High-performance and automatic computing for simulation science

Paolo Bientinesi

AICES, RWTH Aachen
pauldj@aices.rwth-aachen.de

November 8th, 2013
Kickoff workshop
Simulation Lab "ab initio Methods in Chemistry and Physics"
Jülich Supercomputing Center, Forschungszentrum Jülich

Overview

- **High-performance computing**
- **Automatic computing**
- **Applications / Simulation science**

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  $\rightarrow$ time-to-solution, efficiency, scalability, ...

- **Automatic computing**

- **Applications / Simulation science**

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  $\rightarrow$ time-to-solution, efficiency, scalability, ...

- **Automatic computing**
  optimization, search, but also derivation & deduction
  $\rightarrow$ range of algorithms, productivity

- **Applications / Simulation science**

Overview

- **High-performance computing**
  numerical computations, parallel architectures
  → time-to-solution, efficiency, scalability, . . .

- **Automatic computing**
  optimization, search, but also derivation & deduction
  → range of algorithms, productivity

- **Applications / Simulation science**
  → application-specific properties & needs, large scale, full code

Collaborative approach

Opportunities

- **High-performance computing**

  Disconnect with applications ⇒ (too) general assumptions
  Often marginal gains (vs. algorithmic improvements)
  Cyclic, long development; focus on asymptotic behaviour

- **Automatic computing**

  Automation = "search"

- **Applications / Simulation Science**

  Disconnect with state-of-the-art algorithms
  Legacy, non-extensible code
  Culture of "wait-for-faster-computer"

Simulation codes

- `GenABEL` Prof. Y. Aulchenko (SD-RAS) `www.genabel.org`
  Statistical genomics

  New package. Goal: new analysis, “mixed models”; performance

- `FLEUR` Prof. S. Blügel (FZ-Jülich) `www.flapw.de`
  Investigation of structural, electronic and magnetic properties of periodic systems

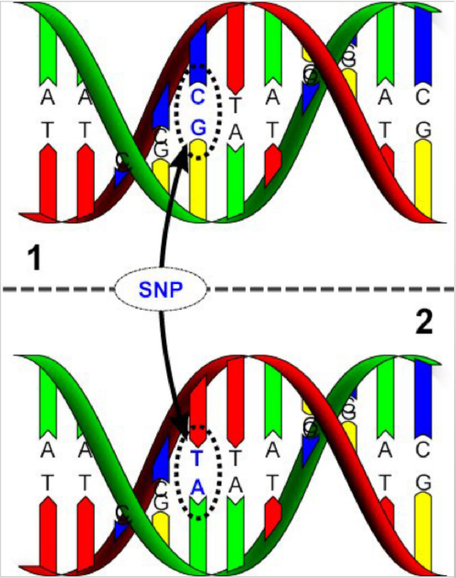
  Existing code. Goal: scalability & portability (of performance)

- `LAMMPS` Prof. A. Ismail (RWTH) `lammps.sandia.gov`
  Molecular Dynamics Simulator

  New kernel. Goal: new algorithm for $1/r^6$ potential; linear complexity

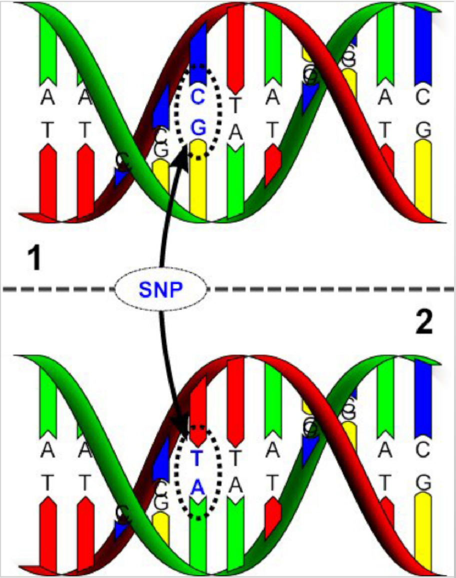
1. HPAC: Introduction
2. Application: Genome-Wide Association Studies
3. Application: Density Functional Theory
4. Conclusions

Genome-Wide Association Studies



Source: David Hall

Genome-Wide Association Studies



Source: David Hall

Y. Aulchenko

Genome-Wide Association Studies

<u>Yurii</u> <u>Paolo</u>

Genome-Wide Association Studies

Yurii	Paolo
“Mixed models”	???

Genome-Wide Association Studies

Yurii	Paolo
"Mixed models"	???
Linear regression with non-independent outcomes	???

Genome-Wide Association Studies

Yurii	Paolo
“Mixed models”	???
Linear regression with non-independent outcomes	???
Generalized least-square problems	. . .

Genome-Wide Association Studies

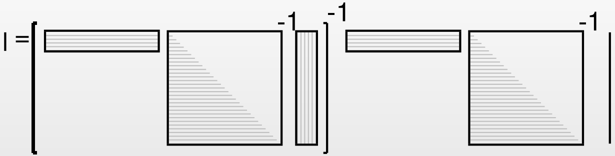
Yurii	Paolo
"Mixed models"	???
Linear regression with non-independent outcomes	???
Generalized least-square problems	...

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

- Inputs: $M \in \mathbb{R}^{n \times n}, \quad X \in \mathbb{R}^{n \times p}, \quad y \in \mathbb{R}^n$
- Output: $b \in \mathbb{R}^p$

$\star$**To be repeated millions of times**$\star$

Mixed models $b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$



Genome-wide association analysis

- y: **phenotype** (outcome; vector of observations)

  E.g.: height, blood pressure for a set of people

- X: **genome measurements** and covariates (design matrix; predictors)

  E.g.: sex and age over height

- M: **dependencies** between observations

  E.g.: tall parents have tall children

- b: **relation** between a variation in the outcome (y) and a variation in the genome sequence (X)

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

$$\Downarrow$$

for $i = 1, \ldots, m$

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

Problem definition (1)

$$b := \left(X^T M^{-1} X\right)^{-1} X^T M^{-1} y$$

"to be repeated millions of times"

$$\Downarrow$$

for $i = 1, \ldots, m$

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

$$\Downarrow$$

Problem size

$M_i \in \mathbb{R}^{n \times n}$	$1000 \leq n \leq 20k$+	7.5MBs – 3GBs
$X_i \in \mathbb{R}^{n \times p}$	$3 \leq p \leq 20$	30 – 625KBs
$y_j \in \mathbb{R}^{n}$		8 – 780KBs
$b_i \in \mathbb{R}^{p}$		24 – 160 Bytes
Total	$10^6 \leq m \leq 10^8$	7.5 – 3000 TBs

Problem definition (2)

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

Problem definition (2)

$$b_i := \left(X_i^T M_i^{-1} X_i\right)^{-1} X_i^T M_i^{-1} y_i$$

$$\Downarrow$$

for $i = 1, \ldots, m$

$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y,$

and $X_i = [X_L | X_{Ri}]$

$$\Downarrow$$

Problem size

$M \in \mathbb{R}^{n \times n}$	$1000 \leq n \leq 100k$	7.5MBs – 74.5GBs
$X_{Ri} \in \mathbb{R}^n$		8 – 780KBs
$b_i \in \mathbb{R}^p$		24 – 160 Bytes
Total	$10^6 \leq m \leq 10^8$	74GBs – 7 TBs

Problem definition (3)

$$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y$$

Problem definition (3)

$$b_i := \left(X_i^T M^{-1} X_i\right)^{-1} X_i^T M^{-1} y$$

$$\Downarrow$$

for $i = 1, \ldots, m$

for $j = 1, \ldots, t$

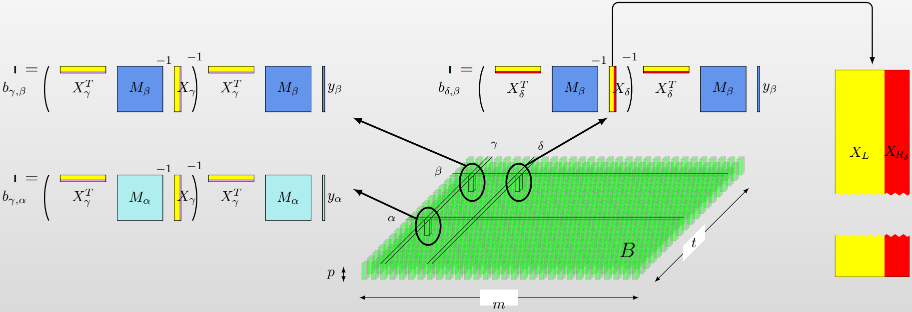
$b_{ij} := \left(X_i^T M_j^{-1} X_i\right)^{-1} X_i^T M_j^{-1} y_j,$

and $X_i = [X_L | X_{Ri}],$

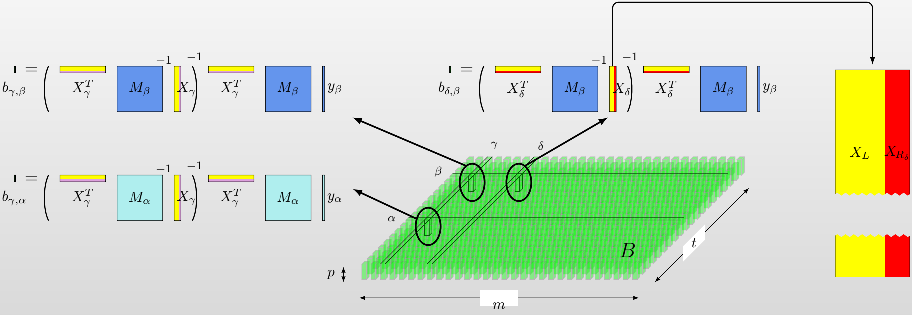
and $M_j = \sigma_j(\Phi + h_j I).$

Moreover, either $t = 1$ or $t \leq 10^5$

GWAS: complete problem definition



GWAS: complete problem definition



Problem size

$M \in \mathbb{R}^{n\times n}$	$1000 \leq n \leq 100k$	7.5MBs – 74.5GBs
$X_{Ri}, y_j \in \mathbb{R}^n$		8 – 780KBs
$b_{ij} \in \mathbb{R}^p$	$3 \leq p \leq 20$	24 – 160 Bytes
Total	$m \leq 10^8,\ t \leq 10^5$	1.5 – 100s TBs

Algorithms generated

Algorithm 1	Algorithm 2	…	Algorithm 20	…
$LL^T = M$	$LL^T = M$		$ZWZ^T = \Phi$	
$X := L^{-1}X$	$X := L^{-1}X$		$D := (hW + (1-h)I)^{-1}$	
$S := X^T X$	$QR := X$		$KK^T = D$	
$GG^T = S$	$y := L^{-1}y$		$X := Z^T X$	
$y := L^{-1}y$	$b := Q^T y$		$X := K^T X$	
$b := X^T y$	$b := R^{-1}b$		$QR := X$	
$b := G^{-1}b$			$y := L^{-1}y$	
$b := G^{-T}b$			$b := Q^T y$	
			$b := R^{-1}b$	

Many algorithms! Predictions?

Flop count – rough estimate

	Alg. 1	Alg. 2	Alg. 20
Single instance ($t = 1$)	$\mathbf{O(n^3)}$	$O(n^3)$	$O(n^3)$
2D sequence ($t \gg 1$)	$O(tn^3 + mtn^2)$	$O(tn^3 + mtn^2)$	$\mathbf{O(n^3 + mtn)}$

Analytic models

Model-based prediction

Algorithm → implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

Algorithm $\to$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

Algorithm $\rightarrow$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

  Does M fit in GPU-memory?

  - Yes $\Rightarrow$ accelerator
    streaming HD$\leftrightarrow$CPU$\leftrightarrow$GPU, triple+double buffering, CPU+GPU implementation

Algorithm $\to$ implementations

operands

X	input	100s GBs – 2 TBs	streaming from disk
y	input	1 – 10 GBs	streaming from disk
M	input	MBs – 80 GBs	read once
b	output	100s MBs or 10s TBs	streaming to disk

Does M fit in memory?

- YES $\Rightarrow$ single node + multithreading
  streaming HD$\leftrightarrow$CPU, double buffering, in-core implementation

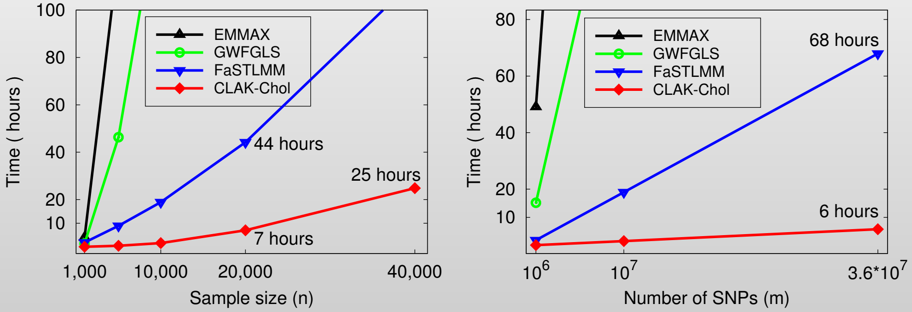
  Does M fit in GPU-memory?

  - Yes $\Rightarrow$ accelerator
    streaming HD$\leftrightarrow$CPU$\leftrightarrow$GPU, triple+double buffering, CPU+GPU implementation

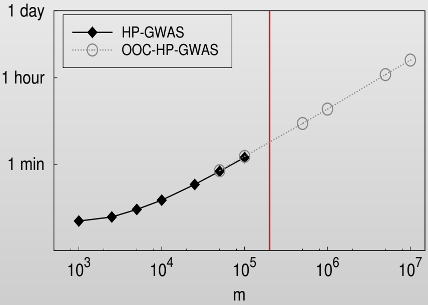
- NO $\Rightarrow$ distributed memory + MPI
  partitioning + streaming HD$\leftrightarrow$CPUs, double buffering, data distribution

Results $t = 1$

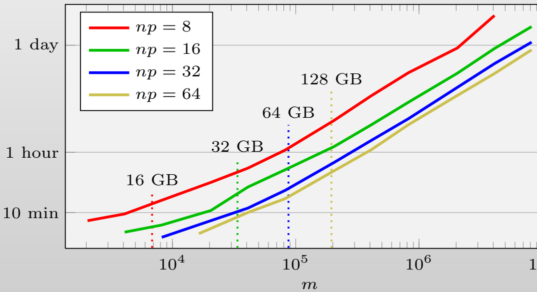
Single-trait analysis



Single node

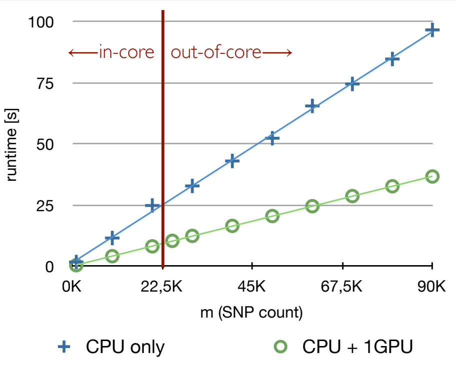


MPI

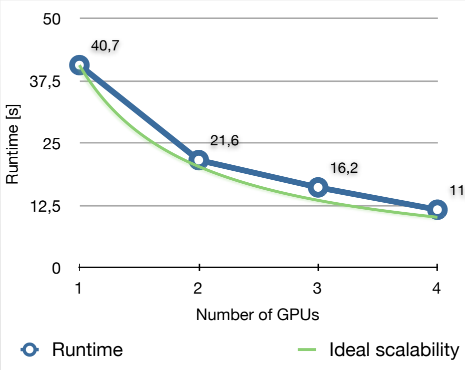


Single-trait analysis

1 GPU

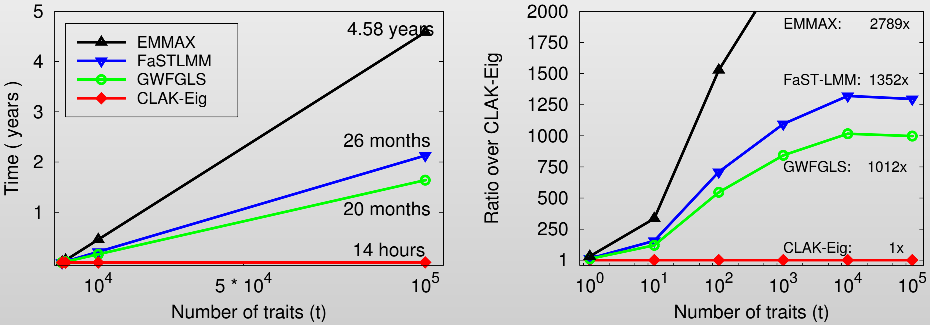


Scalability



Results $t \gg 1$ – full problem

Multi-trait analysis, “OMICS”-data



Comments & lessons

HPC's perspective

- Computation time

  $\rightarrow$ in-core efficiency
  $\rightarrow$ how to sustain efficiency?

Comments & lessons

HPC's perspective

- Computation time

  $\rightarrow$ in-core efficiency
  $\rightarrow$ how to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

Comments & lessons

HPC's perspective

- Computation time

  $\rightarrow$ in-core efficiency
  $\rightarrow$ how to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

- Language barrier!
- Problem-specific properties:
  Expose them, exploit them!

Comments & lessons

HPC's perspective

- Computation time

  $\rightarrow$ in-core efficiency
  $\rightarrow$ how to sustain efficiency?

App's perspective

- Many data formats
- Missing data, bogus data
- Output data. Post-processing?
- New features

- Language barrier!
- Problem-specific properties:
  Expose them, exploit them!
- HUGE gap:
  algorithm $\leftrightarrow$ optimized implementation
  (data management, parallelism)
- Development cycle: several months!

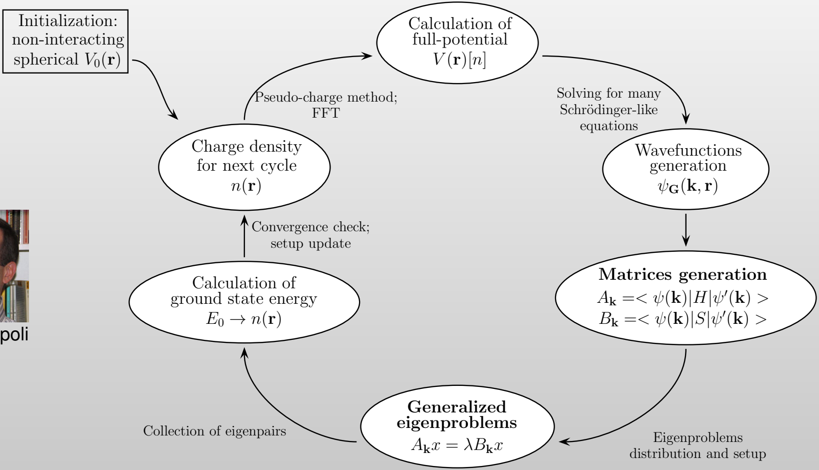
FLEUR: Density Functional Theory

Schrödinger equation $\rightarrow$ self-consistent cycle

FLEUR: Density Functional Theory

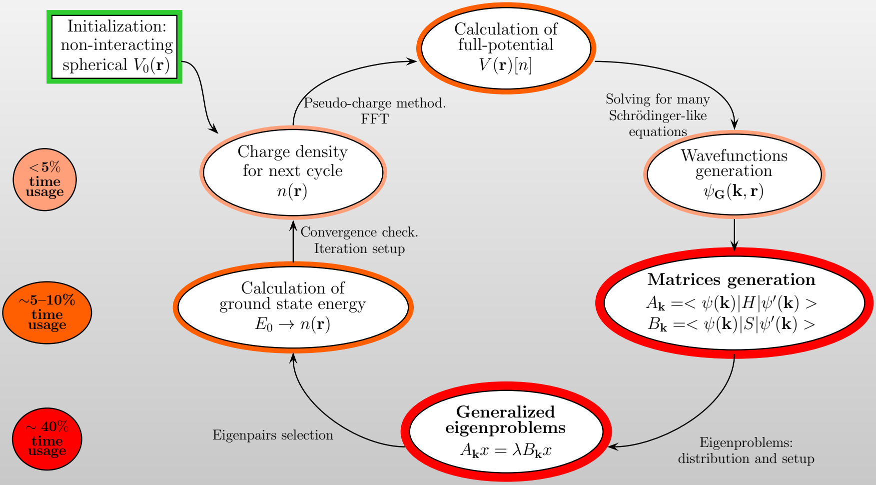
Schrödinger equation → self-consistent cycle

E. Di Napoli



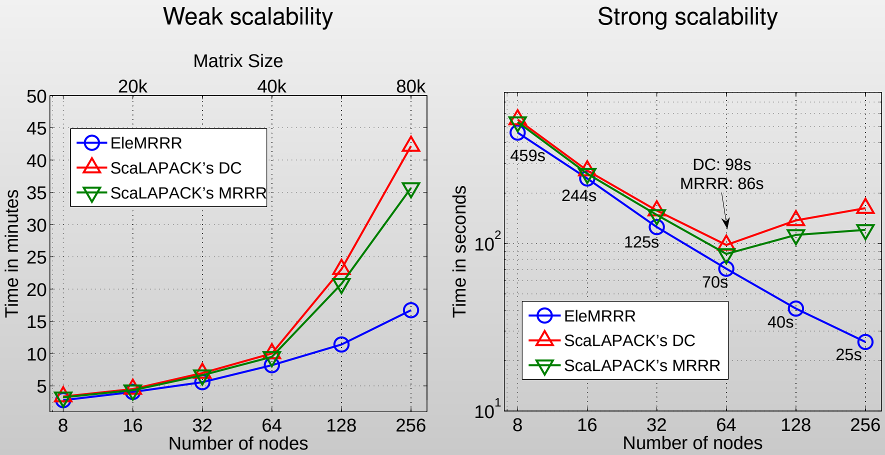
FLEUR: Density Functional Theory

Sequences of problems, accuracy vs. speed, evolution?



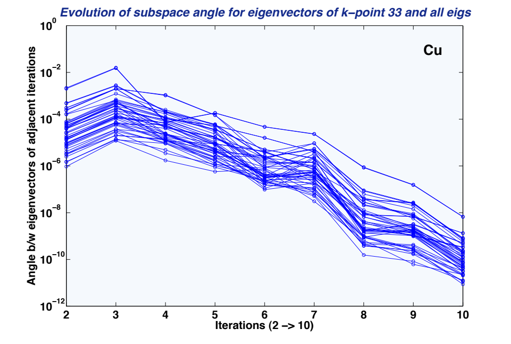
EleMRRR: $Ax = \lambda Bx$

Distributed-memory & hybrid architectures

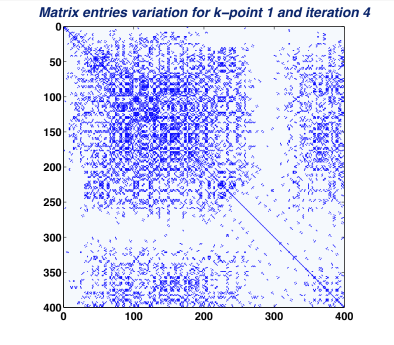


Cycle evolution

Eigenvectors



Hamiltonian



Conclusions

- Simulation science $\Rightarrow$ new computer science

Conclusions

- Simulation science $\Rightarrow$ new computer science

Opportunities for a paradigm shift

- From problems in isolations
  $\Rightarrow$ sequences of operations
- From black box libraries
  $\Rightarrow$ problem-specific knowledge
- From isolated algorithms
  $\Rightarrow$ integration into simulation code

Conclusions

- Simulation science $\Rightarrow$ new computer science

Opportunities for a paradigm shift

- From problems in isolations
  $\Rightarrow$ sequences of operations
- From black box libraries
  $\Rightarrow$ problem-specific knowledge
- From isolated algorithms
  $\Rightarrow$ integration into simulation code

- Morale:
  Instead of one semi-expert in different fields
  $\Rightarrow$ different experts at one table