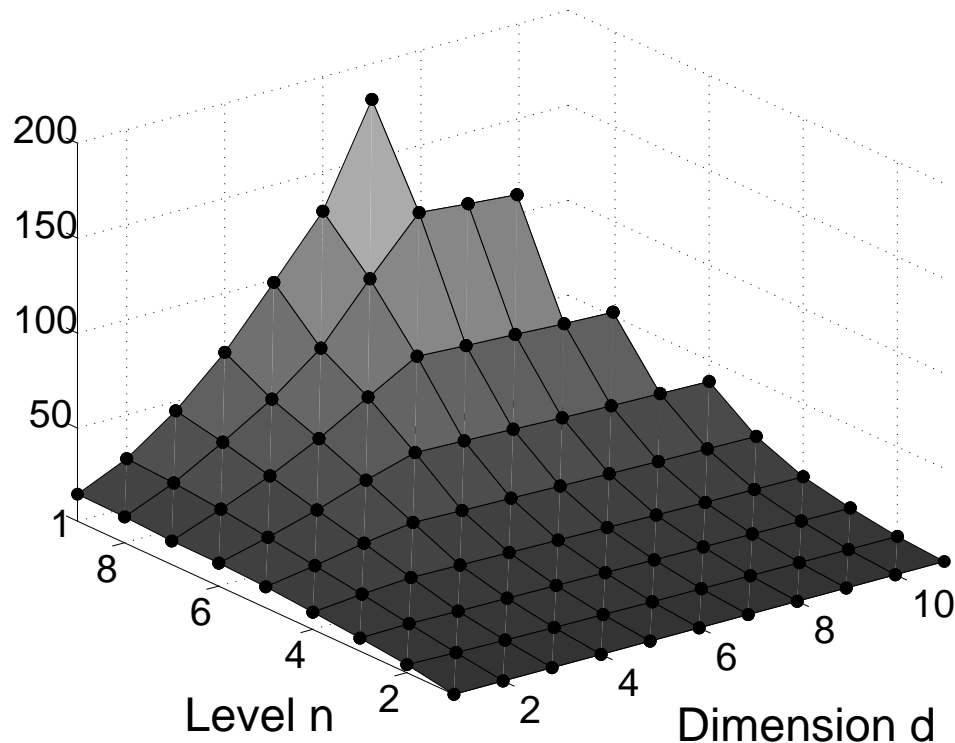


Storing and accessing sparse grid points

- work abstract with multiindices $(l_1, \dots, l_d | i_1, \dots, i_d)$
- access $u_{\mathbf{l}, \mathbf{i}}$: with **fast indexing routine** $U[r(\mathbf{l}, \mathbf{i})]$ in array $U[\cdot]$
- time for grid traversal: “almost” $O(N)$ (**quasi optimal**)
 - “almost”: **one** value $r(\mathbf{l}, \mathbf{i})$ can be re-used **many** times
 - compute one $r(\mathbf{l}, \mathbf{i})$ in time $O(d)$
- storage: $N \cdot \text{sizeof}(\text{double})$ **optimal**



Condition of linear systems: Laplacian



- diagonal preconditioning with prewavelet basis

- theory Griebel/Oswald '98: for $n \rightarrow \infty$:

$$\text{Cond}(n, d) = O(c^d)$$

$$\leftarrow \text{Cond}(n, d) = O(c^{\min\{d, n\}})$$

- potential problem; however: condition numbers low

- experience: # Iterations between 15 and 80 (also in unsymmetric case)



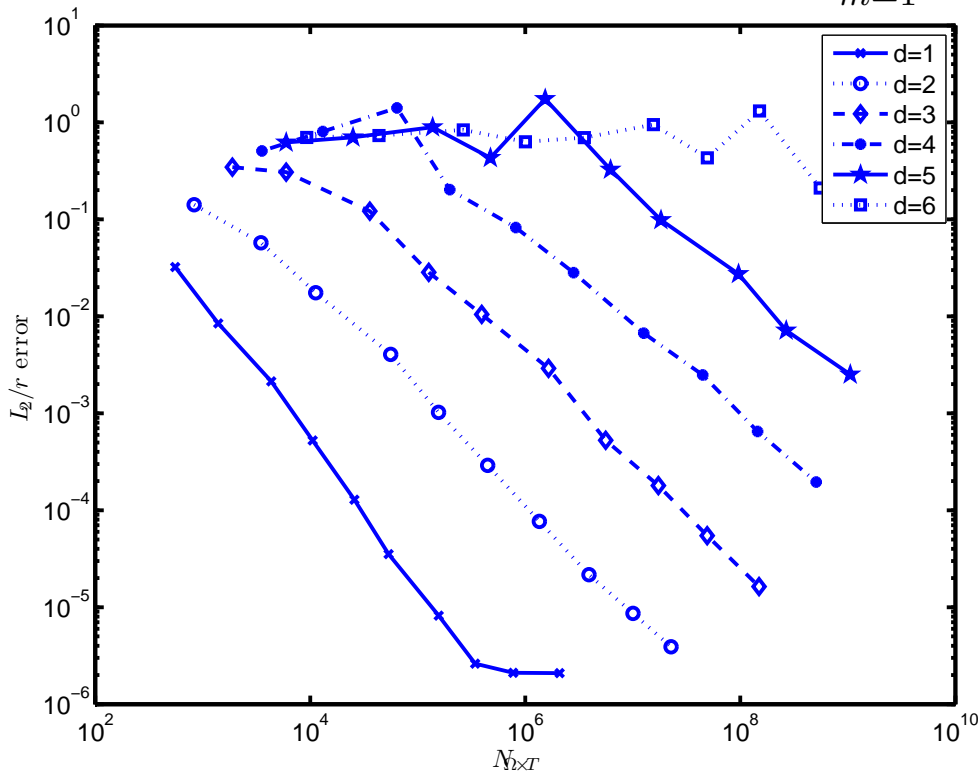
Numerical Results

- assume error is dominated by space discretisation
 - solve up to **space** discretisation error
 - choose ϵ for time stepping accordingly
 - solve linear systems iteratively up to space discretisation order
- measure error at end time T



Ornstein-Uhlenbeck process: relative L_2 error $p(\cdot, T)$

$$\partial_t p(x, t) - \frac{1}{2} \sigma^2 \Delta p(x, t) + \sum_{m=1}^d (\theta \mu - \theta x_m) \partial_{x_m} p(x, t) - d \theta p(x, t) = 0$$



- Initial value: Gaussian, $\Sigma_{ii} \approx 0.1$
- $\Omega = [-2, 5]^d$
- $N_{\Omega \times T} := \# \text{ lin. EQS} \cdot |V_h|$
- **table below:** $\text{err}(N_{\Omega \times T}) = N_{\Omega \times T}^{-\alpha}$
- analyse $\text{err}(n, d) / \text{err}(n + 1, d)$:
 $\rightsquigarrow \text{err}(n) = O(2^{-2n} n^{d-1})$ or better
- offsets between plots $\rightsquigarrow \text{err}(d) = O(c^d)$

	$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 6$
--	---------	---------	---------	---------	---------	---------

in plot: α

in plot: α	1.8	1.2	1.1	1	0.9	-
-------------------	-----	-----	-----	---	-----	---

expected for **uniform** $\delta_t = 2^{-n}$

expected for uniform $\delta_t = 2^{-n}$	1	0.8	0.7	0.6	0.45	0.35
---	---	-----	-----	-----	------	------

Ornstein-Uhlenbeck process: covariance in $d = 6, 7$

- choose smaller domain $\Omega := [-1, 4]^d$ instead of $[-2, 5]^d$

d	n	$ V_h $	N_T	Cov err	Cov err/r	Mean err	Mean err/r	time	mem	
6	12	13 516 801	42	3.1e-03	7.4e-03	7.1e-04	5.3e-04	155h	2.5GB	(*)
7	12	37 175 295	32	2.78	6.6	0.5	0.37	115h	15GB	(**)

N_T = num of timesteps

(*) using BiCGStab

(**) using GMRes

- $d = 6$: accurate solution
- $d = 7$: with $n = 12$ not accurate enough

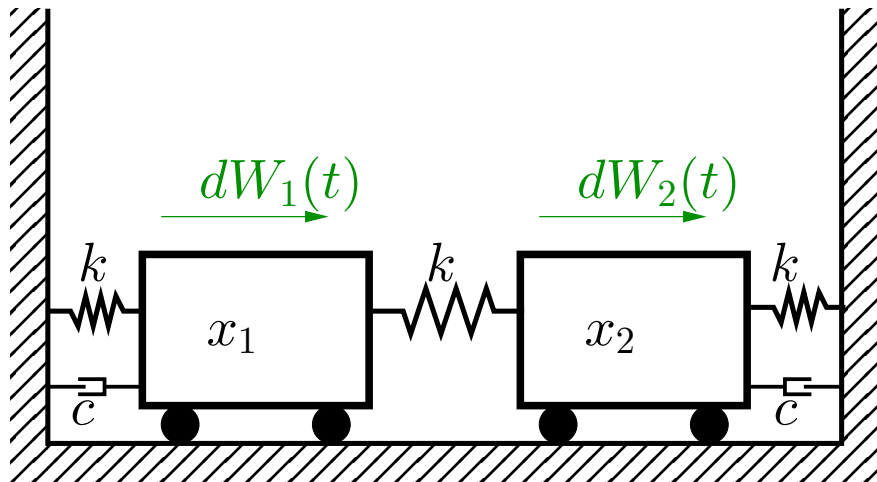


Remarks: Precision vs increasing d

- no error reduction before level $n = d + 1$ observed
→ lots of points required
 - for interpolation known: $|u|_2 = \left\| \frac{\partial^{2d}}{\partial_1^2 \dots \partial_d^2} u \right\|$ influences error
 - $u = \text{pdf of Normal Distribution} \rightsquigarrow |u|_{2, \{\infty, 2\}} = c^d$
 - $\Omega = [a, b]^d \rightsquigarrow$ linear coordinate trafo $\phi: [0, 1]^d \rightarrow \Omega$
 - $|u \circ \phi|_2$ leads to $(b - a)^{2d}$
- “constants” move n -asymptotics beyond range



Application example: Coupled oscillators



- Accelerated by random force
- linear **benchmark system**: 2 degrees of freedom, second order

$$\ddot{x}_1 = -2kx_1 - c\dot{x}_1 + kx_2 + \sqrt{2D} dW_1(t)$$

$$\ddot{x}_2 = kx_1 - c\dot{x}_2 - 2kx_2 + \sqrt{2D} dW_2(t)$$

- write as first order system $\rightsquigarrow d = 4$ FPE, $p(x, 0)$: normal distribution

- search for **variance** of elongation (mean = 0), $\Omega = [-4, 4]^4, T = [0, 15]$

		$ V_h $	timesteps	Cov err	Cov err/r	time	mem
sparse grid	$n = 10$	178177	1500	7.76e-04	1.55e-03	65h	70MB
Wojtkiewicz/ Bergman	$N = 41^4$	2 825 761	≥ 3000	5.9e-03	1.18e-02	-	-



Application example: Credit Derivatives Pricing

- (Escobar/Seco) Let $X_i(t) := \log(\text{Value of firm } i)$ and $\mathbf{X} := (X_1, \dots, X_d)$. Search for joint probability that
 - for all i :
 - value of firm i in dx_i
 - firm i has not defaulted before time t : $\underline{X}_i(t) := \min_{s \leq t} X_i(s) \geq b_i$

$$P\left(\mathbf{X}(t) \in d\mathbf{x}, \underline{X}_1(t) \geq b_1, \dots, \underline{X}_d(t) \geq b_d\right) = p(\mathbf{x}, t) d\mathbf{x}$$

- density $p(x, t)$ is solution of FPE

$$\partial_t p(x, t) - \frac{1}{2} \text{div}(\Sigma p(x, t)) + \mu \cdot \nabla p(x, t) = 0$$

with **absorbing boundary** $p(x, t) = 0$ for $x_i = b_i$.



Application example: Credit Derivatives Pricing (2)

- Probabilities of interest: Integrals over p . Example:

$$\tilde{P}(T) := \int_{\Omega} p(x, T) dx = P(\underline{X}_1(T) \geq b_1, \dots, \underline{X}_d(T) \geq b_d)$$

→ Probability, that no firm has defaulted before T years.

- compute $p(x, T)$, then integrate over domain
- Two numerical examples with selected parameters:

	d	$\tilde{P}(3)$	n	$ V_h $	timesteps	time	mem	I_0
Example 1:	4	60.6%	13	3 080 193	200	46h	1.3GB	1.9e-4
Example 2:	5	51%	12	4 374 527	124	51h	1.8GB	6e-2

- reliability indicator $I_0 := \left| \int_{\Omega} p(x, t_0) dx - 1 \right| \stackrel{!}{=} 0$



Summary and Conclusion

- Summary
 - algorithms possible in time $O(dN)$ (“almost”) and memory $O(N)$
 - condition numbers increase exponentially in d
 - adaptive time stepping allows efficient time discretisation
- Conclusion
 - presented method can **solve FPE applications for $d \in \{4, 5, 6\}$** , maybe $d = 7$
 - still: accuracy low with increasing d
 - drops exponentially in d
 - asymptotics starts late \rightsquigarrow lots of points required
 - **“curse of dimension” reduced**: shifted from $d = 4$ to $d = 8$



Thank you for the attention

References

- [1] H.J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, pp. 1-123, Cambridge University Press, 2004.
- [2] R. Balder and C. Zenger. The d-dimensional helmholtz equation on sparse grids. *Technische Universität München, SFB-Bericht Nr. 342/21/92 A*, 1992.
- [3] S.F. Wojtkiewicz and L.A. Bergman. Numerical solution of high dimensional fokker - planck - equations. *Specialty Conference on Probabilistic Mechanics and Structural Reliability*, 8, 2000.
- [4] M. Escobar and L. Seco. A partial differential equation for credit derivatives pricing. *CRM proceedings of the conference on Partial Differential equations in very high dimensions*, 2005.
- [5] C. Feuersänger. Dünngitterverfahren für hochdimensionale elliptische partielle Differentialgleichungen. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2005.

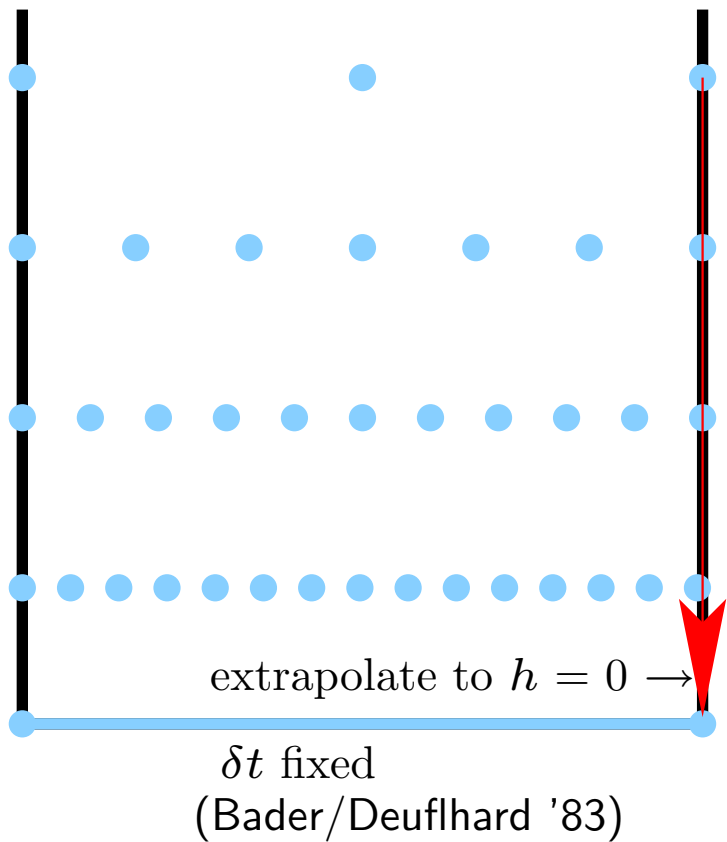


Thank you for the attention

- [6] S. Achatz. *Adaptive finite Dünngitter-Elemente höherer Ordnung für elliptische partielle Differentialgleichungen mit variablen Koeffizienten*. Doktorarbeit, Technische Universität München, 2003.



Implicit extrapolation method



- for each step $t \rightarrow t + \delta_t$:
 - solve with multiple step sizes h_i
 - each $u(\cdot) \rightarrow u(\cdot + h_i)$: solve linear system (ell. problem)
- extrapolate to $h = 0$
 - accurate solution by extrapolation
 - efficient and reliable **adaptive control** of δ_t^{next}
 - solve up to $\text{err} \leq \epsilon$
- for smooth solution:
 - δt large, **comparable with p-adaptivity**

