Challenges in Advanced Computing: Multi-this and Multi-that

Hans-Joachim Bungartz

TUM, Department of Informatics, Chair of Scientific Computing
Leibniz Supercomputing Centre of the Bavarian Academy of Sciences
Munich Centre of Advanced Computing

INFOCOMP 2011
Barcelona, October 23, 2011
Part I – General Remarks

Part II – Peano: Space-Filling Curves for PDE Solvers

Part III – Sparse Grids for High-Dimensional Numerics
Part I – General Remarks

Advanced Computing

The Software Challenge

The Renaissance of Parallel Computing
Computational Science & Engineering (CSE) ➔ Key Technology for Science & Industry
High-Performance Computing (HPC) ➔ Core Enabler for CSE

Mathematical model
\[
\frac{\partial u}{\partial t} + (u \cdot \nabla) u + \frac{1}{\rho} \nabla p - \nu \Delta u = 0
\]

Discretization & solver
\[
M u_h = 0
\]
\[
A \hat{u}_h + D u_h + C(u_h) u_h - M^T p_h / \rho = 0
\]

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

Parallel implementation, HPC

Validation

Insight, Design

Software

Exploration
Atomistic Simulation of Protein Purification: State of the Art

- Small scenarios (< $10^6$ interaction sites)
- Short times (< $10^{-6}$ seconds)
- Moderately scaling codes (< $10^3$ processes)
- Simulation times of weeks
- Benefit: qualitative insights
Atomistic Simulation of Protein Purification: State of the Art ... and Vision Exascale

Polymer-modified protein in aqueous electrolyte solution

- Small scenarios (< $10^6$ interaction sites)
- Large scenarios (> $10^{10}$ interaction sites)
- Short times (< $10^{-6}$ seconds)
- Long times (> $10^{-3}$ seconds)
- Moderately scaling codes (< $10^3$ processes)
- Massively parallel codes (> $10^6$ processes)
- Simulation times of weeks
- Response times of minutes
- Benefit: qualitative insights
- Quantitative insights and predictions

Solvent-hydrogel interactions
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 data / images / numbers …
  … to information / insight
- From counting operations …
  … to energy awareness
- From sequential algorithm design …
  … to massive parallelism
- 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 embedding & integration
- From flat algorithms & data …
  … to hierarchy

Insight through computation & visualisation
CSE/HPC Challenges

From parameter assumptions …
… to identification & estimation

From forward problems …
… to inverse problems

From single-physics problems …
… to coupled scenarios

From simulation …
… to optimisation

From one (spatial/temporal) scale …
… to cascades of scales

From data / images / numbers …
… to information / insight

From sequential algorithm design …
… to massive parallelism

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 flat algorithms & data …
… to hierarchy

From island fun …
… to embedding & integration

Multi-disciplinary

From deterministic models …
… to random & uncertainty

From qualitative descriptions …
… to quantitative prediction

From counting operations …
… to energy awareness

From simulation …
… to optimisation
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 data / images / numbers …
… to information / insight

From counting operations …
… to energy awareness

From sequential algorithm design …
… to massive parallelism

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 embedding & integration

From flat algorithms & data …
… to hierarchy

Multi-physics
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 data / images / numbers ... … to information / insight
- From counter operations ... … to energy awareness
- From sequential algorithm design ... … to massive parallelism
- 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 flat algorithms & data ... … to hierarchy
- From island fun ... … to embedding & integration
- From qualitative descriptions ... … to quantitative prediction
- From simulation ... … to optimisation
- From simulation ... … to optimisation
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 data / images / numbers ... ... to information / insight
- From counting operations ... ... to energy awareness
- From sequential algorithm design ... ... to massive parallelism
- 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 island fun ... ... to complex workflows
- From flat algorithms & data ... ... to hierarchy
- From sequential algorithm design ... ... to massive parallelism
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 data / images / numbers …
… to information / insight

From counting operations …
… to energy awareness

From sequential algorithm design …
… to massive parallelism

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 embedding & integration

From flat algorithms & data …
… to hierarchy

From qualitative descriptions …
… to quantitative prediction

From simulation …
… to optimisation

From one (spatial/temporal) scale …
… to cascades of scales

CSE/HPC Challenges

Multi-modal
CSE/HPC Challenges

- From parameter assumptions ... to identification & estimation
- From forward problems ... to inverse problems
- From deterministic models ... to random & uncertainty
- From single-physics problems ... to coupled scenarios
- From one (spatial/temporal) scale ... to cascades of scales
- 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 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 embedding & integration
- From flat algorithms & data ... to hierarchy

Multi-core
CSE/HPC Challenges

From parameter assumptions ...
... to identification & estimation

From forward problems ...
... to inverse problems

From simulation ...
... to optimisation

From qualitative descriptions ...
... to quantitative prediction

From data / images / numbers ...
... to information / insight

From counting operations ...
... to energy awareness

From sequential algorithm design ...
... to massive parallelism

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 single-physics problems ...
... to coupled scenarios

From flat algorithms & data ...
... to hierarchy

From island fun ...
... to embedding & integration

From one (spatial/temporal) scale ...
... to cascades of scales

From flat algorithms & data ...
... to hierarchy

Multi-dimensional
Beyond Moore: Enablers of Computing

In general: increasing relevance of informatics
[ languages, compilers, tools, (par)prog paradigms, data exploration, visualization, algorithms and data structures (hierarchy & recursion), hardware-aware algorithmics, HW-SW-co-design, software engineering ... ]

In particular: increasing relevance of software issues
- Classical software engineering
  - specification
  - modular design
  - configuration management
  - documentation and sustainability
  - ...
- Algorithmic verification and validation
- Adaptive software lifecycle management
- Handling PSE & simulation workflows
- Systematic testing
- Languages, compilers, runtime systems
- User interfaces
- Tool development & professional tool support
- Performance tuning (esp. parallel)

Computational Algorithms
- Computational statistics
- Differential equations & multi-physics problems

Parallel Architectures
- {Homo/hetero}geneous multicore architectures
- Parallelisation support & programming models

Data Engineering / Exploration
- Database technology & data mining
- Visual analytics
Software in CSE/HPC – State of the Art

“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.”

Studies on CSE-Related SW Development in the US (DoD, DoE – ASCI and successors; study from 2004)

- Development team size (median): 6 increasing
- Code size (median): 300 k LoC increasing
- Number of users (median): 25 stable
- Code project age (median): 17.5 years increasing
- Presence of Fortran: 58% (24% F77, 34% F90/95) decreasing

Source: Kendall, Post, et al.
Some Observations on SW Development in CSE

Roadblocks:

- Most developers are domain scientists and engineers, not computer scientists
- Typical priority: science >> code performance >> software quality
- Intellectual level assigned: models >> algorithms >> programs
- No “software engineering mainstreaming”: design, process models, workflow models, …
- No “team understanding”: co-operative work, trans-disciplinary, project management, …
- Instead, still the lonely heroes with their heroic codes (and sometimes accumulated heroism)
- No systematic testing culture
- No formal support – verification
- No best practices
- Many groups working on PSE – in general with limited success

“Software development is the principal bottleneck in CSE” (R. Kendall)
Energy Density – the Fundamental Problem

Source: Fred Pollack, Intel. New Microprocessor Challenges in the Coming Generations of CMOS Technologies, Micro32
We’re no longer getting faster – we’re getting more!

Traditional Sources of Performance Improvement are Flat-Lining

- **New Constraints**
  - 15 years of exponential clock rate growth has ended

- **But Moore’s Law continues!**
  - How do we use all of those transistors to keep performance increasing at historical rates?
  - Industry Response: #cores per chip doubles every 18 months *instead* of clock frequency!

**Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith**
Increase of Numbers of Cores

Total # of Cores in Top15

Source: W. Nagel
Sandy Bridge MatMult – Best Ingredients

Source: W. Nagel
Sandy Bridge MatMult – Straightforward (C, gcc, ...)
Why All New?

… with 4 strong jet engines …

Large Scale Simulation Software

Would you prefer to equip an A 380 …

… or rather with 100,000 hair dryers??

Moderately Parallel Computing

Massively Parallel MultiCore Systems

Source: U. Rüde
Part II – Peano: Space-Filling Curves for PDE Solvers

The Scope of Space-Filling Curves

The Peano Project

Applications
[ not discussed here ]
Algorithmic Challenges – and Changes

- **Tackling the “memory wall”:**
  - cache-awareness via sophisticated traversal strategies
  - cache-oblivious vs. cache-conscious

- **Tackling on-chip parallelism (multi-core):**
  - multi-threading, fine-grain parallelism
  - no more “sequential kernel”
  - non-standard hardware: accelerators, such as GPGPU, (Cell), FPGA (?)

- **Tackling scalability: hybrid concepts, sophisticated & cheap load balancing**
  - heterogeneous scenarios (non-standard geometry, multi-level schemes, …) require dynamic load balancing
  - intra- and inter-system (from hybrid systems to the Grid)

→ **A promising paradigm: space-filling curves**
  [ SFC: continuous and surjective mapping from unit interval onto unit square/cube ]
  - **Lebesgue:** the classical one (Morton, Z, octree)
  - **Hilbert:** the most famous one
  - **Peano:** our favourite for Cartesian grids
  - **Sierpinski:** the newcomer for triangles & Co.

---

**Annual gain in last years: (avg.)**
- CPU performance: 60%
- memory bandwidth: 23%
- memory latency: 5%
**What is a Space-Filling Curve?**

- **curve**: continuous mapping from the unit interval to a 2D/3D/… reference domain (unit square, unit cube, potato, …)
- **space-filling**: mapping is surjective \([\text{rather strange that this works} \ldots \text{bijective won’t}]\)
- **purpose**: serialize high-dim data \([\text{facilitates data organization, access, queries, partitioning,} \ldots]\)
- **construction**: recursive process, following *inclusion* and *neighbourhood*
- **example**: Hilbert curve
Overview of Space-Filling Curves *(there are more!!)*

Lebesgue
- organizing simulation tasks

Hilbert
- dynamic load balancing

Peano
- matrix-matrix multiplication

Sierpinski
- Tsunami simulation
TifaMMy – cache-efficient matrix multiplication

- Peano-based traversal with high locality (dense or sparse matrices)
- block-structured data structure and algorithm
- parallel @ multicore: HW-conscious kernel, OpenMP
- parallel @ clusters: distributed caches, MPI
- application: quantum control (states via matrices)

LATEST NEWS from ISC ‘11, Hamburg, June 20, 2011:
Intel announces TifaMMy among the very few first and best applications world-wide on Intel’s completely new Many Integrated Core (MIC) Architecture “Knights Ferry”
SFC #1b – Peano: Iterative Solvers for PDE

PDE solvers:
- high cache efficiency (>99.9% L2), low memory requirement ($10^9$ d.o.f. on 1 GB RAM!)
- easy access to parallelisation and dynamic load distribution
- no restrictions w.r.t. adaptivity
- full multigrid potential

Organisation:
- Recursive partitioning

Ordering:
- based on space-filling (Peano) curves
- for geometry representation, iterations, and parallelisation
SFC #2 – Sierpinski: Tsunami Simulations

**Sierpinski space-filling curves**

- FEM with strong adaptive refinement & coarsening
- structured, but triangular / tetrahedral
- high locality and HW-/cache-efficiency
- Sierpinski-based traversal, newest vertex bisection
- discontinuous Galerkin discretization
- **application: Tsunami simulation (shallow water eqs.)**
Part II – Peano: Space-Filling Curves for PDE Solvers

The Scope of Space-Filling Curves

The Peano Project

Applications
[ not discussed here ]
Yet Another Framework – Ideas & Goals

general PDE framework, with focus on CFD/FSI

discretization issue: FE – strictly conservative

grid issue: Cartesian – at least logically

dynamics issue: straightforward generation & adaptation

solver issue: inherent multi-level solvers and parallelisation

dimensionality issue: general dimensionality

high efficiency (memory footprint w.r.t. size & access, parallel, …)
Peano Concept #1: Grid Organisation via Spacetrees

Cartesian grid cells
- squares/cubes/hypercubes

recursive refinement
- data structure: tree
- Peano: tri-partitioning
Peano Concept #2: Cell-Oriented Operator Evaluation

\[(\Delta_h u_h)_{i,j} = \frac{u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1}}{h^2}\]
Peano Concept #2: Cell-Oriented Operator Evaluation

\[
(\Delta_h u_h)_{i,j} = \frac{1}{2} u_{i-1,j} + \frac{1}{2} u_{i,j-1} - u_{i,j}
\]

\[
\begin{pmatrix}
\frac{1}{2} & -1 \\
\frac{1}{2} & \frac{1}{2}
\end{pmatrix}
\]
Concept #2: Cell-Oriented Operator Evaluation

\[
(\Delta_h u_h)_{i,j} = \frac{1}{2} u_{i+1,j} + \frac{1}{2} u_{i,j-1} - u_{i,j}
\]

\[
\begin{pmatrix}
-1 & 1/2 \\
1/2 & 1
\end{pmatrix}
\]
Peano Concept #3: Grid Traversal via Stacks

traversal along a Peano curve ➔ also adaptive

cell-oriented operator evaluation ➔ need for intermediate / non-persistent storage

stacks as non-persistent data structure

adaptivity & generating systems ➔ hierarchy

high spatial and time locality of data access
Peano Features

Feature #1: Multilevel inside
[exploits hierarchy – matrix-free concept, ML-preconditioned cg, multigrid, …]

Feature #2: Load balancing inside
[so far: works nice for moderate proc numbers up to O(10000)]

Feature #3: Efficient memory use and access
[throughout the hierarchy disk ↔ main memory ↔ cache; L2 hit rates > 99.9%]

- dehierarchisation
- compute residual
- smooth
- restrict residual

- fine grid data
- coarse grid data
FSI Fluid Solver Benchmark, Re = 200
Part III – Sparse Grids for High-Dimensional Numerics

Sparse Grids: Main Properties

First Applications: Quadrature and PDE (Flows, Finance)
[ not discussed here ]

Application to Classification & more
A Hot Topic: High-Dimensional Numerics

- **High**: not 2, not 3, not 3 plus time, but 10 … 100 … 1000

- **Where?**
  - quantum mechanics
  - finance
  - 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 simplest 1-D discretisation … ... and in 100-D
      \[
      1^{100} = 1
      \]
    - The second simplest 1-D discretisation … ... and in 100-D
      \[
      2^{100} \approx 10^{30}
      \]
Sparse Grids in a Nutshell

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

$$\max_{u \in X(\Omega): |u| = 1} \| u - u_{V(\text{opt})} \|$$

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

- result: sparse grids
  [Zenger et al., 1990]
**Sparse Grids in a Nutshell (cont’d)**

- **Appearance:**

- **Cost** (number of grid points) vs. **benefit** (contribution to interpolant):
  (finest mesh width $h_n=2^{-n}$, hierarchical bases of piecewise degree $p$)

<table>
<thead>
<tr>
<th></th>
<th>sparse</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td># grid points</td>
<td>$O(h_n^{-1} n^{d-1})$</td>
<td>$O(h_n^{-d})$</td>
</tr>
<tr>
<td>error (max, $L_2$)</td>
<td>$O(h_n^{p+1} n^{d-1})$</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>

- Extensions: straightforward access to **adaptive refinement**, generalisation to **piecewise polynomial** hierarchical bases, energy-optimal sparse grids
Part III – Sparse Grids for High-Dimensional Numerics

Sparse Grids: Main Properties

First Applications: Quadrature and PDE (Flows, Finance)
[ not discussed here ]

Application to Classification & more
Classification & Regression in Data 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 \(\rightarrow\) noise \(\rightarrow\) 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 \( \|f\|_K^2 := \|\nabla f\|_{L^2}^2 \)
  - 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
Examples

Ripley data set (\(d=2\))
- training (250) and test data (1000), 8% noise (i.e. 92% maximum accuracy)
- most refinement in critical region, accuracy of 91.5% on test data
- after only 8 refinement steps, overfitting takes over and accuracy deteriorates

Bupa liver data set (345 patients for liver illness, \(d=6\), \(\lambda=0.01\))

<table>
<thead>
<tr>
<th>adapt. sg (\lambda = 0.01)</th>
<th>comb. techn. lin. anisotrop.</th>
<th>SVM linear</th>
<th>SVM non-linear acc. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td># refinements</td>
<td># grid</td>
<td>acc. [%]</td>
<td>acc. [%]</td>
</tr>
<tr>
<td>7</td>
<td>403</td>
<td>72.22</td>
<td>73.9</td>
</tr>
<tr>
<td>13</td>
<td>1091</td>
<td>74.61</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1371</td>
<td>76.30</td>
<td></td>
</tr>
</tbody>
</table>
Towards Higher Dimensionalities

Optical recognition of hand-written digits, \(d=64\)
- 8x8 pattern of grey-scale values, 3,823 training and 1,797 test data
- one classifier for each class (ten successive 2-class problems)
- result: 97.74%
  - compared to k-NN: 98.00%
  - compared to RBF-DDA networks: 97.45%
  - compared to Tree-SVM: 97.27%
  - compared to MLP: 89.05%
- grid points on each level for one of the classifiers (no boundary points)
- adaptive refinement crucial

<table>
<thead>
<tr>
<th>Level</th>
<th>Regular Sparse Grid</th>
<th>Adaptive Sparse Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>128</td>
</tr>
<tr>
<td>3</td>
<td>8,324</td>
<td>993</td>
</tr>
<tr>
<td>4</td>
<td>366,080</td>
<td>510</td>
</tr>
<tr>
<td>5</td>
<td>12,263,680</td>
<td>128</td>
</tr>
</tbody>
</table>
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
Applications of Regression

• Cosmological redshift estimation
  [use cheap photometric measurements to predict expensive, but accurate spectroscopic ones; 5D; large data set: 430 k data points, 60 k used for testing]

• Option pricing

• Cosmological parameter sampling
  [cosmic microwave background radiation emitted just after Big Bang; 6-9 parameters; model available; compare results with observations; determine the best/most probable set of parameters (inverse problem); so far stochastic approaches only]

• Parameter scans in plasma physics
  [task arising in gyrokinetics, quasi-linear model; part of optimization problem; aim: interpolate]
Lessons Learned

- Efficient parallelization possible
Part I – General Remarks

Part II – Peano: Space-Filling Curves for PDE Solvers

Part III – Sparse Grids for High-Dimensional Numerics
Thanks to all those who really did the work

Thanks for your attention!