Karl Meerbergen:
Matrix algorithms for parametric dynamical systems

Applications

- Structures and vibrations, pharma-industry
  - Uncertainty quantification
  - Design optimization, PDE constrained optimization
- Bifurcation analysis
- (Distance problems)

Karl Meerbergen:
Matrix algorithms for parametric dynamical systems

Applications

- Structures and vibrations, pharma-industry
  - Uncertainty quantification
  - Design optimization, PDE constrained optimization
- Bifurcation analysis
- (Distance problems)

Linear algebra problems

- Eigenvalue problems: linear, non-linear
- Model reduction: linear, non-linear
- Linear systems: preconditioning
- Tensors

Karl Meerbergen:
Matrix algorithms for parametric dynamical systems

Applications

- Structures and vibrations, pharma-industry
  - Uncertainty quantification
  - Design optimization, PDE constrained optimization
- Bifurcation analysis
- (Distance problems)

Linear algebra problems

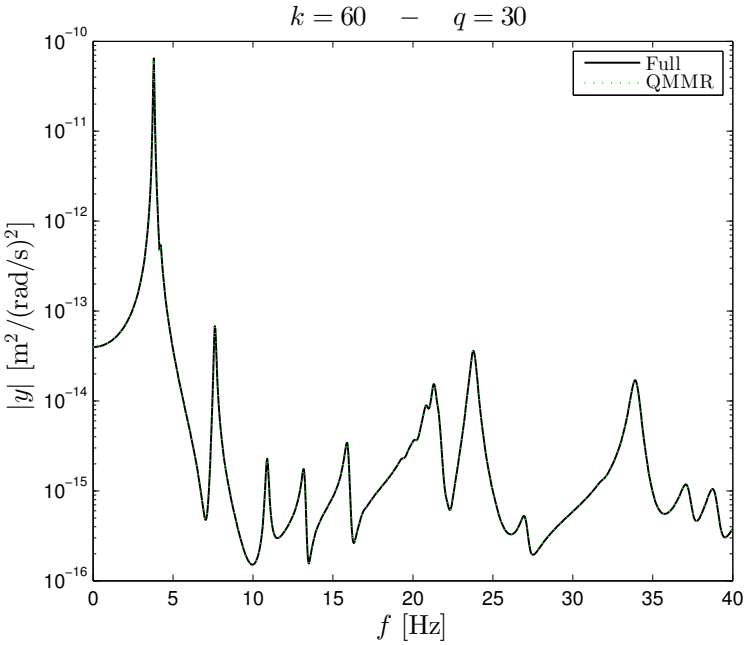
- Eigenvalue problems: linear, non-linear
- Model reduction: linear, non-linear
- Linear systems: preconditioning
- Tensors

Exascale computing

- Flanders ExaScience Lab (Intel) founded in June 2010.
- Co-design lab for future hardware for HPC

Parametric matrices in uncertainty analysis/optimization





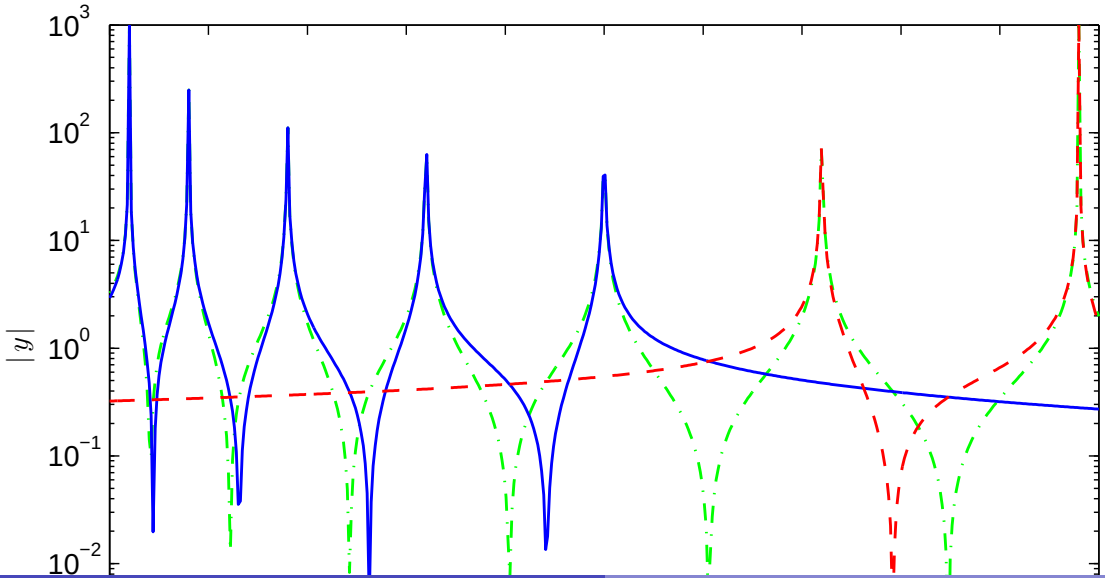
- Large linear systems depending on parameters (frequency, physical parameters)
  - Preconditioning for indefinite systems (= hard)
  - Many linear systems to solve $\longrightarrow$ parametric model reduction
- Eigenvalue problem depending on (physical) parameters
  - Eigenvalues play an important role in the dynamics of such problem
  - Parametric modal truncation: Maryam Saadvandi, Wim Desmet (mechanical engineering)

Recycling/ Quadratic output

- SISO linear system

$$\begin{aligned}(K - \omega^2 M)x &= f \\ y &= x^* S x\end{aligned}$$

  with S symmetric. (Joint with Van Beeumen, Van Nimmen, Lombaert)
- Method:
  - Krylov space with $f \Rightarrow V_k = [v_1, \ldots, v_k]$
  - Extract Ritz vectors (M. & Bai)
  - Use combination of modal superposition (recycle Ritz vectors) with block Krylov for SV_k



Dominant poles for nonlinear frequency dependent models

- SISO Nonlinear system

$$\begin{aligned} A(s)x &= fu(s) \\ y &= d^T x \end{aligned}$$

- For many problems, y can be developed in a modal expansion

$$y = \sum_{j=1}^{\infty} \frac{R_j}{s - \lambda_j}$$

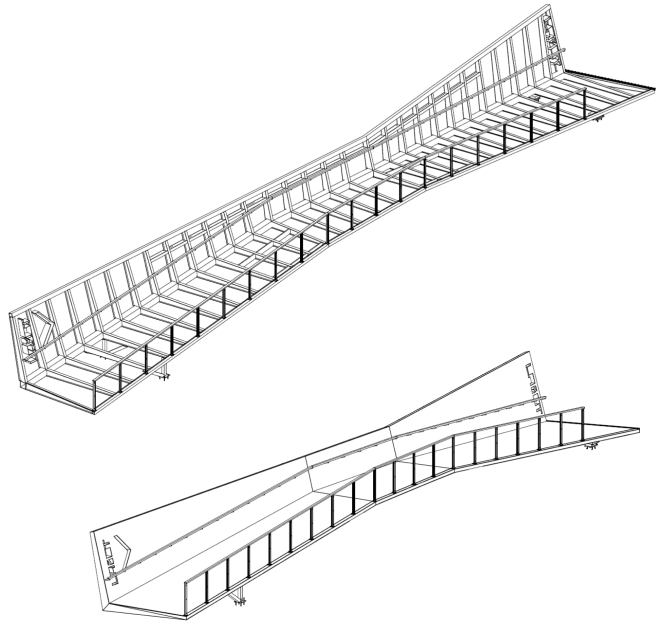
  where λ_j is a root of $\det(A(s)) = 0$.
- Dominant pole algorithm makes a reduced model based on the dominant terms:

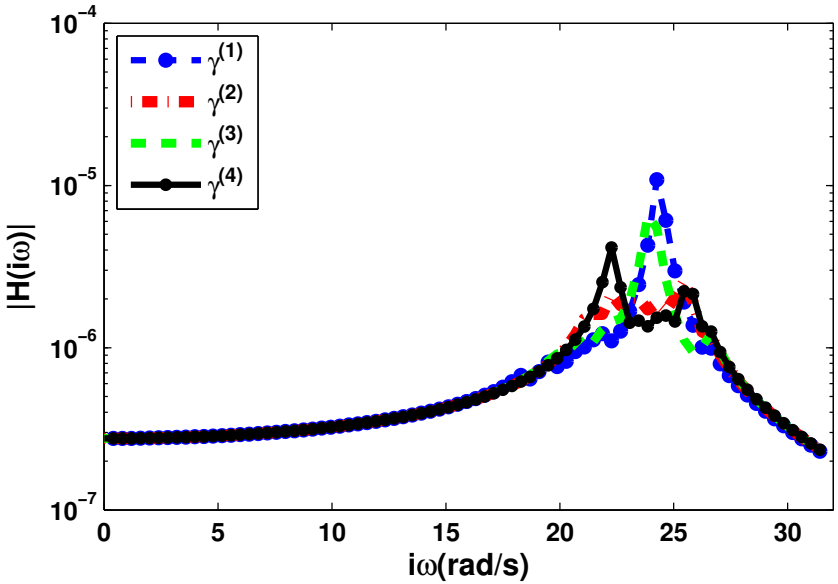
$$y \approx \sum_{j=1}^{k} \frac{R_j}{s - \lambda_j}$$

- Joint work with Maryam Saadvandi and Wim Desmet (Mechanical engineering): compute dominant poles for parametric systems

Dominant poles for parametric equations

$$\begin{cases} \left((1+0.02i)K_0 + (k_1 + i\omega c_1)K_1 - \omega^2 M\right) x &= \\ y &= \end{cases}$$



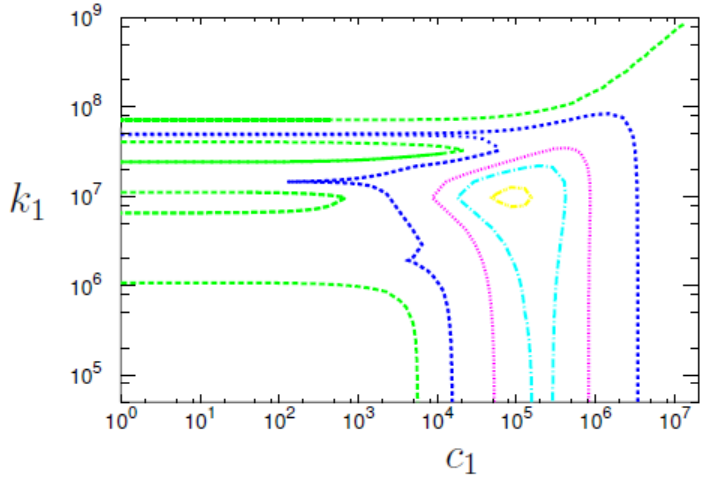


	# iterations	time (min)
$\gamma^{(1)}$	8	0.45
$\gamma^{(2)}$	4	0.25
$\gamma^{(3)}$	7	0.43
$\gamma^{(4)}$	3	0.19
Total	22	1.32

Model order reduction (MOR) in optimization context

Dynamical system:

$$\begin{aligned} A(\omega,\gamma)x &= f \\ y &= d^T x \\ g &= \int_{\omega_{\min}}^{\omega_{\max}} |y|^2 d\omega \quad \text{(energy)} \end{aligned}$$



- Objective: minimize $g(\gamma)$
- Model reduction using moment matching = Padé approximation via Krylov methods
- Allows for cheap computation of g and $\nabla_\gamma g$
- Penalty and trust region methods based on error estimation of reduced model (Yao Yue)

Example of the Lamot footbridge

- OPTEC Application Problem 4
- Lamot bridge finite element model ($n = 25,962$)
- The goal is to determine the optimal stiffness and damping coefficient of four bridge dampers (=8 parameters).



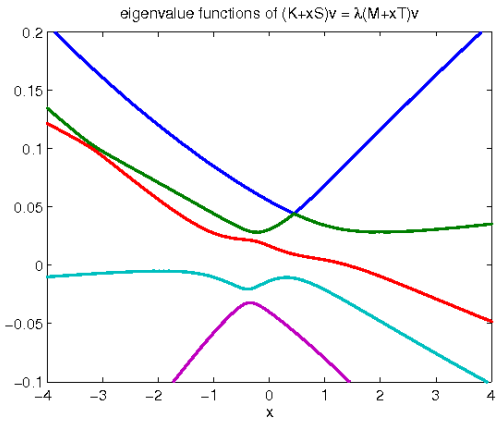
- Computation times:
  - without reduced modeling: 545 for one function evaluation, 70 function evaluations needed.
  - with reduced model to evaluate g and ∇g: 879 sec.
  - trust region method: 200 sec.
  - penalty method: 189 sec.

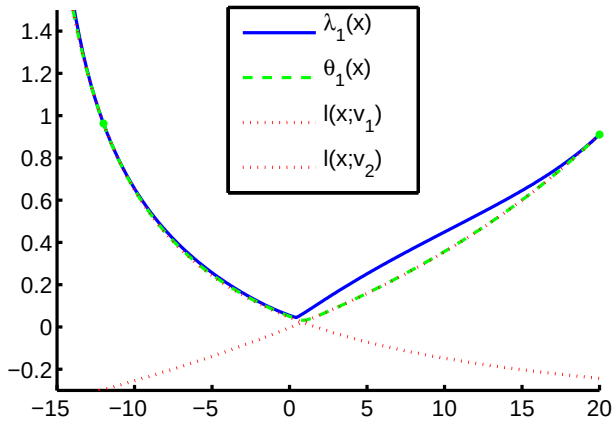
Minimal maximal eigenvalue

- Parametric eigenvalue problem

$$A(x)u = \lambda B(x)u$$

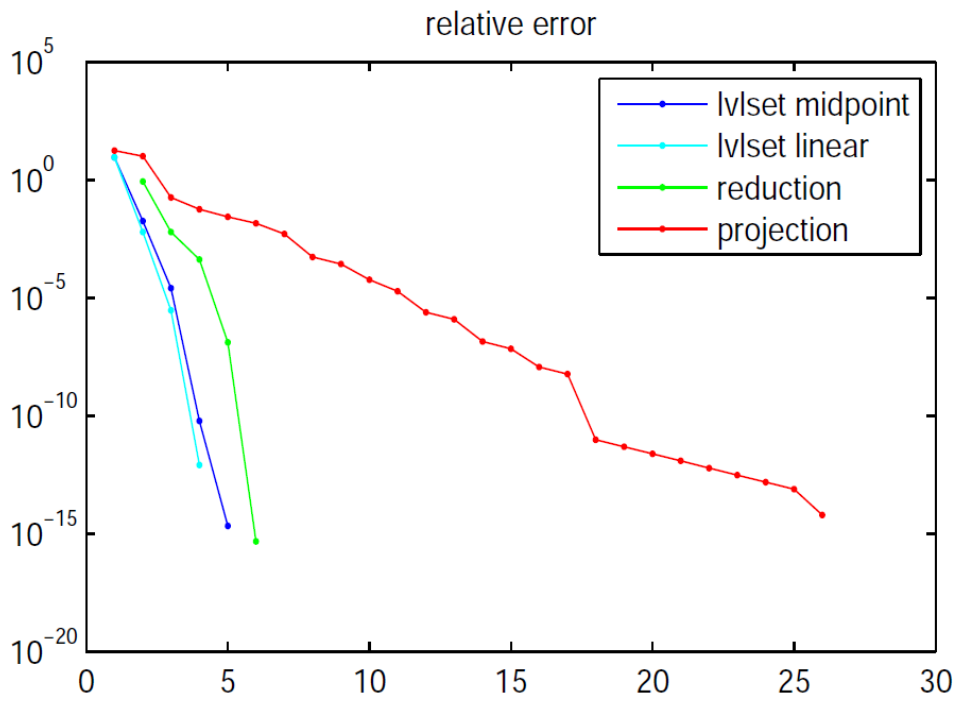
with $x \in \mathcal{H} \subset \mathbb{R}^m$



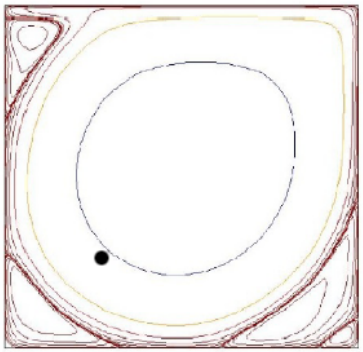


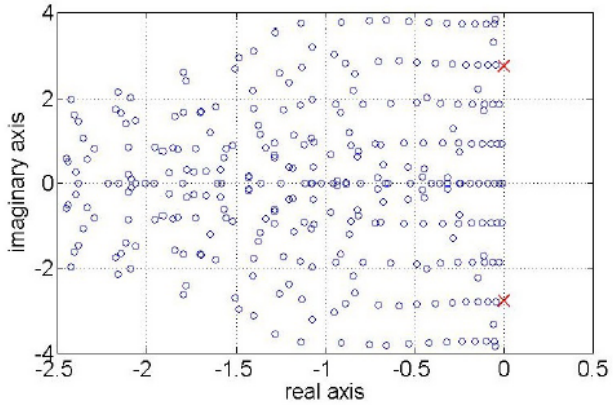
- A and B symmetric and B positive definite for $x \in \mathcal{H}$
- Largest eigenvalue is a quasi convex function
- Our numerical method (Jeroen De Vlieger):
  - subspace projection method (= bundle method)
  - Solve the projected problem by any method for non-smooth problems or a method that approximates the solution by sequence of a 1D problems
  - Provably convergent

Minimal maximal eigenvalue



Parametric eigenvalue problems for stability analysis





- Find parameter values p so that

$$A(p)x = \lambda Bx$$

  has purely imaginary eigenvalues λ.
- Is translated to

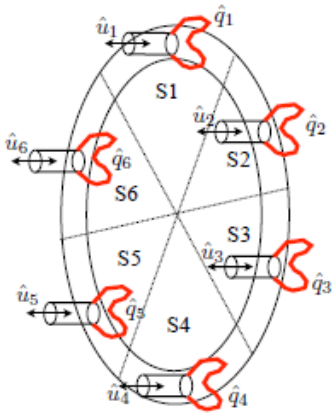
$$(A(p) \otimes B + B \otimes A(p))z = 0$$

- Exploit the specific Kronecker structure for efficient solvers
- Appears to be as efficient as solving eigenvalues of $A(p)$ for fixed p
- Much more efficient than continuation
- Work with Spence, Elman, Voss, Schroeder, Vandebril

Nonlinear eigenvalue problems

- Nonlinear eigenvalues problems arise in many applications:

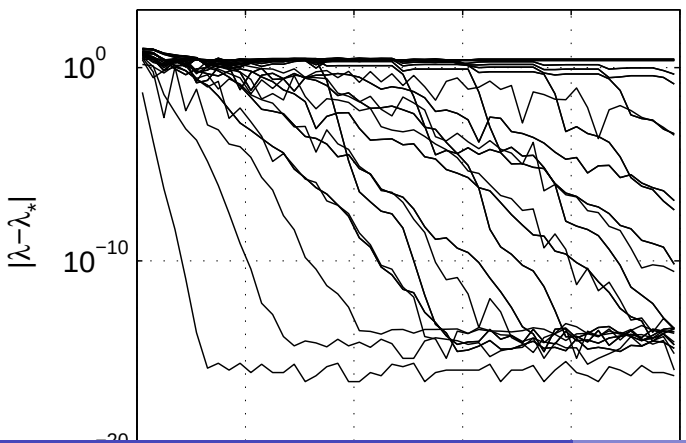
$$A(\lambda)x = 0$$

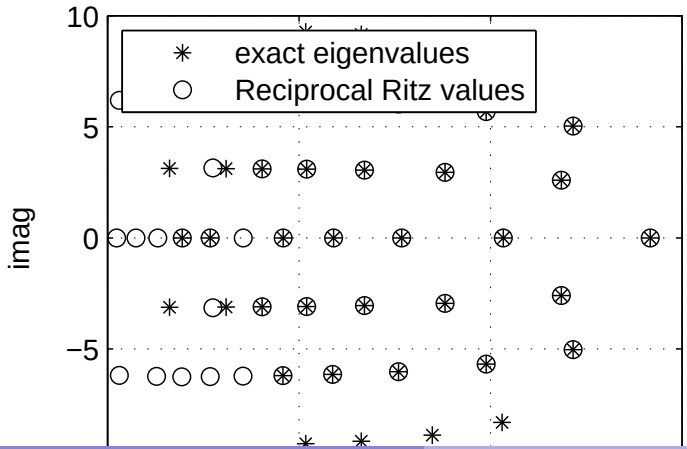


- Classical iteration schemes converge to one eigenvalue at a time
  - Approximate

$$A(\lambda) \simeq A_0 + \lambda A_1 + \lambda^2 A_2 + \cdots$$

  - Apply Arnoldi's method to the Companion linearization: several eigenvalues converge at the same time
  - With Michiels, Jarlebring, Van Beeumen





Nonlinear eigenvalue problems

- Rational Krylov method: pole can change
- Approximate $A(\lambda)$ by Newton interpolation in $\sigma_1, \ldots, \sigma_k$

$$\begin{aligned} A(\lambda) &\simeq A_0 n_0(\lambda) + A_1 n_1(\lambda) + \cdots + A_k n_k(\lambda) \\ n_j &= \prod_{i=1}^{j} (\lambda - \sigma_i) \end{aligned}$$

- Rational Krylov method with starting vector v corresponds to moment matching:

$$A(\sigma_1)^{-1} v, A(\sigma_2)^{-1} v, \ldots, A(\sigma_k)^{-1} v$$

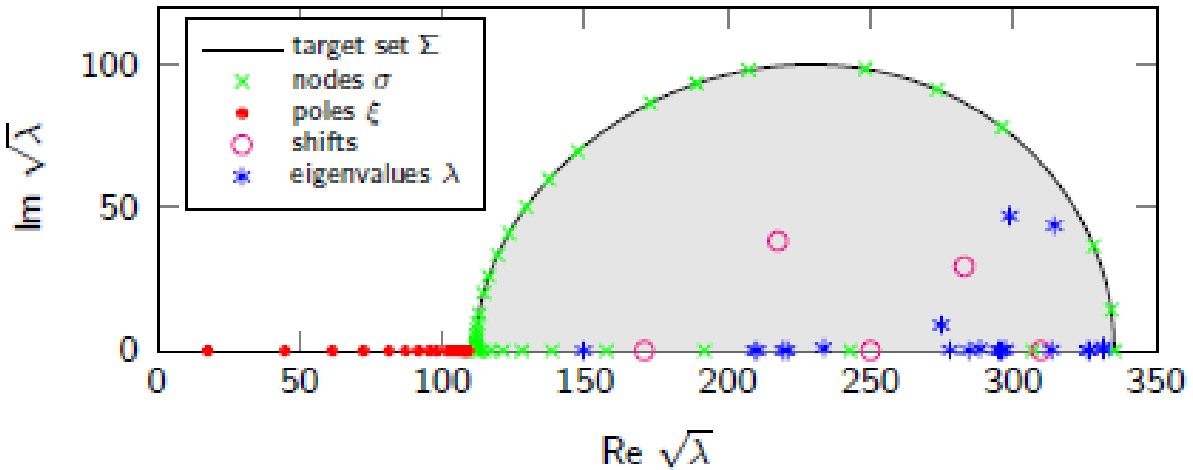
are contained in the subspace.

Nonlinear eigenvalue problems

- Gun problem:

$$F(\lambda)x = 0 = \left(K - \lambda M + i\sqrt{\lambda - \sigma_1^2}W_1 + i\sqrt{\lambda - \sigma_2^2}W_2 \right) x$$

- Compute eigenvalues near 5 poles.



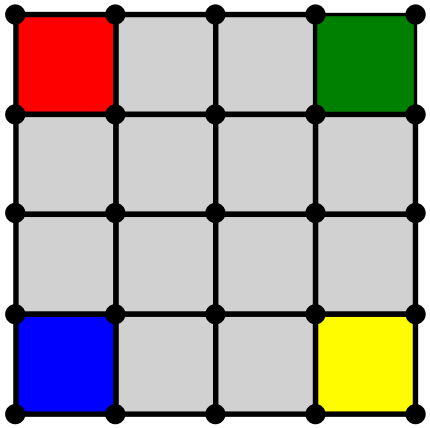
Incomplete multifrontal factorization

- Large sparse linear system $Ax = b$ where A is given in finite element format:

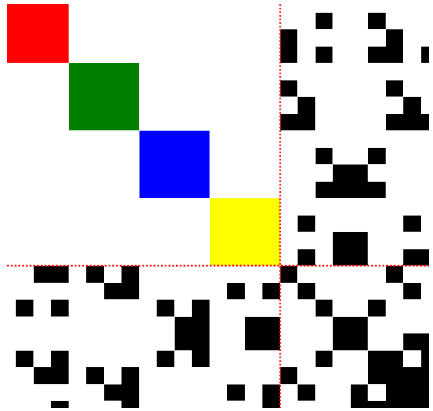
$$A = \sum_{e \in \mathcal{E}} P_e A_e P_e^T$$

  - A_e is dense (full) matrix
  - P_e selects degrees of freedom associated with the element

- Multilevel preconditioning using element structure instead of sparse.



$\mathcal{D}_k \equiv \langle$ ■, ■, ■, ■ $\rangle$

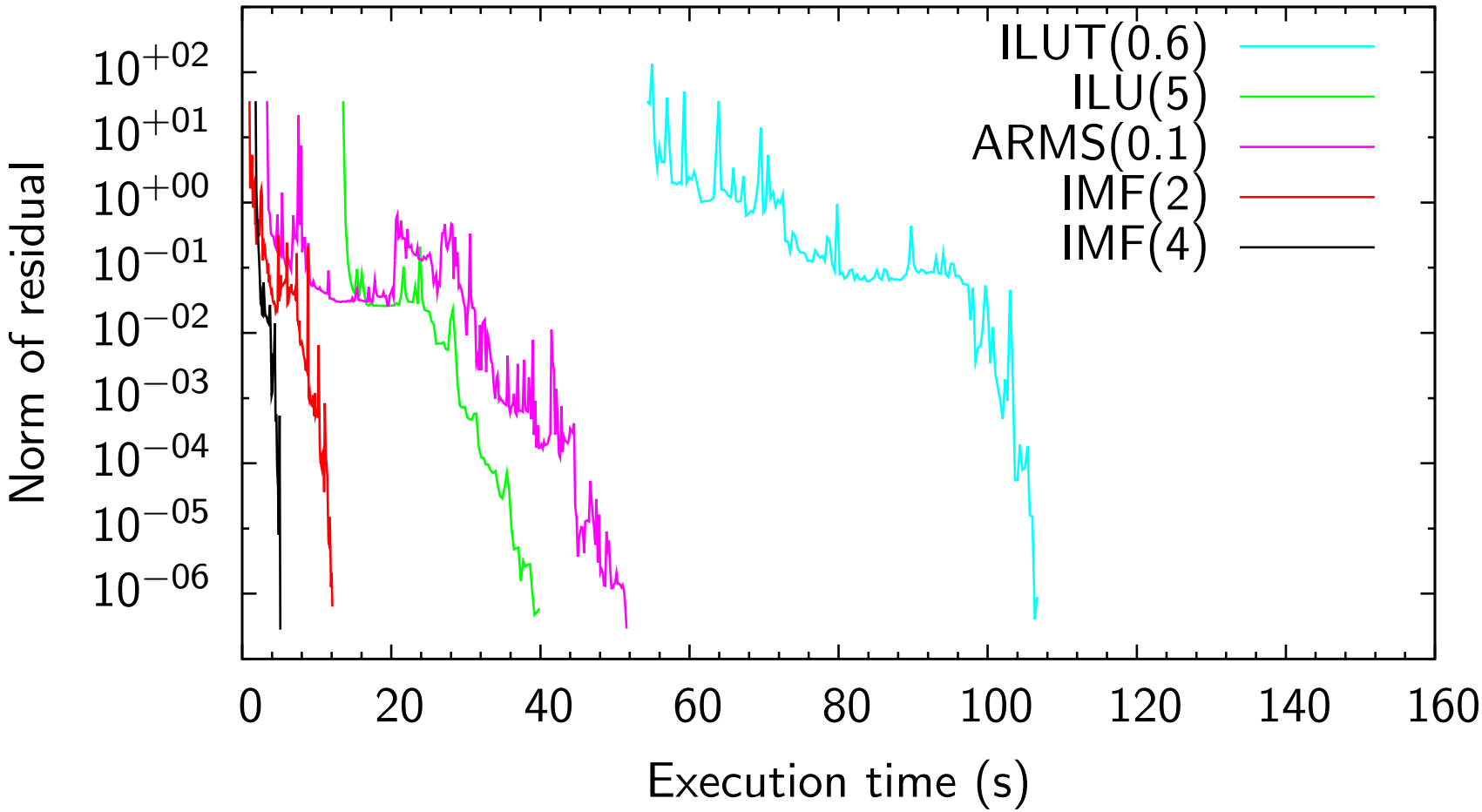


$D_k \equiv \mathrm{diag}($ ■, ■, ■, ■ $)$

- IMF (Nick Van Nieuwenhoven)

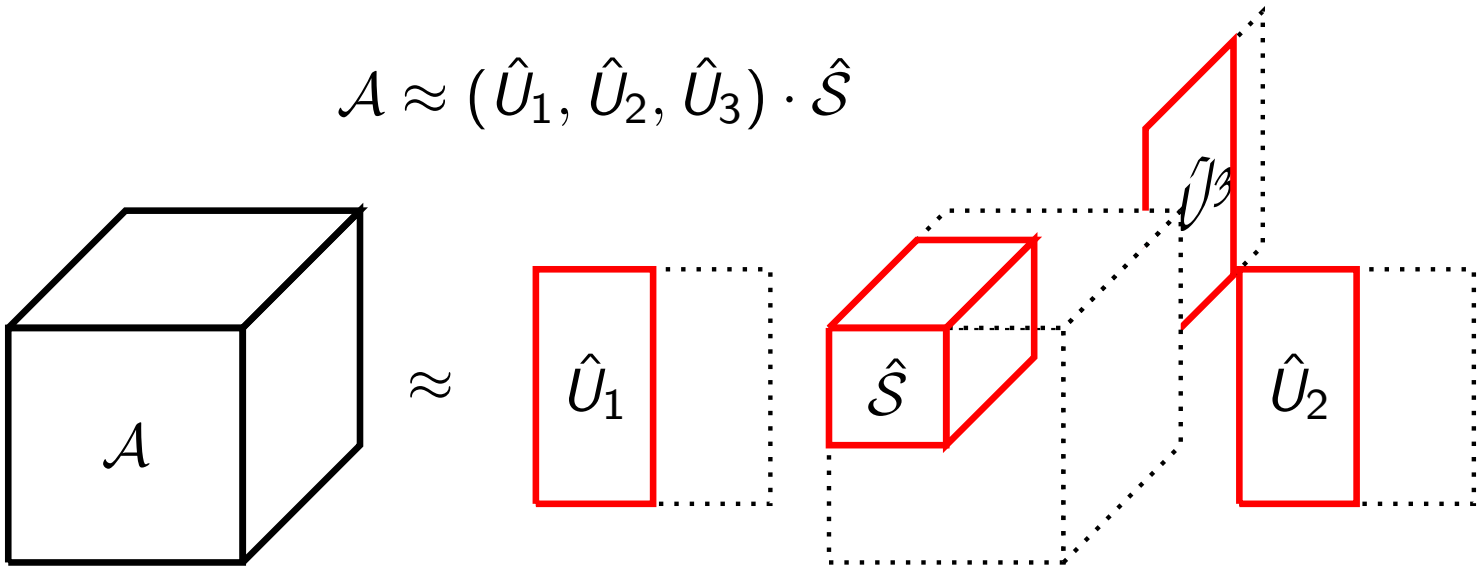
Incomplete multifrontal factorization

- Advantages
  - Use dense matrices: high throughput
  - More efficient preconditioner than other multilevel techniques
- 2D Navier-Stokes equation, flow around obstacle, Reynolds number 4000, 16 128 elems (Q1-Q1), 49 440 unknowns, 1 243 530 nnzs.



Sequentially truncated HOSVD

$$\mathcal{A} \approx (\hat{U}_1, \hat{U}_2, \hat{U}_3) \cdot \hat{\mathcal{S}}$$

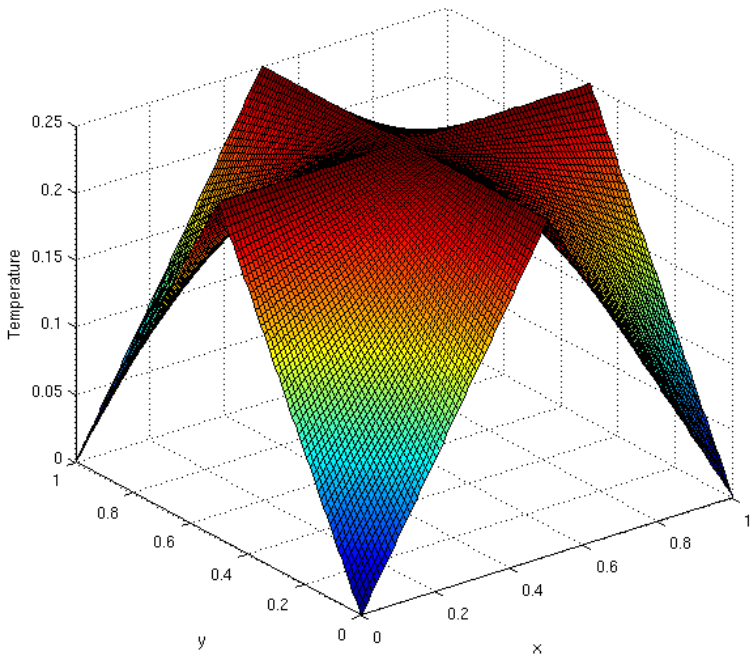


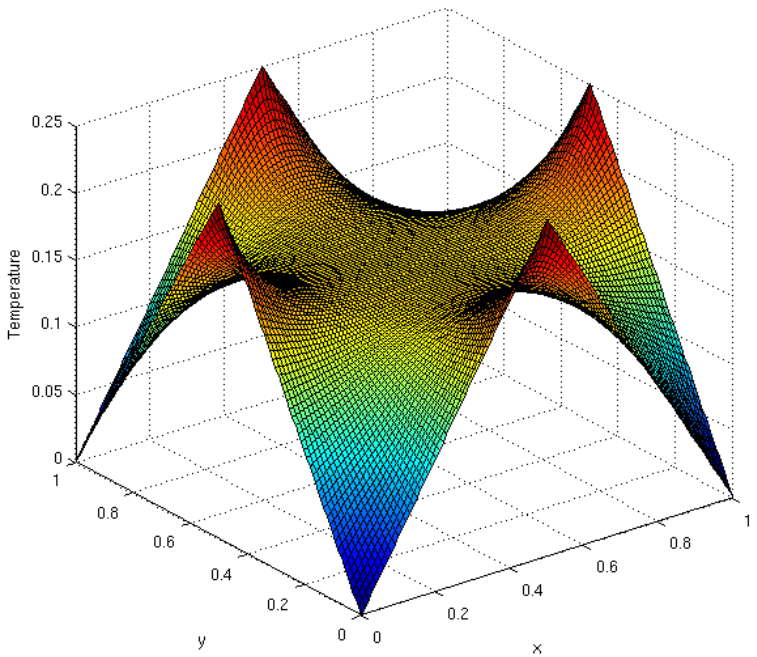
Rank (r_1, r_2, r_3) orthogonal Tucker approximation to $\mathcal{A}$

$$\text{Columns of } \begin{cases} \hat{U}_1 \in \mathbb{R}^{n_1 \times r_1} \\ \hat{U}_2 \in \mathbb{R}^{n_2 \times r_2} \\ \hat{U}_3 \in \mathbb{R}^{n_3 \times r_3} \end{cases} \text{ can be extended to a basis of } \begin{cases} \mathbb{R}^{n_1} \\ \mathbb{R}^{n_2} \\ \mathbb{R}^{n_3} \end{cases}$$

Compression of simulation results I

Results for Nick Vannieuwenhoven's ST-HOSVD.
Compression of numerical solution of the heat equation on a square domain computed by explicit Euler.





Tensor of size $101 \times 101 \times 10001$ ($x \times y \times t$).

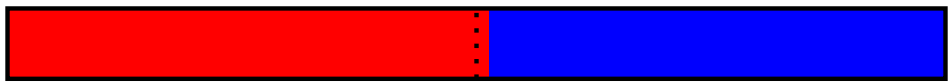
- Partially symmetric, 102.0 million non-zeros.
- T-HOSVD and ST-HOSVD truncated to absolute error of 10^{-4}.

Compression of simulation results II

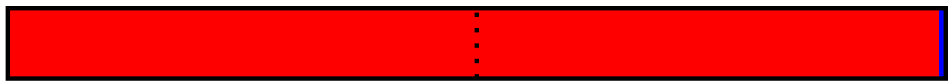
	T-HOSVD	ST-HOSVD
Abs. error	$8.512 \cdot 10^{-5}$	$9.587 \cdot 10^{-5}$
Rank	$(22, 22, 20)$	$(22, 21, 19)$

T-HOSVD ST-HOSVD

Storage (nb. of values)

214144  203140

Factorization time

2h 46m  1m 14.7s

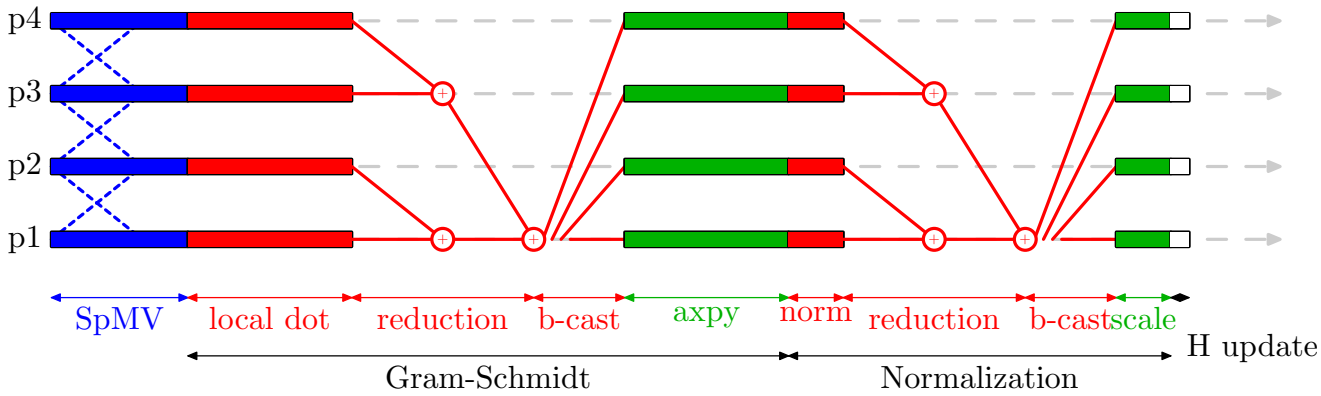
133x speedup!

Flanders ExaScience Lab

- Project to give hints for future (exascale) HPC hardware
- Partners: all Flemish universities, imec, intel, later Janssen pharma
- Power requirements:
  - Exascale machine limited by 25MW
  - With Blue Gene technology, 593MW (Nuclear power plant: 1008MW)
  - Energy budget:
    - ★ Reduce clock speed (= lower voltage)
    - ★ Increase number of cores
    - ★ Therefore higher power efficiency (= ops/watt)
    - ★ Less fault correction hardware
    - ★ Higher chances of hardware errors to be solved by software
- Work done:
  - Support for Space Weather simulation: Adaptive Mesh Refinement, Particle in Cell.
  - Iterative linear system solvers
  - Load balancing
  - Tensor computations for parameter estimation for PDE and for machine learning

Latency hiding in Krylov methods

- Communication avoidance (Demmel et al.): perform s Krylov steps within one bulk synchronisation/communication step.
- Classical Krylov:



- Latency hiding:

