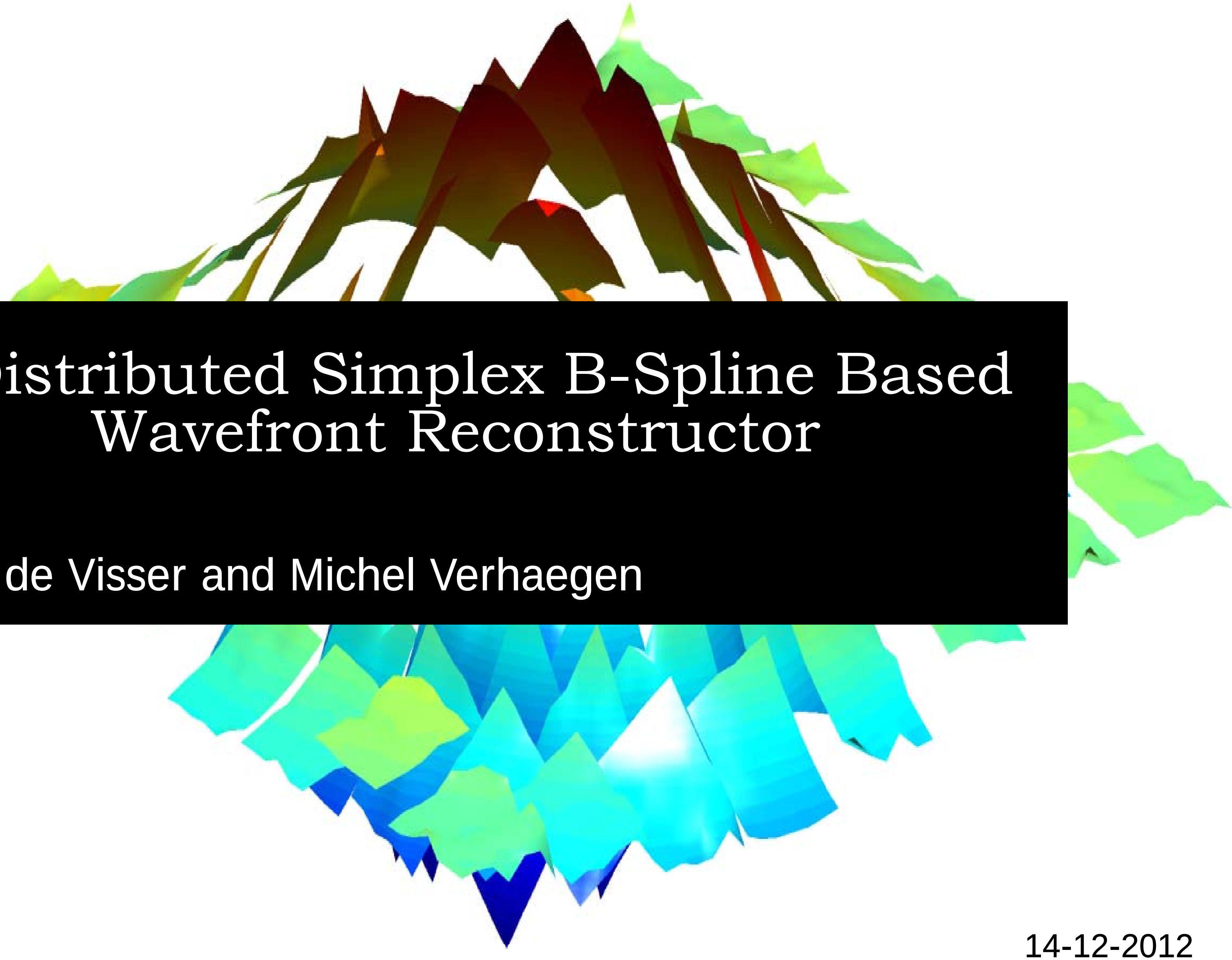
A Distributed Simplex B-Spline Based Wavefront Reconstructor

Coen de Visser and Michel Verhaegen



14-12-2012

TU Delft
Delft University of Technology

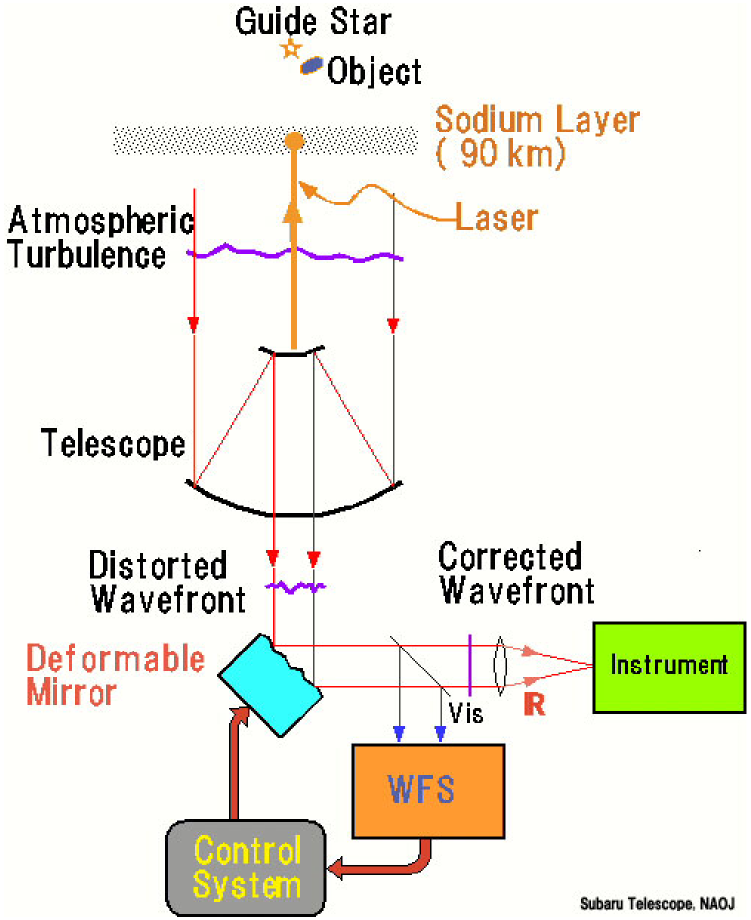
Contents

Introduction

Wavefront reconstruction (WFR):

- necessary because wavefront phase cannot be measured directly
- computationally expensive and “Key” operation in AO

Example: for E-ELT XAO system using standard Matrix-Vector-Multiplication:
4.8 TFLOPS
Current single core CPU performance:
18 GFLOPS (Core i7-980)

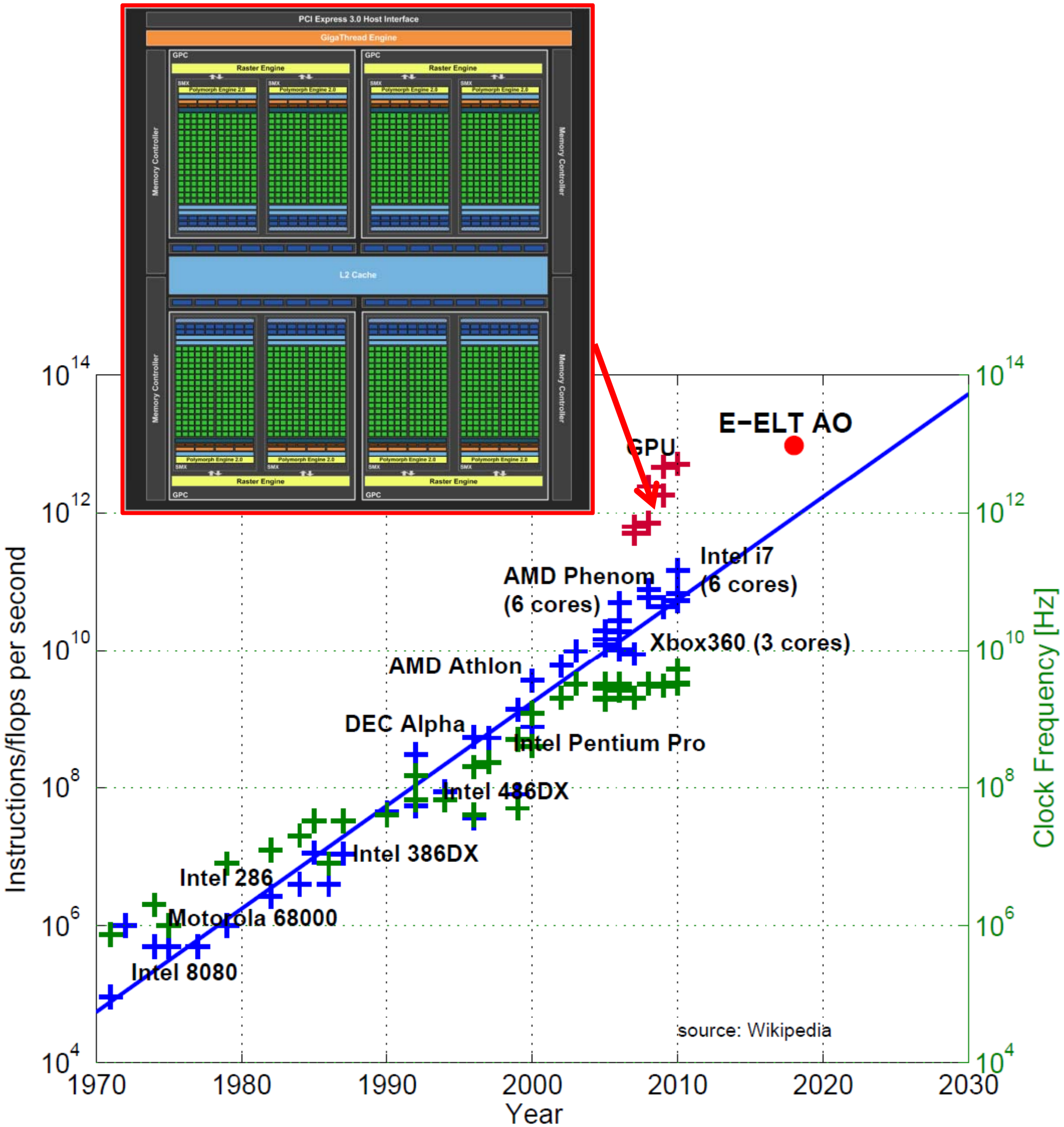


Introduction

Increase of computational performance in the near future **only through parallelization**.

Large scale WFR for XAO requires parallelization!

Simplex B-spline (SABRE*) method is a WFR method that enables massive parallelization and implementation on GPU.



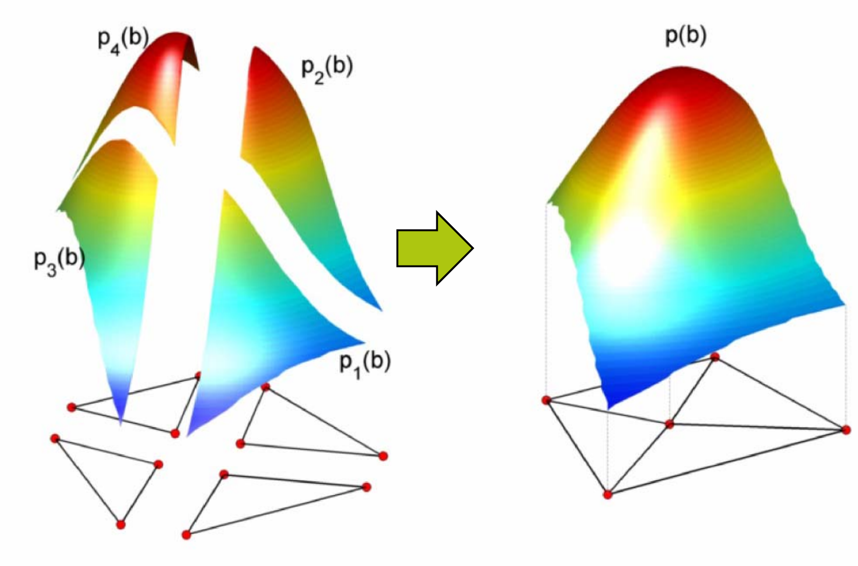
* C.C. de Visser and M. Verhaegen, *A Wavefront Reconstruction in Adaptive Optics Systems using Nonlinear Multivariate Splines*, **JOSA A**, accepted for publication.

Spline based Aberration Reconstruction

Recently, a new method called the SABRE (Spline based ABeration REconstruction) for **local wavefront reconstruction** was introduced*.

The SABRE uses nonlinear bivariate **splines** to locally approximate the wavefront.

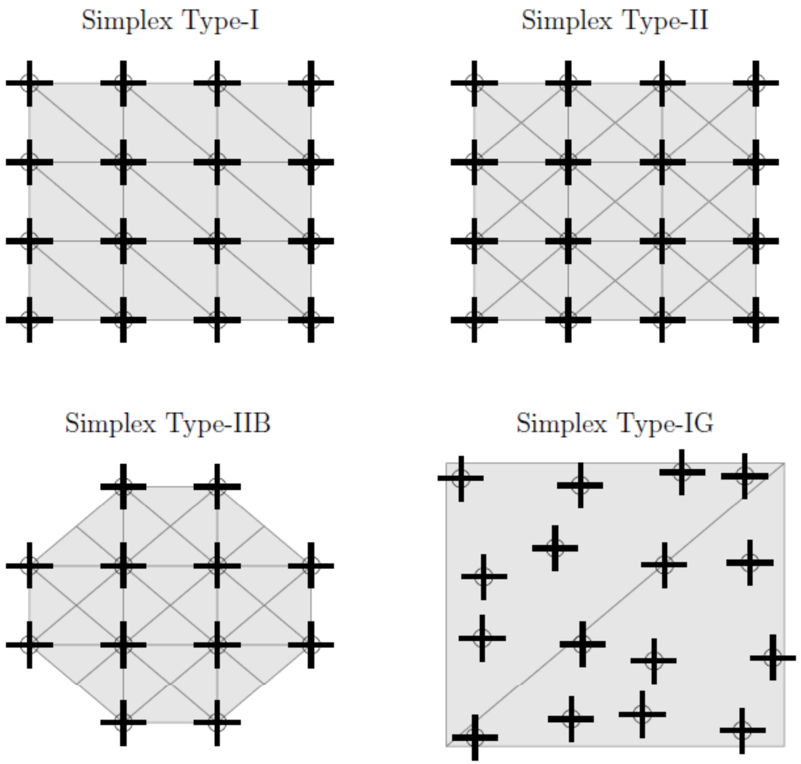
The SABRE uses **triangular sub-partitions** of the global wavefront sensor grid and estimates local wavefront phase.



* C.C. de Visser and M. Verhaegen, *A Wavefront Reconstruction in Adaptive Optics Systems using Nonlinear Multivariate Splines*, **JOSA A**, accepted for publication.

Spline based Aberration Reconstruction

- SABRE is compatible with many different wavefront sensor geometries (occlusion, misalignment, etc.).
- SABRE can approximate the wavefront using nonlinear polynomial basis functions.
- SABRE was shown to exceed reconstruction accuracy of Fried FD methods for all noise levels(*).
- SABRE can be implemented in a distributed manner*



Black crosses: SH lenslet locations

Grey lines: triangular sub-partitions

*This lecture

Spline based Aberration Reconstruction

SABRE models the wavefront through "local" basis functions plus continuity constraints:

$$\hat{\Phi}_{\mathbf{SABRE}}(x,y) = B^d(x,y).\hat{c} \quad d \in \mathbb{N}^- \qquad A\hat{c} = 0$$

Polynomial basis function of degree d

Estimated spline coefficients

SABRE slope sensor model is linear in the parameters (c):

$$s(x,y) = d \cdot B^{d-1}(x,y) \cdot P^{d,d-1}(u) \cdot c + n(x,y)$$

Slope measurements

"De Casteljau" matrix(*) of degree d to d-1 as a function of derivative direction u

Noise model

(*)C.C. de Visser *et al.*, *Differential Constraints for Bounded Recursive Identification with Multivariate Splines*, **Automatica**, 2011

Spline based Aberration Reconstruction

Constrained optimization problem for the spline coefficients c is :

$$\min_{c} \left(s - d.B^{d-1}.P^{d,d-1}(u)c \right) \quad \text{subject} \quad \text{to} \quad A.c = 0$$

With the **sparse matrix** A containing the spline smoothness constraints.

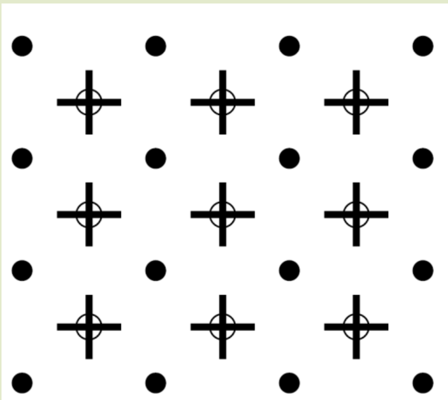
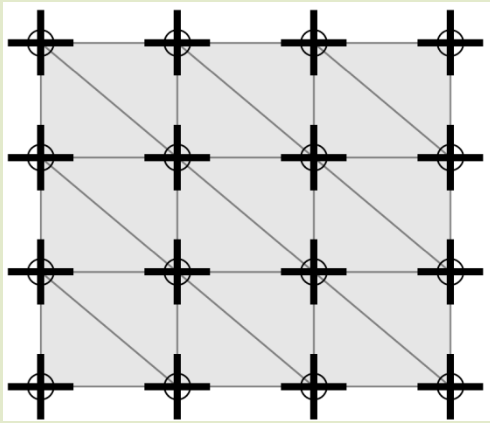
Now define N_A as the null-space projector of A:

$$N_A = \ker(A)$$

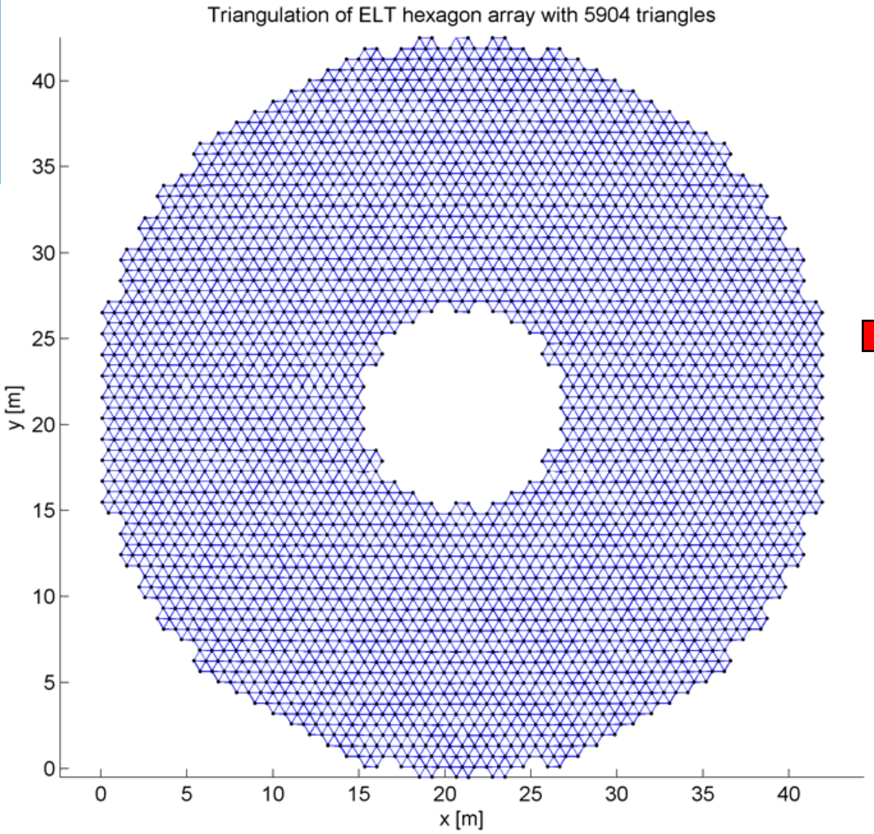
The constrained optimization problem can now be reduced to an **unconstrained** problem by using a projector on the null-space of A as follows:

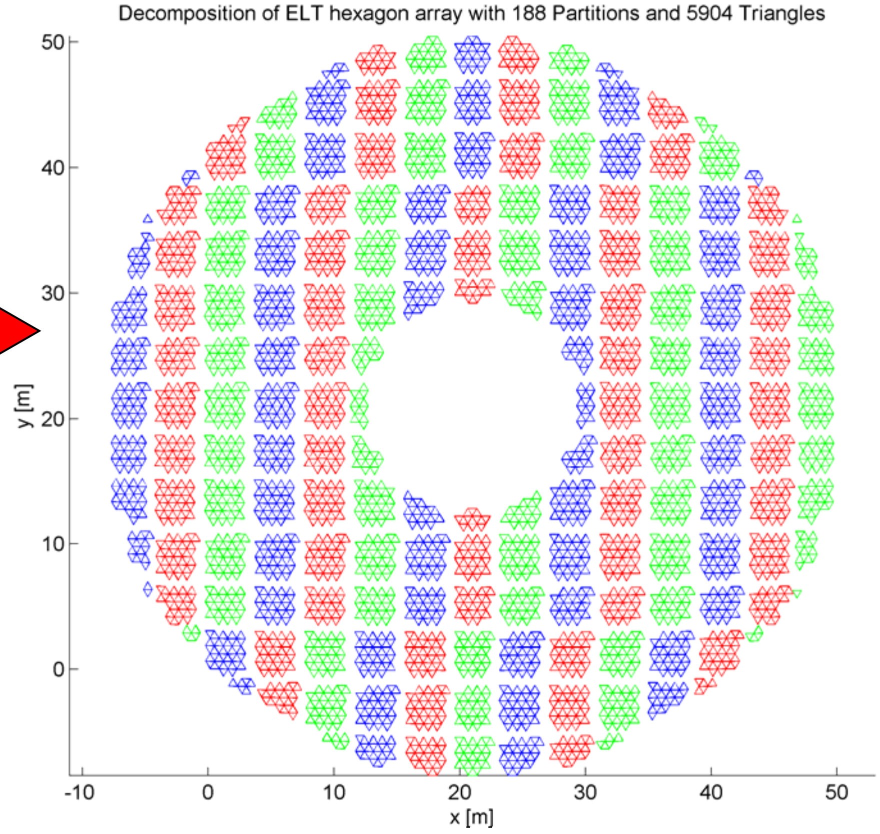
$$\min_{\gamma} \left(s - d.B^{d-1}.P^{d,d-1}(u)N_A\gamma \right)$$

Comparison Fried FD and SABRE

	Fried Finite Difference	**SABRE**
Wavefront model	$\hat{\phi}_{FD} = G^{+}s$	$\hat{\phi}_{SABRE}(x,y) = B^{d}(x,y)\cdot\hat{c},\ \ d > 0$
Reconstruction matrix	G^{+} $(pseudo\ inverse\ of\ G)$	$N_{A}(D^{T}D)^{-1}D^{T}$
Sensor geometry		

Distributed-SABRE



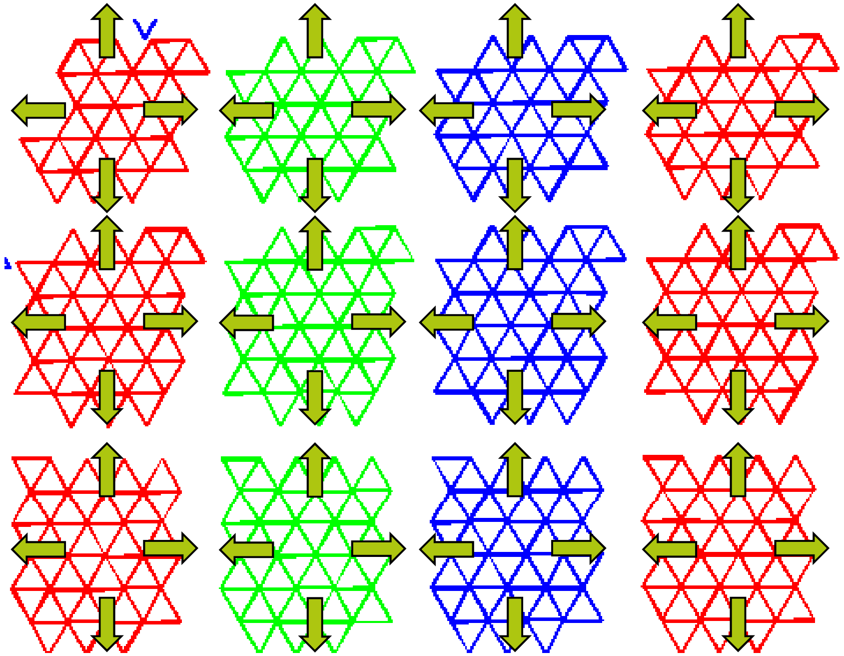


Full domain is partitioned into any number of partitions.

Each partition runs on a separate CPU/GPU core.

Distributed-SABRE

Principle of Distributed WFR: each partition depends only on its direct neighbors



Problem: Each partition will have an unknown piston mode, and will be discontinuous with its neighbors on its borders... a three stage solution

Distributed-SABRE

D-SABRE is a three stage method:

Stage 1: local wavefront reconstruction (local LS problem) for partition i:

$$\hat{c}_i = N_{A_i}(D_i^T D_i)^{-1} D_i^T s_i$$

where c_i are the coefficients of the splines used to model the wavefront over the i-th partition

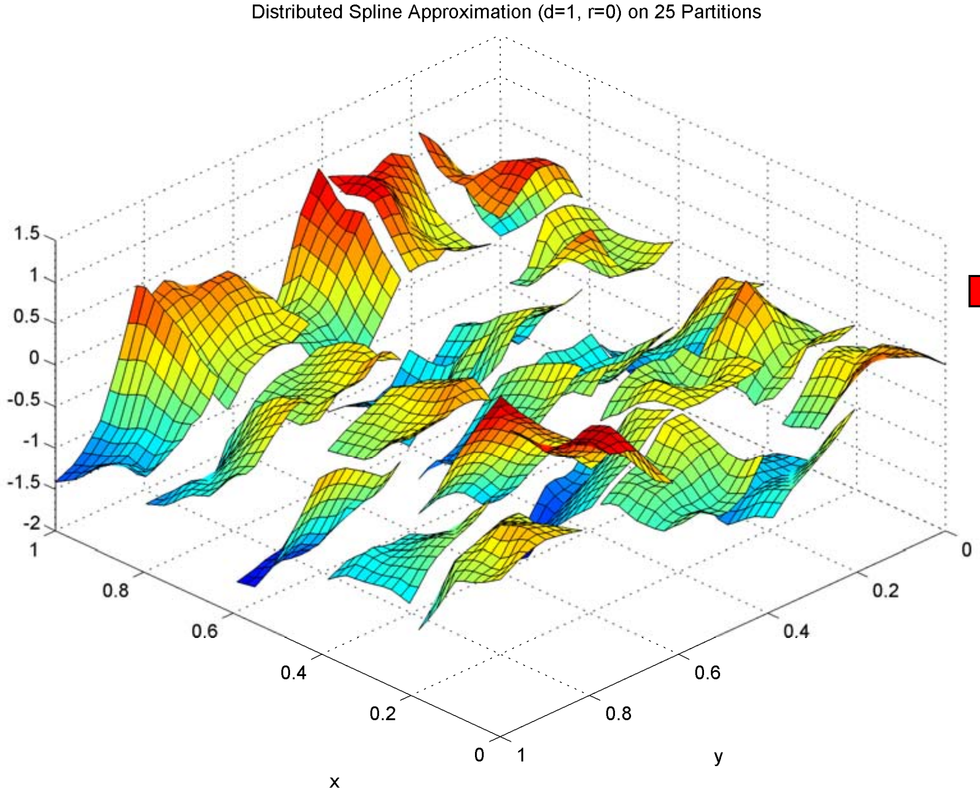
Stage 2: distributed (iterative) Piston Mode Equalization (PME) for partition i with respect to neighbor partition j:

$$m = mean\left(\hat{c}_i(I) - \hat{c}_j(J)\right)$$

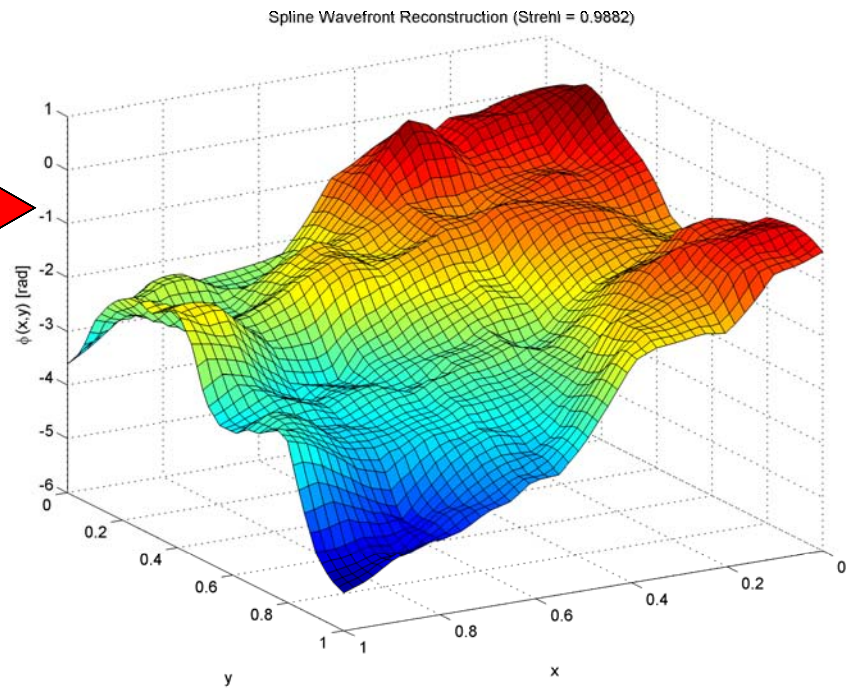
$$\hat{c}_i = \hat{c}_i - m$$

First 2 stages of D-SABRE illustrated

Stage 1



Stage 2



Local WF is estimated using local WF measurements.

Global WF is reconstructed in two extra stages: distributed piston mode equalization (PME) and inter-partition smoothing.

Stage 3 of D-SABRE

Stage 3: distributed iterative inter-partition smoothing using distributed Dual Ascent (DA) method(**):

Dual variable y is updated using partition A_{i+j} of constraint matrix A:

$$y_i(k+1) = y_i(k) + \alpha \cdot A_{i+j} \cdot \hat{c}_{i+j}(k), \qquad 0 < \alpha < 1$$

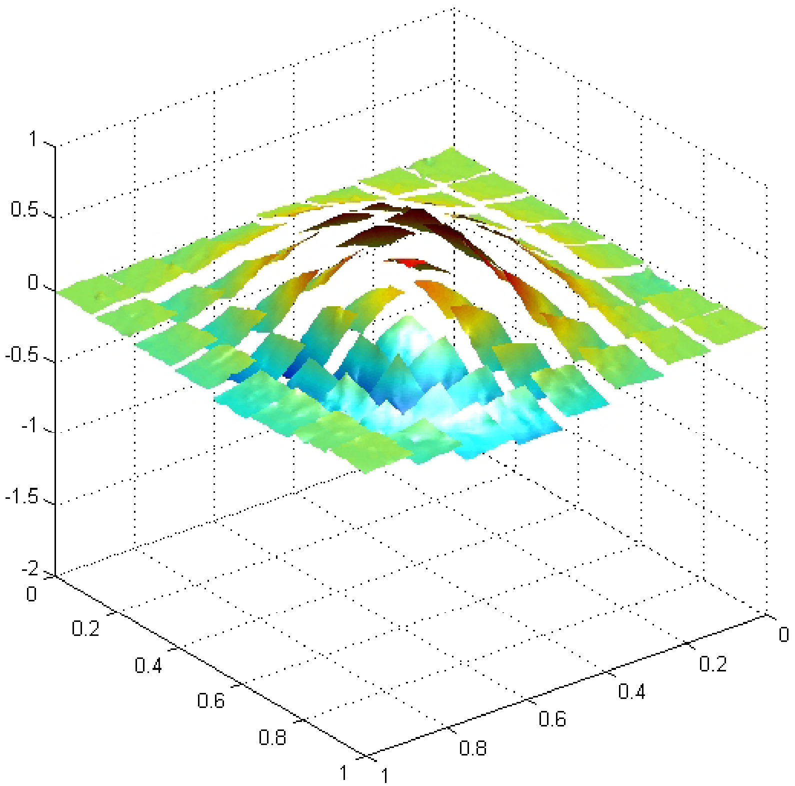
Spline coefficients are updated using dual variable $y(k+1)$ and local partition of constraint matrix A_i:

$$\hat{c}_i(k+1) = \hat{c}_i(k) - (A_i)^T y_i(k+1)$$

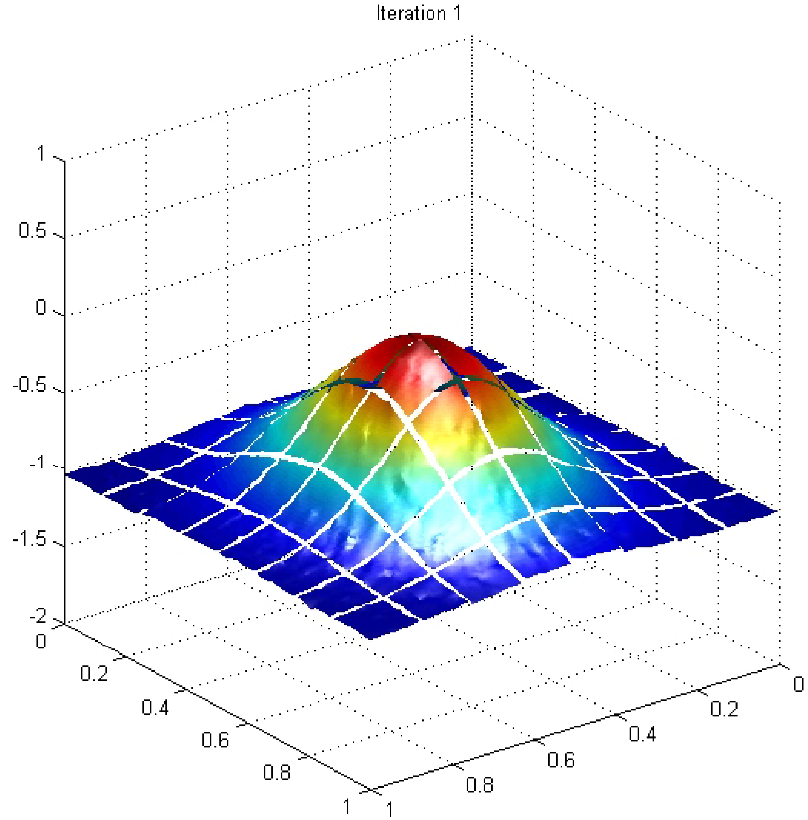
Distributed Optimization made possible by the **highly sparse structure** of the constraint matrix A!

(*) S. Boyd *al.*, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, **Foundations and Trends in Machine Learning**, 2010

Distributed-SABRE



Movie: Stage 2; distributed PME

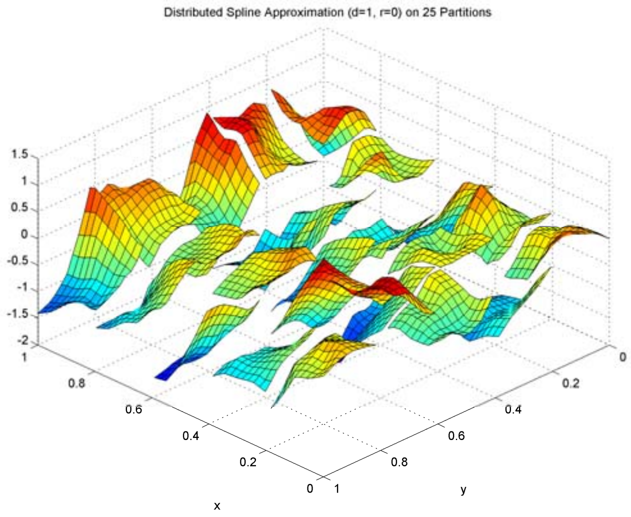


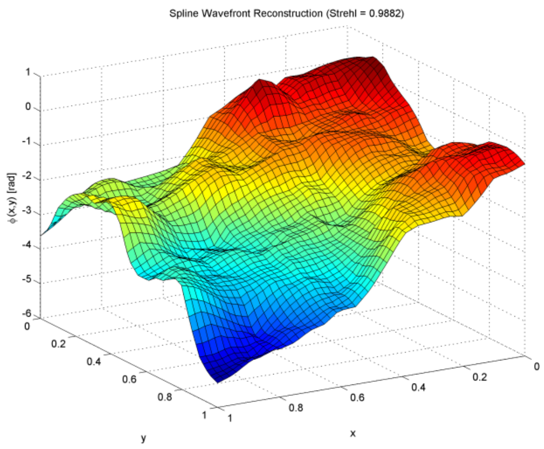
Movie: Stage 3; distributed Dual Ascent

Numerical Experiment with D-SABRE

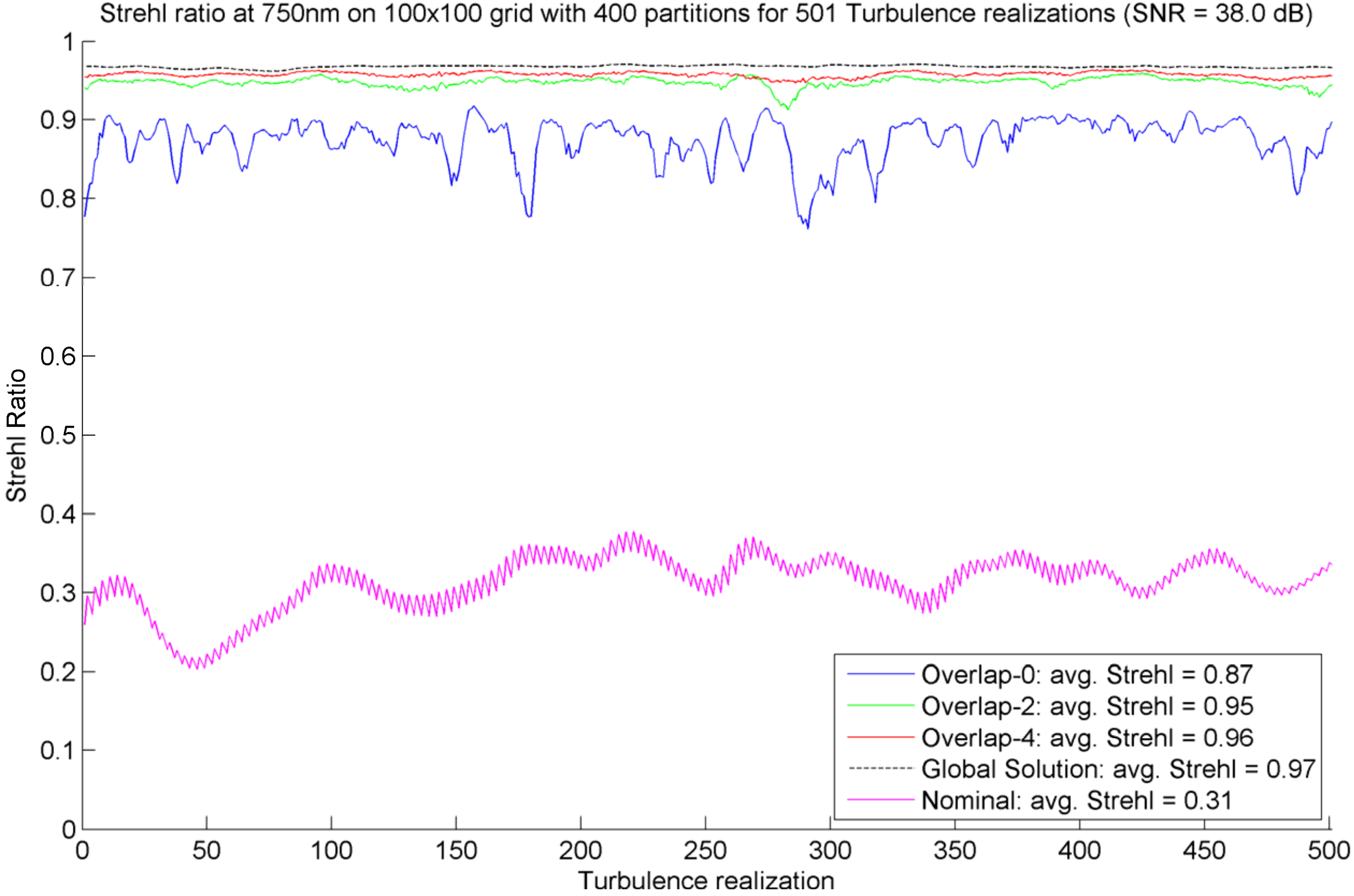
Quarter scale (100x100 sensor grid) numerical experiment setup:

- Simulated EPICS turbulence wavefronts (Strehl@750nm = 0.3+/- 0.1)
- Dynamic wavefront reconstruction using simple bi-cubic DM model
- 38 [dB] signal to noise ratio
- 500 turbulence realizations
- 100x100 sensor grid
- 400 partitions for distributed method





Numerical Experiment with D-SABRE



Computational Aspects of D-SABRE

D-SABRE compute requirements per triangulation partition per stage

Stage 1 (local wavefront reconstruction):

Matrix-Vector-Multiplication: $\hat{c}_i = Q_i \cdot s_i$

Requirement: $O(N_i^2)$

N_i = Total number of B-coefficients per partition

Stage 2 (Distributed Piston Mode Equalization)

p Vector-Add operations: $\hat{c}_i = \hat{c}_i - m$

Requirement: $O(p \cdot N_i)$

Stage 3 (Distributed Dual Ascent Smoothing)

k iterative Sparse-MVM operations: $(A_i)^T y_i(k+1),\ A_{i+j} \cdot \hat{c}_{i+j}(k)$

Requirement: $O(k \cdot N_i / E)$

Computational Aspects of D-SABRE

D-SABRE total compute requirements per triangulation partition

Stage 1+2+3:

Compute requirement: $O(N_i^2 + p \cdot N_i + k \cdot N_i / E)$

- Stage 2 iteration count p depends on the total number of simplices in a partition, Stage 3 iteration count k depends on continuity order and noise levels.
- Stage 1 (local reconstruction) is dominant if $p < N_i$ and if $k < E \cdot N_i$
- In general $p \ll N_i$, $k \ll E \cdot N_i$
- Conclusion: Stage 1 reconstruction is determining factor in compute performance!

Computational Aspects of D-SABRE

Compute budget for WFR on an ELT class system:

Sensor grid: 240x240,

Total triangles: $2*240^2$ = 115200 triangles,

Total partitions: 768, with 150 triangles per partition (includes overlap)

FLOP's per partition per cycle: $(150*3)^2$ = 202 KFLOP

FLOPS per partition for 3000Hz update rate = 3000*202e3 = **610 MFLOPS**

TOTAL FLOPS for 768 partitions = **469 GFLOPS**

Conclusion Hardware Requirement:

2 NVidia Tesla C2050 GPU's with peak DP performance 2 * 448 cores * 1 GFLOPS = 896 GFLOPS

running 1 partition per core (requires 768 cores total)

8 Intel Core i7-980 CPU's with peak DP performance 8 * 6 cores * 18 GFLOPS = 864 GFLOPS

running 18 partitions per core (requires 43 cores total)

Conclusion

- The SABRE method can locally reconstruct wavefronts on non-rectangular domains using non-linear spline functions.
- The D-SABRE method is a distributed version of the SABRE spline WFR method published in JOSA-2012; it is specifically designed for parallel operations on multi-core hardware..
- D-SABRE has all potential to perform real-time Wavefront Reconstruction at 3000Hz for the E-ELT challenges using 8 Intel Core i7-980 class CPU's, or 2 NVidia Tesla C2050 class GPU's.

Future Work

- The D-SABRE method will be implemented in a C-GPU language like CUDA or OpenCL.
- The SABRE method will be refined to enable non-linear wavefront reconstruction, and the use of non-Shack-Hartmann based wavefront sensors.
- A full scale simulation based on simulated E-ELT phase screens and operational (GPU) hardware will be created.

Thank you for your attention!