Matrix-free vs. matrix-based real-time reconstruction on CPU and GPU

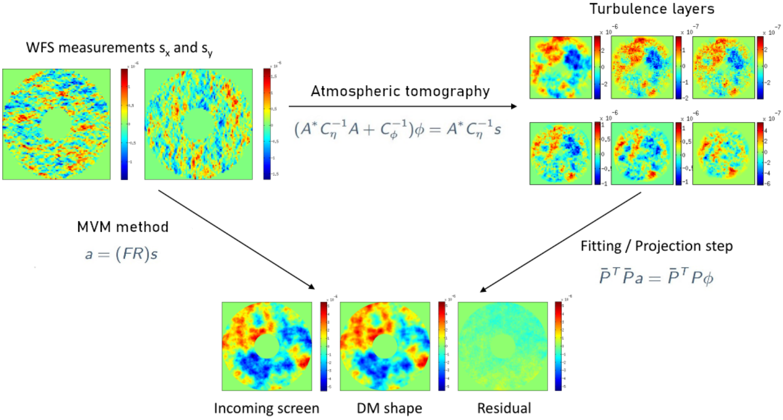
Bernadett Stadler, Daniel Jodlbauer, Ronny Ramlau

bernadett.stadler@ricam.oeaw.ac.at

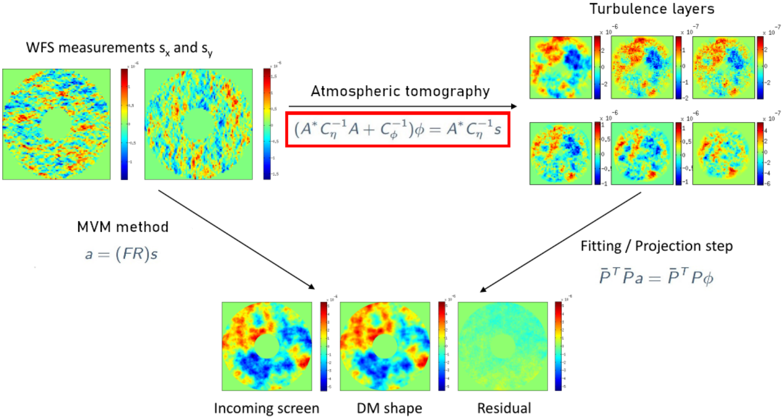
Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences (ÖAW)
Linz, Austria

November 9th, 2023

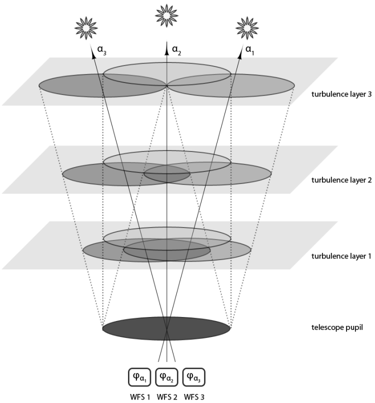
Adaptive Optics control



Adaptive Optics control



Atmospheric tomography



- **Goal:** Reconstruct turbulent layers ϕ from sensor measurements s

$$s = A\phi$$

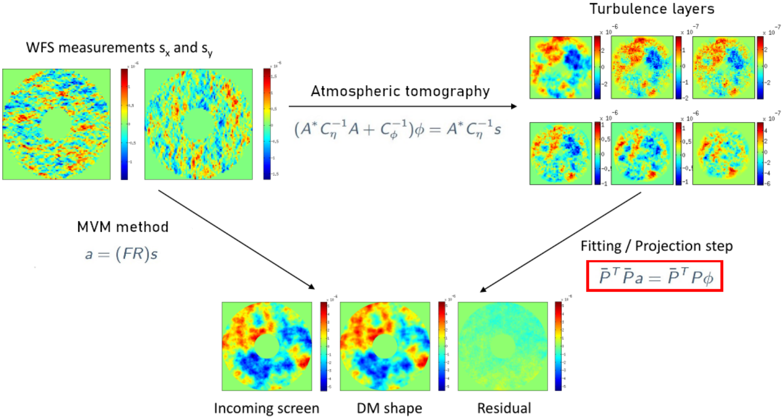
- **Regularization:** Bayesian framework and maximum a-posteriori estimate

$$(A^* C_\eta^{-1} A + C_\phi^{-1})\phi = A^* C_\eta^{-1} s$$

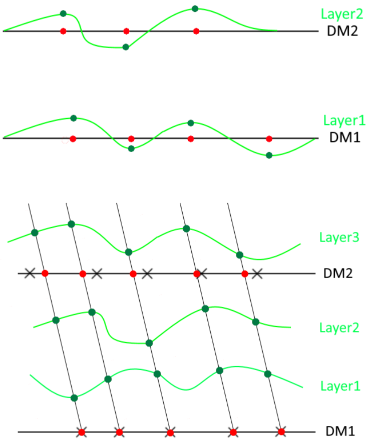
- **Reconstruction matrix**:

$$R := (A^T C_\eta^{-1} A + C_\phi^{-1})^{-1} A^T C_\eta^{-1}$$

Adaptive Optics control



Mirror fitting



- **Goal**: Project reconstr. layers onto DMs.
- If more layers than DMs are reconstructed an additional problem has to be solved

$$\bar{P}^T \bar{P} a = \bar{P}^T P \phi$$

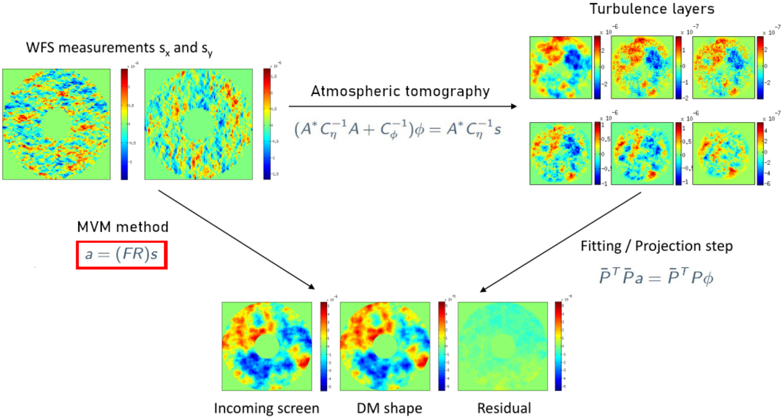
P ... project. through layer ℓ in dir. $\bar{\theta}_n$

$\overline{P}$ ... project. through DM m in dir. $\bar{\theta}_n$

- **Fitting matrix**:

$$F := (\bar{P}^T \bar{P})^{-1} \bar{P}^T P$$

Adaptive Optics control



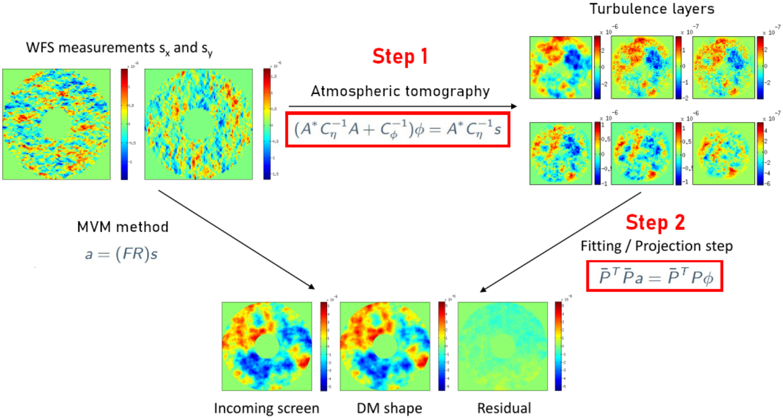
Direct, matrix-based approach

- **Discretization**: Typically with the basis of Zernike polynomials
- **Soft real-time**: Precomputation of the system matrix (FR), which consists of the reconstruction matrix R and the fitting matrix F.
- **Hard real-time**: Actuator commands computed from WFS measurements via matrix-vector multiplication

$$a = (FR)s$$

- Factorization techniques, such as Cholesky decomposition, used to compute and store the matrix in a more efficient way
- **Parallelization**: Straight forward with standard libraries

Adaptive Optics control



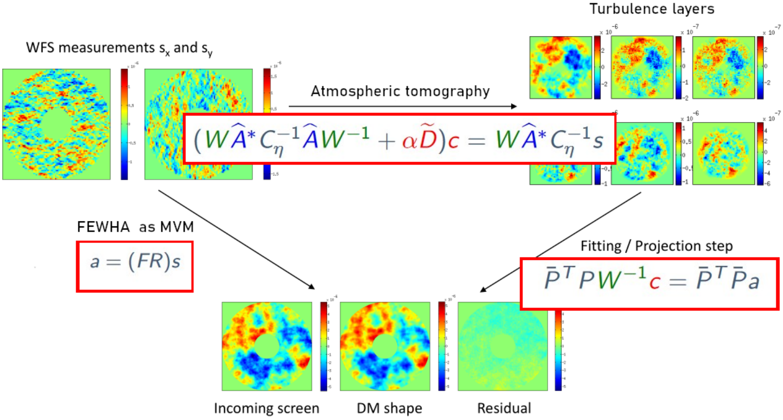
Iterative, matrix-free approach

- **Discretization**: With a basis that allows a sparse representation (e.g. Fourier, Wavelets, Hierarchical matrices)
- **Soft real-time**: No precomputation of the inverse is required
- **Hard real-time**: Tomography and fitting problem are solved separately
- **Iterative solver**: Conjugate Gradient method
- Various methods exist:
  - $\mathcal{O}(n \cdot \log(n))$: Fourier Domain PCG (FD-PCG)
  - $\mathcal{O}(n)$: Fractal Iterative Method (FrIM)
  - $\mathcal{O}(n)$: Finite Elem. Wavelet Hybrid Algorithm (FEWHA)

Iterative, matrix-free approach

- **Discretization**: With a basis that allows a sparse representation (e.g. Fourier, Wavelets, Hierarchical matrices)
- **Soft real-time**: No precomputation of the inverse is required
- **Hard real-time**: Tomography and fitting problem are solved separately
- **Iterative solver**: Conjugate Gradient method
- Various methods exist:
  - $\mathcal{O}(n \cdot \log(n))$: Fourier Domain PCG (FD-PCG)
  - $\mathcal{O}(n)$: Fractal Iterative Method (FrIM)
  - $\mathcal{O}(n)$: **Finite Elem. Wavelet Hybrid Algorithm (FEWHA)**

Adaptive optics control with FEWHA

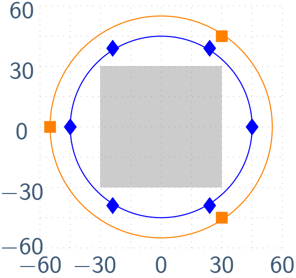


Numerical simulations

- **Test configuration**: MORFEO-like setting
- **Control algorithm**: FEWHA
- **Goal**: Compare matrix-based and matrix-free implementation
- **Evaluation**: LE Strehl ratio and run-time

Parameter	Value
Telescope diameter	37 m
Central obstruction	11%
DMs	3
WFS	6 LGS, 3 NGS
Reconstr. layers	9
Simulated duration	1s
Evaluation	LE Strehl
Wavelength	K band

Guide star asterism:

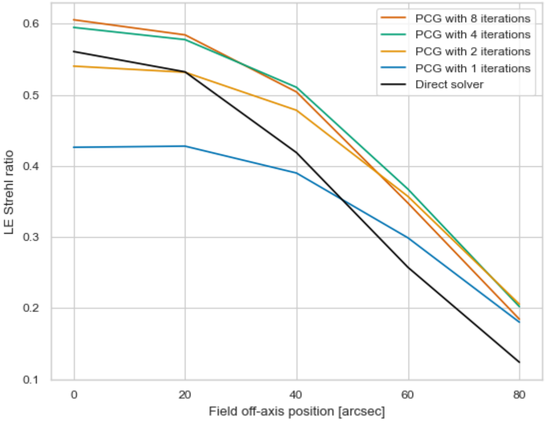


Implementation & Testing

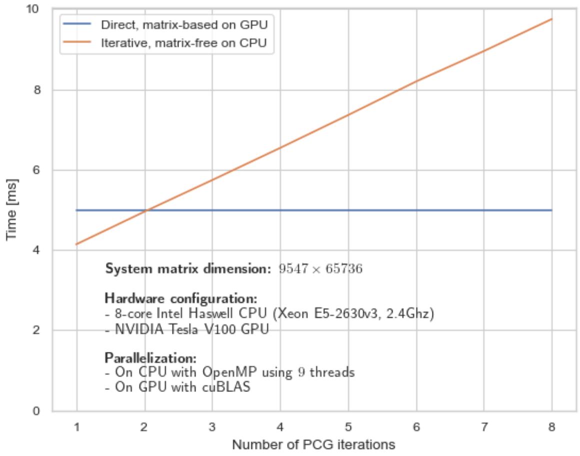
- Languages: C++, CUDA
- Parallelization: OpenMP, cuBLAS
- Interfaces: **pyfewha**, cfewha
- Simulators: **COMPASS**, Octopus
- Formatting: clang-format
- Compilers: gcc, **clang**
- Automatic testing on merge requests:
  - Using Gitlab CI with Docker runners
  - Unit tests with Catch2
  - Integration tests with simulators
  - Sanitizer: address, memory, thread, undefined behavior

Reconstruction quality

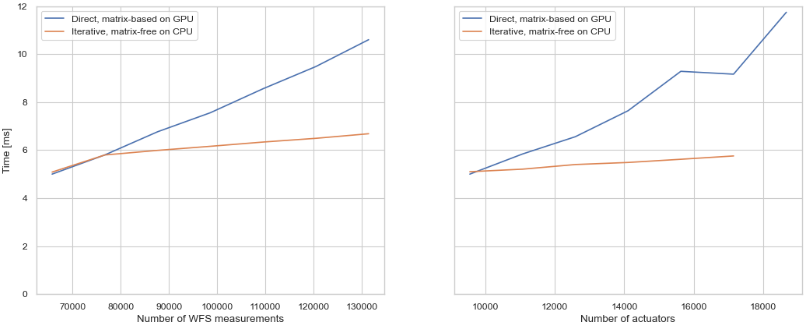
- **Direct solver**: FEWHA as MVM
- **Iterative solver**: Varying tomography PCG iter., 1 fitting PCG iter.



Run-time


Direct, matrix-based on GPU
Iterative, matrix-free on CPU
Time [ms]
Number of PCG iterations
System matrix dimension: 9547 × 65736
Hardware configuration:
- 8-core Intel Haswell CPU (Xeon E5-2630v3, 2.4Ghz)
- NVIDIA Tesla V100 GPU
Parallelization:
- On CPU with OpenMP using 9 threads
- On GPU with cuBLAS

Scaling with number of WFSs or DMs



Conclusion: Quality

- For FEWHA an iterative solver with $2-8$ iteration provides a better LE Strehl ratio than a direct one.
- An earlier stopping of the PCG method omits overfitting to the regularized problem.
- The number of PCG iterations act as additional tuning parameter.

$\Rightarrow$ Similar results with other iterative reconstructors e.g. FrIM?

Conclusion: Run-time

- Depending on the system size hard real-time AO control can be faster with a matrix-based or a matrix-free approach.
- The iterative solver does not require the precomputation of the inverse in soft real-time.
- For FEWHA the iterative solver is faster on a CPU, because parallelization possibilities are limited.
- Pipelining is not considered here and only possible for the matrix-based approach.

Thank you!

- Bernadett Stadler et al. "Matrix-free vs. matrix-based real-time control for a MORFEO-like setting". In: Proceedings AO4ELT7. Submitted.
- Bernadett Stadler and Ronny Ramlau. "Performance of an iterative wavelet reconstructor for the Multi- conjugate Adaptive Optics RelaY of the Extremely Large Telescope". In: Journal of Astronomical Telescopes, Instruments, and Systems 8.2 (2022), p. 021503.
- Bernadett Stadler et al. "Parallel implementation of an iterative solver for atmospheric tomography". In: 2021 21st International Conference on Computational Science and Its Applications (ICCSA). 2021, pp. 123-132.

The project has received funding by the Austrian Research Promotion Agency (FFG) No. FO999888133 and the NVIDIA Corporation Academic Hardware Grant Program.