Sparse Grids: Higher Dimensionalities and HPC Aspects

Hans-Joachim Bungartz

TUM – Department of Informatics, Chair of Scientific Computing
Computational Science & Engineering (CSE) ➔ Key Technology for Science & Industry
High-Performance Computing (HPC) ➔ Core Enabler for CSE

Mathematical model
\[
\frac{\partial u}{\partial t} + (u \cdot \nabla) u + \frac{1}{\rho} \nabla p - \nu \Delta u = 0
\]
\[
A\dot{u}_h + D u_h + C(u_h) u_h - M^T p_h / \rho = 0
\]

Impact of each step on all other steps!
Hence #1: no pipeline any more, no cycle, but a complete graph
Hence #2: less space for single-field experts
CSE&HPC Challenges

“Multi-X” increase complexity

- From parameter assumptions …
  ... to identification & estimation
- From forward problems …
  ... to inverse problems
- From deterministic models …
  ... to random & uncertainty
- From one (spatial/temporal) scale …
  ... to cascades of scales
- From single-physics problems …
  ... to coupled scenarios
- From qualitative descriptions …
  ... to quantitative prediction
- From simulation …
  ... to optimisation
- From data / images / numbers …
  ... to information / insight
- From counting operations …
  ... to energy awareness
- From sequential algorithm design …
  ... to massive parallelism
- From simple tools & codes …
  ... to 2x complex ones
- From heroic PhD codes …
  ... to large teams / SW
- From hacker’s delight …
  ... to complex workflows
- From one-way batch jobs …
  ... to user interaction
- From island fun …
  ... to embedding & integration
- From flat algorithms & data …
  ... to hierarchy
- From multi-level
  multi-modal
  multi-disciplinary
  multi-dimensional
  multi-core
  multi-physics
  multi-architecture
CSE&HPC Challenges

- From parameter assumptions ... to identification & estimation
- From forward problems ... to inverse problems
- From deterministic models ... to random & uncertainty
- From one (spatial/temporal) scale ... to cascades of scales
- From single-physics problems ... to coupled scenarios
- From simulation ... to optimisation
- From one-way batch jobs ... to user interaction
- From simple tools & codes ... to 2x complex ones
- From heroic PhD codes ... to large teams / SW
- From hacker’s delight ... to complex workflows
- From island fun ... to complex workflows
- From data / images / numbers ... to information / insight
- From counting operations ... to energy awareness
- From sequential algorithm design ... to massive parallelism
- From flat algorithms & data ... to hierarchy
- From island fun ... to embedding & integration

multi-dimensional
Contents

Why High Dimensionalities?

Sparse Grids – Fundamentals

Sparse Grids – Applications

Sparse Grids in the ExaHD Project

By the way: no permanent torture “slide 6 of 999” – there will be 58.
A Hot Topic: High-Dimensional Numerics

• **High**: not 2, not 3, not 3 plus time, but everything beyond (10s, 100s, …)
  
  *note: in literature, often “high” means more than “higher” 😊*

• **Where?**
  - quantum mechanics
  - finance, statistics & stochastics
  - parameter identification, optimisation (search in high-dim parameter spaces)
  - data mining, classification, information extraction

• **Why a problem?**
  - FEM: think of 11-dimensional hyper-tetrahedra and their adaptive refinement … 😊
  - Computational demand – the **curse of dimension**: …
    - the cheapest 1-D discretisation … … and in 100-D?
      
      \[
      1^{100} = 1 \quad \text{cost 😊, benefit 😊}
      \]

    - the second cheapest 1-D discretisation … … and in 100-D?
      
      \[
      2^{100} \approx 10^{30} \quad \text{benefit ? }, \text{ cost 😊}
      \]
A Hot Topic: High-Dimensional Numerics

Remember: “Exa” stands for $10^{18}$!

- **Why a problem?**
  - FEM: think of 11-dimensional hyper-tetrahedra and their adaptive refinement ...
  - Computational demand – the **curse of dimension**:
    - the simplest 1-D discretisation ...
    - ... and in 100-D?
    - $10^{100} = 1$
    - the second simplest 1-D discretisation ...
    - ... and in 100-D?

$2^{100} \approx 10^{30}$
Contents

Why High Dimensionalities?

Sparse Grids – Fundamentals
- not a solver, not multigrid …
- not a sparse matrix approach …
- not next-generation FEM …
- but a grid-point selection scheme combinable with all the above …

Sparse Grids – Applications

Sparse Grids in the ExaHD Project
Cost and Benefit of Numerical Algorithms

- For a continuous problem $P$ with solution $u$, provide discrete approximations $P_n$ and $u_n$ such that some error $\varepsilon$ gets small.

- **Cost-benefit consideration (cf. notion of $\varepsilon$-complexity):**
  - **benefit:** obtained error/accuracy $\varepsilon$
  - **cost (I):** the classical ones – $N(\varepsilon)$ degrees of freedom to obtain $\varepsilon$, $g(N)$ ops. to solve $P_n$
  - **cost (II):** (parallel) runtime $T(g(N))$, energy consumption $E(g(N))$ for building & solving $P_n$
  - **cost (III):** more metrics – memory efficiency, cache efficiency, communication avoiding, …

- **Classical approach ($d$ dimensions, regularity $p$):**
  - exponential growth of $N$: $N = O(\varepsilon^{-d/p})$ (curse of dimension)
  - polynomial growth of $g$ in $N$: $g = O(N^k)$
  - $T$ and $E$ just observed, tunings for $T$, more experience than theory … [but be aware! 😊]

- **Example:**
  - $\varepsilon = 10^{-4}$ (4 digits), $p = 1$ (first-order regularity), $d = 4$ (think of CM – space plus time)
  - $k = 2$ (a non-optimal non-multilevel solver, as it is still widespread)
  - result: $N = O(10^{16})$, $g = O(10^{32})$, $T & E$ already relevant at all?
Cost and Benefit of Numerical Algorithms

- For a continuous problem $P$ with solution $u$, provide discrete approximations $P_n$ and $u_n$ such that some error $\varepsilon$ gets small.

- **Cost-benefit consideration** (cf. notion of $\varepsilon$–complexity):
  - benefit: obtained error/accuracy $\varepsilon$
  - cost (I): the classical ones – $N(\varepsilon)$ degrees of freedom to obtain $\varepsilon$, $g(N)$ ops. to solve $P_n$
  - cost (II): (parallel) runtime $T(g(N))$, energy consumption $E(g(N))$ for building & solving $P_n$
  - cost (III): more metrics – memory efficiency, cache efficiency, communication avoiding, …

- **Classical approach** ($d$ dimensions, regularity $p$):
  - exponential growth of $N$: $N = O(\varepsilon^{-dp})$ (curse of dimension)
  - polynomial growth of $g$ in $N$: $g = O(N^k)$
  - $T$ and $E$ just observed, tunings for $T$, more experience than theory … [but be aware! 😄]

- **Starting points for improvements**:
  - fast solvers for $P_n$ (multilevel): $g$ optimal (linear in $N$, for example)
  - efficient discretizations: improve $N$ (small, no or only weak $d$-dependence)
    - approaches of higher order, adaptive mesh refinement (problem-dependent)
    - structural considerations (a priori grid structure, not depending on given problem)
A Priori Structure? Cf. Numerical Quadrature!

- **Gauß** quadrature: higher degree of accuracy via non-equidistant choice of nodes (exact up to degree $2n-1$)
- multivariate formulas: inequality of **Koksma** and **Hlawka**

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \int_{[0,1]^d} f(x)dx \right| \leq V(f) \cdot D_n^*(x_1, \ldots, x_n),$$

where
- $V$: variation (indicates the integrand's smoothness, depends on problem)
- $D$: discrepancy (indicates deviation of the grid points' distribution from a uniform one; depends on method)

$$D_n^*(x_1, \ldots, x_n) := \sup_{E \in \mathcal{M}} \left| \frac{1}{n} \sum_{i=1}^{n} \chi_E(x_i) - \int_{[0,1]^d} \chi_E(x)dx \right|,$$

$$\mathcal{M} := \left\{ [0,a_1] \times \ldots \times [0,a_d] \subseteq [0,1]^d \right\}$$

- **Smolyak** quadrature, **Babenko’s** hyperbolic crosses
- **Archimedes** quadrature (divide & conquer), **Cavalieri** for $d>1$
A Priori Structure? Cf. “Battleship” ("Schiffe versenken")!

Minimizing discrepancy means: Minimize the rectangular area where a ship can hide.
A Priori Structure? Cf. “Battleship” ("Schiffe versenken")!

Minimizing discrepancy means: Minimize the rectangular area where a ship can hide.
A Priori Structure? Cf. “Battleship” ("Schiffe versenken")!

Minimizing discrepancy means: Minimize the rectangular area where a ship can hide.
A Priori Structure – 4 More Points

The biggest (abstract 😊) ship that can be hidden?
A Priori Structure – 4 More Points

The biggest (abstract 😊) ship that can be hidden?
Sparse Grids in a Nutshell

Starting point: finite element thinking!

*How to choose grids, elements, basis functions?*

- \( d=1 \): **hierarchical bases** (here linear)
- \( d>1 \): **tensor product** approach
- **hierarchical subspaces**: collect all basis functions with support of same aspect ratio
- **regularity**: spaces \( X(\Omega) \) of bounded mixed derivatives
- discretization / approximation as an **optimisation problem**; find optimum choice of subspaces (cf. **knapsack problem**)

\[
\max_{u \in X(\Omega): \|u\|_1 = 1} \|u - u_{V_{\text{opt}}}\|
\]

\[
= \min_{U: |U|=N} \max_{u \in X(\Omega): \|u\|_1 = 1} \|u - u_U\|
\]

- result: **sparse grids**
  
Zenger & co-workers, 1990
Sparse Grids in a Nutshell (cont’d)

- **Appearance:**

- **Cost** (number of grid points) vs. **benefit** (contribution to interpolant):
  
<table>
<thead>
<tr>
<th># grid points</th>
<th>sparse</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td>error (max, $L_2$)</td>
<td>$O(h_n^{p+1} r^{d-1})$</td>
<td>$O(h_n^{p+1})$</td>
</tr>
<tr>
<td>error (energy)</td>
<td>$O(h_n^p)$</td>
<td>$O(h_n^p)$</td>
</tr>
</tbody>
</table>

  (finest mesh width $h_n = 2^{-n}$, hierarchical bases of piecewise degree $p$)

- **Extensions:** straightforward access to adaptive refinement, generalisation to.piecewise polynomial hierarchical bases, energy-optimal sparse grids
Ways to Sparse Grid Solutions

- **Direct discretization on sparse grid (a “single-grid” view):**
  - full flexibility with respect to ansatz spaces, adaptive refinement, …
  - requires a complete implementation from scratch

- **Indirect access via combination technique (a “multi-grid view”):**
  - extrapolation-type approach – variants: classical, dimension-adaptive, opti-com, ...
  - superposition (combination) of several \(O(d \, n^{d-1})\) but smaller \(O(2^n)\) full grids
  - analogous combination of solutions
  - straightforward implementation: use standard discretizations or codes on standard full grids, get sparse grid behaviour
  - excellent parallelisation properties – embarrassingly parallel
  - drawback: limited access to adaptivity (so-called *dimensional* adaptivity – play with complete subspaces, not single grid points)
Contents

Why High Dimensionalities?

Sparse Grids – Fundamentals

Sparse Grids – Applications

Sparse Grids in the ExaHD Project
Sparse Grids – Applications

Numerical Quadrature
PDE – Continuum Mechanics
PDE – Finance
Classification & Regression in Data Mining
Clustering
Computational Steering
Statistics & Stochastics
Application #1: Numerical Quadrature

- **Roots (approximation, quadrature):**
  - hyperbolic crosses:
  
  \[ \Gamma(n) := \left\{ k \in \mathbb{Z}^d : \prod_{j=1}^{d} \max\{|k_j|, 1\} \leq n \right\} \]

  - Smolyak quadrature:

  \[ Q_n^{(d)} f := \left( \sum_{i=0}^{n} \left( Q_i^{(1)} - Q_{i-1}^{(1)} \right) \otimes Q_{n-i}^{(d-1)} \right) f \]

- **More recent work:**
  - Smolyak with trapezoidal rule or Clenshaw-Curtis [Frank, Heinrich, Novak]
  - explicit sparse grids (piecewise linear) [Bonk, Zenger]
  - Smolyak with incremental schemes such as Gauß-Patterson [Griebel, Gerstner]
  - direct discretization, hierarchical polynomial bases [B., Dirnstorfer]
Application #2: PDE – Continuum Mechanics

- Most work of various groups world-wide done via combination technique:
  - Fluid dynamics: Navier-Stokes
  - Finance: Black-Scholes
  - Quantum mechanics: Schrödinger

- Direct discretization:
  - finite differences [Bonn, e.g.]
  - finite elements [TUM, e.g.]
  - finite volumes [CWI, e.g.]
  - Spectral elements
Application #3: PDE – Finance

- **Definition**: An option gives the holder the right to buy (Call) or sell (Put) an underlying asset at a certain time $T$ in the future (expiration time) or before for a certain price $K$ (strike price).
- **Examples**: European, American, Asian.
- **Problem 1**: inherent non-smoothness.
- **Problem 2**: stochastic modeling of stock price development.
- **Problem 3**: we are interested in whole baskets (DAX), not single assets $\Rightarrow$ high-dim.
Application #4: Classification & Regression in Mining

- **Problem**: machine learning of a 2-class problem
- Given a **pre-classified data set**

\[ S = \{(x_i, y_i) \in [0, 1]^d \times \{-1, 1\}\}_{i=1}^M \]

of normalized data points \( x_i \) with class labels \( y_i \)
(regression: real numbers instead of discrete labels)

- **Typically**: presence of noise

  [ sampling  noise  no mere interpolation ]

- **Computational task**:
  - construct classifier/ machine learner (ML)

\[ f : [0, 1]^d \rightarrow \{-1, 1\} \]

  - provides class predictions applied to new data points
Regularization Network Approach

- Classification as scattered data approximation problem plus additional regularization terms (ill-posed problem, noise):

\[
\text{minimize } H[f] = \frac{1}{M} \sum_{i=1}^{M} \mathcal{V}(y_i, f(x_i)) + \lambda \|f\|_{K}^2
\]

- cost/error function, for example \( \mathcal{V} := (y_i - f(x_i))^2 \)
- regularization operator/stabilizer, for example \( \|f\|_{K}^2 := \|\nabla f\|_2^2 \)
- simpler, but astonishingly useful

\[
\text{minimize } H[f] = \frac{1}{M} \sum_{i=1}^{M} (y_i - f(x_i))^2 + \lambda \sum_{i=1}^{N} \alpha_i^2
\]

- regularization parameter \( \lambda \), \( f_N(x) = \sum_{i=1}^{N} \alpha_i \phi_i(x) \)
- Minimize trade-off between cost and smoothness via \( \lambda \)
- Various approaches (neural networks, support-vector-machines) can be formulated as Regularization Network Approach

- **Common classification algorithms:**
  - discretization of feature space not feasible (curse of dimension, \( O(N^d) \))
  - global ansatz functions associated to data points (\( O(M^2) \), large training data?)
- Idea: use **sparse grids**, i.e. a data-set-independent approach
Example

Learning from crash (simulation) data
Towards Higher Dimensionalities

Musk data sets, $d=166$
- separate molecules (166 attributes, mainly describing distance features of conformations)
- 10x 10-fold cross-validation
- $M=476$ (left) and smaller data set (right, with PCA leading to $d=35$)
- only two refinements before PCA, more possible after PCA
- benchmark study done 2008 to compare 45 classification algorithms; best 9 out of 38
Application #5: Computational Steering

- **Goal**: allow for interaction in high-dim parameter-dependent simulation scenarios
- **Idea**: use a sparse grid of pre-computed simulation data sets (offline/online)
- **Related**: model order reduction, surrogate models, reduced basis functions
Application #5: Computational Steering

- **Goal**: allow for interaction in high-dim parameter-dependent simulation scenarios
- **Idea**: use a sparse grid of pre-computed simulation data sets (offline/online)
- **Related**: model order reduction, surrogate models, reduced basis functions
Application #5: Computational Steering

- **Goal**: allow for interaction in high-dim parameter-dependent simulation scenarios
- **Idea**: use a sparse grid of pre-computed simulation data sets (offline/online)
- **Related**: model order reduction, surrogate models, reduced basis functions

![Image of computational steering concept]
Example: Lid-Driven Cavity
Contents

Why High Dimensionalities?

Sparse Grids – Fundamentals

Sparse Grids – Applications

Sparse Grids in the ExaHD Project
Attention! SPPEXA Commercial Begins …
Software in CSE / HPC

“Today’s CSE ecosystem is unbalanced, with a software base that is inadequate to keep pace with and support evolving HW and application needs.”

“The crisis in CSE software is multifaceted and remediation will be difficult. The crisis stems from years of inadequate investments, a lack of useful tools, a near-absence of widely accepted standards and best practices, ..., and a simple lack of perseverance by the community. This indictment is broad and deep, covering applications, programming models and tools, data analysis and visualization tools, and middleware.”

PITAC report 2005

“The field has reached a threshold at which better organization becomes crucial. New methods of verifying and validating complex codes are mandatory if CSE is to fulfil its promise ....”

“Verification, validation, and quality management, we found, are all crucial to the success of a large-scale code-writing project.”

Post and Votta, Computational Science Demands a New Paradigm, Physics Today, 2005

“In many domains software engineering quality management processes like CMMI and ISO 9000 have been successful, but apparently less so in CSE, especially in HPC-related applications.”

SPPEXA – It Has Been a Long Way…

2006 – first discussions within DFG’s Commission on IT Infrastructure (KfR)
- HPC SW runs into problems – lack of funding mechanisms; cf. international situation

2007/2008 – memorandum initiated by the geosciences
- Title *Scientific Software in the PetaFlop Era*, Roundtable discussion in Tutzing, April 2008

2010 – suggestion by German participants in the exascale initiatives
- Against the background of (1) massive investments in high-end systems world-wide and (2) massive investments in HPC software in the USA (DoE-SciDAC-1/2, NSF-OCI), e.g.

2010 – KfR takes responsibility
- Another strategic paper and a discussion with DFG’s president, Prof. M. Kleiner (Nov. 2010)
- Outcome: suggestion of a flexible, strategically initiated SPP, financed via Strategy Funds

2011 – increase of speed
- Roundtable expert meeting in MAY; DFG-internal discussions (MAY–JUL)
- Submission in AUG; international reviewing in SEP; decision in OCT; call in NOV

2012 – review of proposals
- 68 sketches in JAN, first selection in MAR leading to 24 consortia invited for full proposals
- Submission of full proposals in MAY, review workshop in JUL

2013 – launch of SPPEXA as DFG’s first-ever strategic Priority Program!
SPPEXA Characteristics

• **Strategic initiative of DFG to fund HPC SW in Germany**
  - Fundamental research
  - Establish collaborative, interdisciplinary co-design of HPC applications and HPC methods through several research consortia

• **Aims of SPPEXA’s central coordination**
  - Central SPPEXA events, establish and foster international collaboration, doctoral retreats & coding weeks
  - Support project-specific activities, dynamically distributed network funds, educational impact, gender incentives

• **SPPEXA research is …**
  - … driven by domain sciences / CSE applications
  - … powered by scientific computing & informatics / CSE methodology
  - … in parts smooth/evolutionary, in parts radical/revolutionary
SPPEXA’s 6 Research Topics

• **Computational algorithms**
  – Large-scale machines
  – Efficient w.r.t. “modern” complexity measures

• **System software**
  – Process scheduling
  – System health monitoring
  – Resilience handling

• **Software tools**
  – Compiling, running, verifying, testing, optimizing

• **Application software**
  – Key driver for exascale
  – Hardware-software co-design necessary

• **Programming**
  – Make traditional approaches exascale ready
  – New programming models

• **Data Management**
  – Process large data sets
  – Archive, make data available
SPPEXA Facts

• **13 research consortia funded**
  – Interdisciplinary research consortia
  – Involving 2-5 groups each
  – Addressing at least 2 out of the 6 SPPEXA topics
  – About 60 PIs and 60 PhD students
  – Overall budget of 3.7m € per year

• **Two three-year funding phases**

• **Launch of 1st phase in January 2013**

• **Second phase**
  – Call published last October, review in 2015
  – 01/2016–12/2018
  – **Strong internationalization component: joint call with France and Japan**
A Really Interdisciplinary Endeavor

- **Consortia cover > 15 disciplines**
  - Requires close collaboration within and among SPPEXA consortia
  - The central coordination fosters synergistic effects within SPPEXA

- **Detailed description of projects is available at**

  [http://www.sppexa.de](http://www.sppexa.de)
Advisory Board

• George Biros (U Texas)
  Institute for Computational Engineering and Sciences

• Rupak Biswas (NASA)
  Head, NASA Advanced Supercomputing (NAS) Division

• Klaus Becker (Airbus)
  Industry

• Rob Schreiber (HP Labs)
  Assistant Director, Exascale Computing Lab @HP Labs

• Craig Stewart (Indiana University)
  Executive Directory, Pervasive Technology Institute, Indiana U
Attention again: SPPEXA Commercial Ends …
Combination Technique – Path to Exascale

- Minimizes communication and synchronization
- Additional level of parallelism (sub-problems, cf. MC & DD)
  - More accurate/efficient than Monte Carlo
  - More decoupled than Domain Decomposition
- Many variants: OptiCom, iterative Opticom, ...
- Advantage #1: scalability
  - Communication-light level
- Advantage #2: resilience
  - Error compensation w/o checkpoint restart PLUS error and outlier detection
- Advantage #3: load balancing
  - Job deployment adapts to topology
  - Coarse- and fine-grained scheduling
- Topic of the SPPEXA project EXAHD (S & BN & M + UCLA, ANU)
  (demonstrator application: plasma physics)
Plasmas in Fusion Research: ITER

Idea (!): New source of CO$_2$-free energy for centuries to come

(Deuterium in a bath tub and Lithium in a used laptop battery suffice for a family over 50 years)

Goal: 500 MW of fusion power

7 countries involved
Cost: ~ 15 billion €
www.iter.org
Towards a Numerical Treatment of Plasma Fusion

Understanding microturbulence in weakly collisional plasmas is essential
→ 5D gyrokinetic Vlasov equation

\[
\dot{X} = v_\parallel \mathbf{b} + \frac{B}{B_\parallel^*} \left( \frac{v_\parallel}{B} \mathbf{B}_\perp + \frac{c}{B^2} \mathbf{E}_1 \times \mathbf{B} + \frac{\mu}{m\Omega} \mathbf{b} \times \nabla (B + \mathbf{B}_\parallel) + \frac{v_\parallel^2}{\Omega} (\nabla \times \mathbf{b})_\perp \right)
\]

\[
\dot{v}_\parallel = \frac{\dot{X}}{mv_\parallel} \cdot \left( e\mathbf{E}_1 - \mu \nabla (B + \mathbf{B}_\parallel) \right) \quad \dot{\mu} = 0
\]

Numerical solution efficiently implemented in GENE
Gyrokinetic code GENE (F. Jenko et al., 1999 – )

- Code is publicly available and widely used (http://gene.rzg.mpg.de)
- Part of the Unified European Application Benchmark Suite (PRACE)
- First PRACE Call: Ranked #1 out of 65 projects from all areas of science

On SuperMUC: up to ~16 kcores

- High-dimensional domain decomposition, pure MPI or hybrid (MPI/OpenMP), optimization of subroutines at initialization, automatic choice of optimal time step
Simulations for Actual Tokamaks with GENE

State of the art: only small section can be simulated
Some GENE Issues

- GENE runs are compute-intensive: several open algorithmic questions (solver, grid adaptation, high dimensionality)

- Large individual runs may require up to tens of millions of core-hours

- Large runs use many billion grid points and require many TB of short-term storage

- Many different HPC platforms are used in parallel

- Recently, GENE has been ported to GPGPU & Xeon Phi systems
And now ... Sparse Grids

<table>
<thead>
<tr>
<th>example</th>
<th>full grid</th>
<th>sparse grid</th>
<th>combination technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>6D, level 10</td>
<td>$&gt; 10^{18}$</td>
<td>28,779,521</td>
<td>4,096 $\times$ 249,000</td>
</tr>
<tr>
<td>10D, level 12</td>
<td>$&gt; 10^{37}$</td>
<td>159,057,502,209</td>
<td>352,705 $\times$ 80,641,000</td>
</tr>
</tbody>
</table>

High dimensionality – demand for exascale

**Second**, asynchronous level of parallelism
Two Levels of Parallelism

Grids from the combination technique run independently

+ Existing parallelism in GENE

Challenges:

- Optimize load balancing schemes (varying run times)
- New communication paradigms
- Combination recovery in case of processor failure (resilience)
- Test new architectures, such as GPU
- Investigate new numerical methods
Convergence of the Combination Technique

Small run: 10 combination grids running GENE, increasing level of refinement (s)

The errors of the individual combination grids (gray) and the combined sparse grid (black) converge to the full grid solution with GENE.

The sparse grid combination solution (black) requires half the time of the full grid solution (dashed).
Convergence of the Combination Technique

Larger parallel runs on HPC systems (Hornet, Hermit, SuperMUC). Projected runs:
level 9, ~600 partial solutions on ~92,000 cores
Load Balancing

**Challenge:**
- Anisotropy of partial grids causes load imbalance

**New solution:**
- Develop a mathematical load model to predict execution time of a partial solution [1]

\[ t(l) = t(N, s_i) = r(N)h(s_i) \]

- # unknowns
- anisotropy

\( p \): processor group à 32 cores
Global Communication

Challenge:
• Need to combine solutions after certain no. of time steps to avoid divergence of partial solutions
• Communication is increased

New solution:
• Novel communication schemes that exploit the hierarchical structure of the combination technique [2]
**Resilience**

**Challenge:**
- Processor failure leads to missing partial solutions, which causes inconsistent combinations

**New solution:**
- Substitute missing solutions by combining partial solutions already successfully calculated
- No need to restart the whole simulation using checkpoints
- Precomputation of additional grids: ~1% extra effort in 5D
Resilience

Small GENE run: 35 grids, initial value computation, simulate random faults

Dashed: CT, no faults; circles: CT with faults; squares: CT after recovery
Resilience

Small GENE run: 35 grids, initial value computation, simulate random faults

Left: CT, no faults; middle: CT with faults; right: CT after recovery
Acknowledgements

- All SCCS team members & partners in the projects
- G8 HPC Software program project Nu-FuSE (DFG)
- SPPEXA project ExaHD (DFG)
- HEPP – Helmholtz Graduate School for Plasma Physics (HGF)
- TUM-IAS – Focus Group HPC (TUM)
- IPCC (Intel)

Thanks for your attention

... and we are all Charlie!