

# On distributed wavefront reconstruction for large-scale adaptive optics systems

CORNELIS C. DE VISSER,<sup>1,\*</sup> ELISABETH BRUNNER,<sup>2</sup> AND MICHEL VERHAEGEN<sup>2</sup>

<sup>1</sup>Department of Control & Simulation, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

<sup>2</sup>Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

\*Corresponding author: c.c.devisser@tudelft.nl

Received 30 September 2015; revised 3 February 2016; accepted 1 March 2016; posted 3 March 2016 (Doc. ID 251092); published 7 April 2016

The distributed-spline-based aberration reconstruction (D-SABRE) method is proposed for distributed wavefront reconstruction with applications to large-scale adaptive optics systems. D-SABRE decomposes the wavefront sensor domain into any number of partitions and solves a local wavefront reconstruction problem on each partition using multivariate splines. D-SABRE accuracy is within 1% of a global approach with a speedup that scales quadratically with the number of partitions. The D-SABRE is compared to the distributed cumulative reconstruction (CuRe-D) method in open-loop and closed-loop simulations using the YAO adaptive optics simulation tool. D-SABRE accuracy exceeds CuRe-D for low levels of decomposition, and D-SABRE proved to be more robust to variations in the loop gain. © 2016 Optical Society of America

**OCIS codes:** (010.1080) Active or adaptive optics; (010.7350) Wave-front sensing; (010.1285) Atmospheric correction; (000.3860) Mathematical methods in physics; (000.4430) Numerical approximation and analysis; (350.1260) Astronomical optics.

<http://dx.doi.org/10.1364/JOSAA.33.000817>

## 1. INTRODUCTION

In the coming decade, a new generation of extremely large-scale optical telescopes will see first light. It is well known that increasing the size of the telescope aperture is only beneficial if the adaptive optics (AO) system, which compensates for turbulence-induced wavefront aberrations, scales accordingly [1]. In particular, the total number of wavefront sensors (WFS) and deformable mirror (DM) actuators required in an AO system to obtain a given image quality is determined by the telescope diameter and the Fried coherence length [2]. To update the DM actuator commands such that the incoming aberrated wavefront is corrected, the global wavefront phase has to be reconstructed from the WFS measurements at kilohertz range frequencies.

Most conventional wavefront reconstruction (WFR) methods, like the finite difference (FD) method [2,3,4], can be formulated as a matrix vector multiplication (MVM) in which a reconstruction matrix is first constructed offline, and then multiplied online with a vector of WFS measurements to obtain the unknown wavefront phase. The computation of the reconstruction matrix in its most naive form is an  $O(N^3)$  operation, where  $N$  is the number of unknown phase samples. Applying the real-time MVM scales with  $O(N^2)$ . For the extreme-AO (XAO) system of the future European Extremely Large Telescope, the number of unknowns  $N$  is in the range of  $10^4$ – $10^5$ . Current real-time performance of standard methods fails to meet the required update frequency for such systems.

Hence, there has been a focus on improving the computational efficiency of the WFR operation.

Currently, one of the most computationally efficient zonal reconstruction methods is the cumulative reconstruction (CuRe) method, which is of complexity order between  $O(12N)$  and  $O(19N)$ , depending on the implementation [5]. Fourier domain methods are an efficient alternative to the MVM methods discussed above, because the partial differential equations that relate the wavefront slopes to the wavefront itself can be reduced to division schemes in the complex plane. The Fourier domain method presented by Poyneer *et al.* in [6] has a time efficiency of  $O(N \log N)$ . The Haar-wavelet-based reconstruction method by Hampton *et al.* reaches efficiencies of the order  $O(10N)$  [7]. In the class of iterative methods, the complexity of a multigrid preconditioned conjugate gradient algorithm presented by [8] scales with  $O(N \log N)$ . The fractal iterative method by Thiébaud and Tallon [9] provides a minimum variance solution of linear complexity with the computational cost depending on the number of iterations.

While the state of the art in high-performance WFR methods reaches linear computational complexity orders, it is important to note that these numbers are WFS-array wide, or *global*, numbers. Even with a linear complexity order, the limit in single CPU core performance will be reached at some point. The disadvantage of current wavefront reconstruction methods is that they have not been designed specifically for parallel processing architectures. While certain computational opera-

tions can be parallelized straightforwardly (e.g., MVM multiplications can easily be distributed over multiple CPU cores), their underlying construction is that of a nondistributed global method. As a direct result of this, the reconstructor can only be calculated at a single central point at which all WFS measurements must be gathered and at which the DM influence matrix is calculated. A tell-tale sign of any nondistributed global method is that the global reconstruction matrix is a dense matrix.

Recently, an extension of the CuRe algorithm [10] was proposed which combines the original line integral approach with domain decomposition to tackle the high noise propagation of the centralized algorithm. This distributed version of CuRe, called CuRe-D, scales with  $O(20N)$  and has been shown to be suitable for parallel implementation.

The main contribution of this paper is a distributed wavefront reconstruction method which is designed in particular for use in XAO systems. This new method is the distributed-spline-based aberration reconstruction (D-SABRE) method. The D-SABRE method is an extension of the recently introduced SABRE method which first used multivariate simplex B-splines to locally model wavefront aberrations on nonrectangular WFS arrays [11]. The D-SABRE is based on a decomposition of the global wavefront sensor domain into any number of triangular partitions, where each partition supports a local SABRE model which depends only on local WFS measurements.

The D-SABRE method is a two-stage method in which each stage is a distributed operation. In the first stage, a local wavefront reconstruction problem is solved in parallel on each triangular partition using local WFS slopes, resulting in a local SABRE model. Each SABRE model has an unknown piston mode, and as a result, there is no continuity between SABRE models on neighboring partitions. In the second stage of the D-SABRE, a continuous global wavefront is reconstructed by equalizing the piston modes of the local SABRE models using a new distributed piston mode equalization (DPME) algorithm. Additionally, the dual ascent method from [12] is reformulated into a new efficiently distributed form by exploiting the inherent sparseness of the D-SABRE and constraint matrices.

The advantages of the D-SABRE over current distributed wavefront reconstruction methods can be summarized as follows:

1. D-SABRE is based on a local least-squares estimates and has, in this sense, locally optimal noise rejection.
2. D-SABRE does *not* suffer from noise propagation, and the accuracy and noise resilience actually improve on a global scale as the size of the partitions increases and as the global WFS array increases in size.
3. Certain parallel hardware such as a GPU fulfill their potential speedup only for a sufficiently large computational task per processor, i.e., partitions cannot be chosen too small. This requirement is completely in line with the fact that D-SABRE accuracy and noise resilience increase with increasing partition size.
4. The D-SABRE wavefront is an analytic solution to the wavefront reconstruction problem. Without any further interpolation leading to additional approximation errors, phase estimates are available at any location in the WFS domain. This is an advantage in case of misalignments between the actuator

and subaperture arrays or if a different actuator distribution is given.

The D-SABRE currently presented should be seen as a baseline or the simplest possible version. Future versions will include higher degree (e.g., cubic) splines which will require fewer subapertures and further increase accuracy, more advanced sensor models like that presented in [13], and more advanced estimators such as the minimum variance estimator that exploit a priori knowledge of the turbulence and wavefront statistics.

This paper is outlined as follows. First, we provide brief preliminaries on the SABRE method for wavefront reconstruction in Section 2. We introduce the new D-SABRE method in Section 3 and also provide analyses of algorithm convergence and computational complexity. Additionally, we present a tutorial example of the D-SABRE in Section 3. In Section 4, we present the results from a numerical validation where we compare the D-SABRE to the global SABRE method and the distributed CuRe-D method [10] in open- and closed-loop simulations. Finally, conclusions are provided in Section 5.

## 2. PRELIMINARIES ON THE SABRE METHOD FOR WAVEFRONT RECONSTRUCTION

The D-SABRE method is an extension of the recently introduced SABRE method for wavefront reconstruction. In order to aid the reader in the understanding of the theory, preliminaries on the SABRE method will be provided in this section. The SABRE was first introduced in [11], to which we refer for a more in-depth coverage of the matter.

### A. Wavefront Reconstruction from Slopes and Curvature

The relationship between the slopes of the wavefront phase and the wavefront phase can be described in the form of the following system of first-order partial differential equations [3]:

$$\sigma_x(x, y) = \frac{\partial \phi(x, y)}{\partial x}, \quad (1a)$$

$$\sigma_y(x, y) = \frac{\partial \phi(x, y)}{\partial y}, \quad (1b)$$

with  $\phi(x, y)$  as the unknown wavefront, and with  $\sigma_x(x, y)$  and  $\sigma_y(x, y)$  as the wavefront slopes at location  $(x, y)$  in the directions  $x$  and  $y$ , respectively.

### B. SABRE Method on a Single Triangle

The local model elements of a SABRE model are defined on individual triangles, rather than on the rectangular elements used by FD methods. In [11] it is shown that on a single triangle, denoted  $t$ , the wavefront phase  $\phi(x, y)$  is approximated with a SABRE model of degree  $d$  as follows:

$$\phi(x, y) \approx \mathbf{B}^d(b(x, y)) \cdot \mathbf{c}^t, \quad d \geq 1, \quad (x, y) \in t, \quad (2)$$

with  $\mathbf{B}^d(b(x, y))$  as the vector of basis polynomials and with  $\mathbf{c}^t$  as the vector of B-coefficients. The values in the vector of basis polynomials  $\mathbf{B}^d(b(x, y))$  depend only on the geometry of the sensor array and the polynomial degree  $d$ . Therefore, at any given time instant, the particular form of the SABRE model is determined by the B-coefficients  $\mathbf{c}^t$ . Consequently, wavefront reconstruction with the SABRE essentially consists of estimat-

ing the values of the B-coefficients given a set of WFS measurements.

It is shown in [11] that Eq. (2) leads to the following SABRE slope sensor model on a single triangle:

$$\sigma_x(x, y) = \frac{d!}{(d-1)!} \mathbf{B}^{d-1}(b(x, y)) \mathbf{P}^{d, d-1}(\mathbf{a}_x) \cdot \mathbf{c}^t + n_x(x, y), \quad (3a)$$

$$\sigma_y(x, y) = \frac{d!}{(d-1)!} \mathbf{B}^{d-1}(b(x, y)) \mathbf{P}^{d, d-1}(\mathbf{a}_y) \cdot \mathbf{c}^t + n_y(x, y), \quad (3b)$$

with  $d \geq 1$  as the degree of the SABRE model,  $\mathbf{B}^{d-1}(b(x, y))$  as the basis polynomials of degree  $d-1$ ,  $\mathbf{c}^t$  as the vector of B-coefficients from Eq. (2), and  $n_x(x, y)$  and  $n_y(x, y)$  in Eq. (3) as the residual terms which contain both sensor noise and modeling errors.

The matrices  $\mathbf{P}^{d, d-1}(\mathbf{a}_x)$  and  $\mathbf{P}^{d, d-1}(\mathbf{a}_y)$  in Eq. (3) are the *de Casteljau matrices* in the (Cartesian) directions  $\mathbf{a}_x$  and  $\mathbf{a}_y$  which are essential to the SABRE method [14]. In essence, the de Casteljau matrices allow a reformulation of the PDEs from Eq. (1) into a set of algebraic equations in terms of the B-coefficients  $\mathbf{c}^t$ .

A number of possible sensor geometries were introduced in [11]. In this work we use the Type-I and Type-II sensor geometries, which should be seen as baseline geometries. Recently, a more advanced sensor model for the SABRE was introduced by Guo *et al.* in [13]. This sensor model is better suited for use with real-world SH sensors which provide a spatial average of the wavefront slopes instead of point-wise local spatial derivatives of the wavefront.

### C. SABRE Method on a Complete Triangulation

The SABRE method can be used with large-scale wavefront sensor arrays by combining any number of triangles into a triangulation. The full-triangulation, or *global*, SABRE model has a predefined continuity order  $r$  between the local models which means that the  $r$ th-order directional derivatives of neighboring local SABRE models match exactly on the triangle edges. In [11] it is shown that the wavefront phase can be approximated at any point  $(x, y)$  in the WFS domain with a SABRE model as follows:

$$\phi(x, y) \approx \mathbf{B}^d \mathbf{c}, \quad (4)$$

with  $\mathbf{B}^d$  as the global B-form regression matrix and  $\mathbf{c}$  as the global vector of B-coefficients [11]. Given Eq. (4), the global WFR problem is constructed from rows of the form Eq. (3) as follows:

$$\sigma = \mathbf{Dc} + \mathbf{n}, \quad (5a)$$

$$0 = \mathbf{Ac}, \quad (5b)$$

with  $\sigma = [\sigma_x^T \sigma_y^T]^T \in \mathbb{R}^{2K \times 1}$  as the vector of measured wavefront slopes,  $\mathbf{n}$  as a residual noise vector, and  $\mathbf{A}$  as the global constraint matrix. The spline regression matrix  $\mathbf{D}$  in Eq. (5a) is constructed as follows:

$$\mathbf{D} = d \mathbf{B}^{d-1} \mathbf{P}_u^{d, d-1}, \quad (6)$$

with  $\mathbf{B}^{d-1}$  as the global basis function matrix of degree  $d-1$  and  $\mathbf{P}_u^{d, d-1}$  as the full-triangulation de Casteljau matrix which is constructed as shown in [11]. Note the slight difference in notation of  $\mathbf{D}$  in Eq. (6) compared to that presented in [11]; it will become apparent in following sections why this change was made.

The global constraint matrix  $\mathbf{A}$  in Eq. (5b) is constructed as follows:

$$\mathbf{A} := \begin{bmatrix} \mathbf{H} \\ \mathbf{h} \end{bmatrix} \in \mathbb{R}^{(EV+1) \times j\hat{d}}, \quad (7)$$

with  $\mathbf{H} \in \mathbb{R}^{(EV) \times j\hat{d}}$  as the full-rank *smoothness matrix*. The vector  $\mathbf{h} = [1 \ 0 \ \dots \ 0] \in \mathbb{R}^{1 \times j\hat{d}}$  is the *anchor vector* which was first introduced in [11], which is used to fix the piston mode (the unknown integration constant) to a predetermined constant.

For low-degree ( $d \leq 2$ ) basis polynomials, the resulting system in Eq. (5) will be fully determined given a Type-I or Type-II SH sensor geometry and the constraints in Eq. (7). However, when moving toward higher degree ( $d \geq 3$ ) basis polynomials Eq. (5) will be underdetermined and will not lead to a unique solution. Future work on obtaining higher degree D-SABRE models will therefore be focused on imposing specific (e.g., “do-nothing” boundary conditions [15]) boundary conditions on the external edges of the D-SABRE submodels.

In [11] it was shown that the global wavefront reconstruction problem can be formulated as an equality constrained least-squares optimization problem,

$$\operatorname{argmin} \frac{1}{2} \|\sigma - \mathbf{Dc}\|_2^2 \quad \text{subject to } \mathbf{Ac} = 0, \quad (8)$$

with  $\sigma$  as the slopes from the wavefront sensor,  $\mathbf{D}$  from Eq. (6),  $\mathbf{c}$  as the global vector of B-coefficients, and  $\mathbf{A}$  as the smoothness matrix from Eq. (7).

This constrained optimization problem can be reduced into an unconstrained problem by introducing the Lagrangian for Eq. (8) as

$$\mathcal{L}(\mathbf{c}, \mathbf{y}) = \frac{1}{2} \|\sigma - \mathbf{Dc}\|_2^2 + \mathbf{y}^T \mathbf{Ac}, \quad (9)$$

with  $\mathbf{y}$  as a vector of Lagrangian multipliers.

The minimum of Eq. (9) is

$$\frac{\partial \mathcal{L}(\mathbf{c}, \mathbf{y})}{\partial \mathbf{c}} = -\mathbf{D}^T(\sigma - \mathbf{Dc}) + \mathbf{A}^T \mathbf{y} \stackrel{!}{=} 0, \quad (10)$$

with which the following B-coefficient estimator is derived:

$$\hat{\mathbf{c}} = (\mathbf{D}^T \mathbf{D})^{-1} (\mathbf{D}^T \sigma - \mathbf{A}^T \mathbf{y}). \quad (11)$$

The problem with Eq. (11) is that the undefined piston mode causes  $\mathbf{D}^T \mathbf{D}$  to be rank deficient and therefore not invertible. In [11] this issue was solved by immediately projecting the spline regressors on the nullspace of the constraint matrix  $\mathbf{A}$ .

Here we require a more explicit formulation of

$$\begin{aligned} \hat{\mathbf{c}} &= \mathcal{N}_A (\mathcal{N}_A^T \mathbf{D}^T \mathbf{D} \mathcal{N}_A)^{-1} \mathcal{N}_A^T \mathbf{D}^T \sigma \\ &= \mathbf{R}_N \mathbf{D}^T \sigma \\ &= \mathbf{Q}_N \sigma, \end{aligned} \quad (12)$$

with  $\mathcal{N}_A$  as a basis for *null*( $\mathbf{A}$ ),  $\mathbf{R}_N = \mathcal{N}_A (\mathcal{N}_A^T \mathbf{D}^T \mathbf{D} \mathcal{N}_A)^{-1} \mathcal{N}_A^T$ , and  $\mathbf{Q}_N = \mathbf{R}_N \mathbf{D}^T$  as the SABRE matrix from [11].

### 3. DISTRIBUTED WFR WITH SIMPLEX B-SPLINES

In this section, the D-SABRE distributed wavefront reconstruction method is introduced. The D-SABRE consists of 2 stages: Stage-1, which is the distributed local WFR stage, and Stage-2, which is the combined distributed piston mode

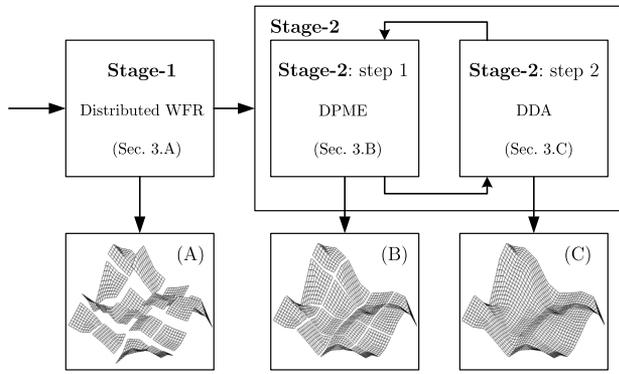


Fig. 1. Outline of the D-SABRE algorithm.

equalization and distributed dual ascent post-smoothing stage. A schematic of the D-SABRE algorithm is shown in Fig. 1.

### A. D-SABRE Stage-1: Distributed Local WFR

The first stage of D-SABRE involves the decomposition of the global WFR problem from Eq. (8) into a set of local subproblems. For this we make use of the domain decomposition method from [16]. First, we decompose the global triangulation  $\mathcal{T}$  into a set of  $G$  subtriangulations as follows:

$$\mathcal{T} = \bigcup_{i=1}^G \mathcal{T}_i, \quad (13)$$

with each  $\mathcal{T}_i$  containing  $J_i$  triangles. Every subtriangulation  $\mathcal{T}_i$  in turn consists of two parts: a core part  $\Omega_i$  and an overlap part  $\Xi_i$  as

$$\mathcal{T}_i = \Omega_i \cup \Xi_i, \quad \Omega_i \cap \Xi_i = \emptyset, \quad (14)$$

with  $\Omega_i$  containing  $J_{\Omega_i}$  triangles and  $\Xi_i$  containing  $J_{\Xi_i}$  triangles such that  $J_i = J_{\Omega_i} + J_{\Xi_i}$ . The purpose of  $\Xi_i$  is to overlap neighboring subtriangulation core parts  $\mathcal{T}_j$  in order to increase numerical continuity between neighboring partitions; see Fig. 2. In this paper, we use the term “overlap level” (OL) to define the size of  $\Xi_i$ . The OL is a scalar, which determines how many layers of simplexes from the core partition are included in  $\Xi_i$ ; see Fig. 2.

We can now introduce the *overlap overhead*  $\rho$  as follows:

$$\rho = \frac{J_i}{J_{\Omega_i}}, \quad \rho \geq 1, \quad (15)$$

with  $\rho = 1$  indicating no overlap. It is not trivial to provide a relation between overlap level and overlap overhead because it strongly depends on the geometry of the triangulation and on the location of a partition within the global triangulation. Nevertheless, for a simplex Type-I geometry, the minimum and maximum  $\rho$  can easily be determined using basic geometry rules:

$$\rho_{\text{Type-I}} = \begin{cases} 1 + \frac{\text{OL}(6\text{OL}+8)\sqrt{J_{\Omega_i}/2}}{J_{\Omega_i}} & (\text{max overhead}), \\ 1 + \frac{\text{OL}(\text{OL}+4)\sqrt{J_{\Omega_i}/2}}{J_{\Omega_i}} & (\text{min overhead}), \end{cases} \quad (16)$$

with  $\text{OL} \geq 0$  as the overlap level.

We now assume that the wavefront can be approximated locally on each subtriangulation  $\mathcal{T}_i$  as follows:

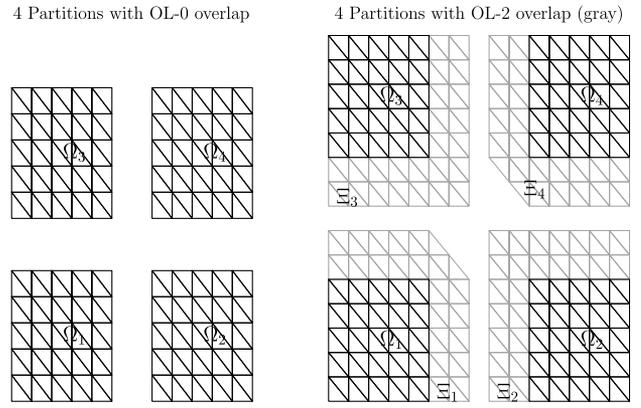


Fig. 2. OL-0 partitioning without overlap (left) and an OL-2 partitioning with 2 levels of overlap between partitions (right) using the same initial triangulation containing 200 triangles. In this case,  $J_{\Omega_i} = 50$  for all partitions, while  $J_{\Xi_1}, J_{\Xi_4} = 44$  and  $J_{\Xi_2}, J_{\Xi_3} = 48$  for the OL-2 partitioning.

$$\phi_i(x, y) \approx s_{r_i}^d(x, y) = \mathbf{B}_i^d \mathbf{c}_i, \quad 1 \leq i \leq G, \quad (17)$$

with  $\phi_i(x, y)$  as the local wavefront phase,  $s_{r_i}^d(x, y)$  as a D-SABRE partition,  $\mathbf{B}_i^d$  as the local matrix of B-form regressors, and  $\mathbf{c}_i$  as the set of local B-coefficients.

The goal now is to determine  $\mathbf{c}_i$  for all  $G$  local models. For each subtriangulation  $\mathcal{T}_i$ , we can decompose Eq. (8) into a set of local subproblems as follows:

$$\text{argmin} \|\sigma_i - \mathbf{D}_i \mathbf{c}_i\|_2^2 \quad \text{subject to } \mathbf{A}_i \mathbf{c}_i = 0, \quad 1 \leq i \leq G, \quad (18)$$

with  $\sigma_i$  as local WFS slopes,  $\mathbf{A}_i$  as a *local* constraint matrix, and  $\mathbf{D}_i$  as the local version of Eq. (6) as follows:

$$\mathbf{D}_i = d \mathbf{B}_i^{d,d-1} \mathbf{P}_{\mathbf{u}_i}^{d,d-1}, \quad (19)$$

with  $\mathbf{P}_{\mathbf{u}_i}^{d,d-1}$  as the local de Casteljau matrix.

Note that the local constraint matrix  $\mathbf{A}_i$  cannot be obtained by decomposing the global constraint matrix  $\mathbf{A}$  into  $G$  blocks because it is not block diagonal. Instead,  $\mathbf{A}_i$  is constructed from a new local smoothness matrix  $\mathbf{H}_i$  and a new local anchor constraint  $h_i$  as

$$\mathbf{A}_i := \begin{bmatrix} \mathbf{H}_i \\ \mathbf{h}_i \end{bmatrix} \in \mathbb{R}^{(E_i V_i + 1) \times J_i \hat{d}}, \quad (20)$$

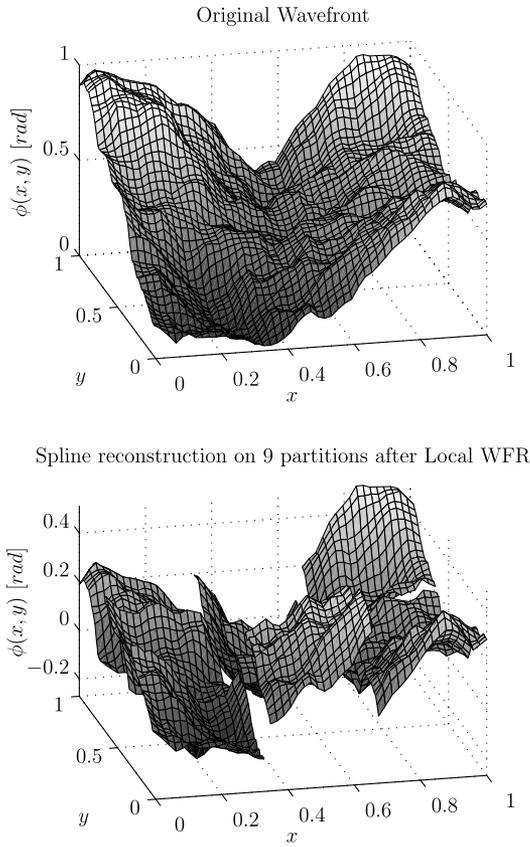
where it is important to note that  $\mathbf{H}_i$  does *not* contain any smoothness conditions linking a partition  $i$  to any other partition. For the anchor vector, we have  $\mathbf{h}_i = [1 \ 0 \ \dots \ 0] \in \mathbb{R}^{1 \times J_i \hat{d}}$ .

The local WFR problem in Eq. (18) can be solved in the same fashion as Eq. (12):

$$\begin{aligned} \hat{\mathbf{c}}_i^{\text{loc}} &= \mathcal{N}_{A_i} (\mathcal{N}_{A_i}^T \mathbf{D}_i^T \mathbf{D}_i \mathcal{N}_{A_i})^{-1} \mathcal{N}_{A_i}^T \mathbf{D}_i^T \sigma_i \\ &= \mathbf{R}_{N_i} \mathbf{D}_i^T \sigma_i \\ &= \mathbf{Q}_{N_i} \sigma_i, \end{aligned} \quad (21)$$

with  $\mathcal{N}_{A_i}$  as a basis for *null* ( $\mathbf{A}_i$ ) and  $\mathbf{R}_{N_i} = \mathcal{N}_{A_i} (\mathcal{N}_{A_i}^T \mathbf{D}_i^T \mathbf{D}_i \mathcal{N}_{A_i})^{-1} \mathcal{N}_{A_i}^T$ .

Note that Eq. (21) does not depend on information from any other partition, and as a result, each of the  $G$  local reconstruction problems can be solved in parallel. In Fig. 3,



**Fig. 3.** Original wavefront (top). The D-SABRE model after completion of the distributed local reconstruction stage (bottom).

an example is shown of the results of the first distributed stage of the D-SABRE. Clearly, the local models are disconnected and do not accurately approximate the global wavefront. This is a direct result of the anchor constraint in each partition being independent from that of neighboring partitions.

### B. Distributed Piston Mode Equalization

In order to equalize the phase offsets (i.e., unknown integration constants) between neighboring D-SABRE partitions, a process called piston mode equalization (PME) is introduced. PME effectively resolves the difference between the estimated per-partition (unknown) integration constants. PME offsets an entire D-SABRE partition with a single constant  $k_i$  such that the maximum phase offset between it and a neighboring partition is minimized:

$$\tilde{s}_{r_i}^d(x, y) = \mathbf{B}_i^d \mathbf{c}_i + k_i, \quad 1 \leq i \leq G, \quad k_1 = 0, \quad (22)$$

where the tilde indicates that an offset  $k_i$  as been applied to  $s_{r_i}^d(x, y)$ .

The first partition (i.e.  $s_{r_1}^d(x, y)$ ) has  $k_1 = 0$  per definition. This partition is indicated as the *master partition*, relative to which all other partitions are equalized. The D-SABRE user is free to choose the location of the master partition, but a smart choice is a partition that is located as close as possible to the center of the global triangulation.

In order to calculate  $k_i$ , a definition of the concept of neighboring partitions is required. For this, we define the neighbors

of  $s_{r_i}^d(x, y)$  as all partitions  $s_{r_m}^d(x, y)$  with  $1 \leq m \leq G$  which share at least one vertex  $v_{i,m}$  in the core parts of their triangulations. We then define  $\mathcal{M}_i$  as the set that contains the indices of all neighbors of partition  $i$ , with  $G_i = |\mathcal{M}_i|$  the total number of neighbor partitions.

The phase offset between partition  $i$  and  $m$  is calculated from the phase offset at the shared vertex  $v_{i,m}$ . A unique property of the simplex B-splines is that spline function value on a vertex is equal to the value of the B-coefficient located at that vertex [11,17]. On the shared vertex  $v_{i,m}$ , we have  $s_{r_i}^d(v_{i,m}) = \mathbf{c}_{\Omega_i}$  and  $s_{r_m}^d(v_{i,m}) = \mathbf{c}_{\Omega_m}$ . Hence, the estimated phase offset  $\hat{k}_{i,m}$  between partition  $i$  and  $m$  is

$$\hat{k}_{i,m} = \mathbf{c}_{\Omega_i} - \mathbf{c}_{\Omega_m}, \quad m \in \mathcal{M}_i, \quad (23)$$

Note that in the presence of noise, the estimate of  $k_{i,m}$  can easily be improved by taking the average of the phase offsets at any number of shared vertices. If partition  $i$  has a total of  $G_i$  direct neighbors, the PME constant  $k_i$  for partition  $i$  is determined as follows:

$$k_i = \max\{\hat{k}_{i,m}\}, \quad \forall (m > i) \in \mathcal{M}_i, \quad (24)$$

where it should be noted that  $k_i$  is the *maximum* offset between partition  $i$  and  $m$  with  $m > i$ ; this asymmetry is required for the PME operation to converge.

In the form of Eq. (23), PME is a sequential operation. However, it is straightforward to modify Eq. (23) into a distributed consensus problem form we indicate as DPME:

$$\begin{aligned} \hat{k}_{i,m}(l+1) &= \mathbf{c}(l)_{\Omega_{i,m}} - \mathbf{c}(l)_{\Omega_{m,i}}, \quad m \in \mathcal{M}_i, \\ l &= 1, 2, \dots, L, \end{aligned} \quad (25a)$$

$$\begin{aligned} k_i(l+1) &= \max\{\hat{k}_{i,m}(l+1)\}, \quad \forall (m > i) \in \mathcal{M}_i, \\ l &= 1, 2, \dots, L, \end{aligned} \quad (25b)$$

which converges when  $l = L$ .

In Fig. 4, the concept of distributed piston mode equalization (DPME) is demonstrated. In essence, each partition continuously adapts its offset based on data obtained from its direct neighbors which are themselves continuously adapting their offsets.

At each iteration, the set  $\mathcal{K}_i^{(l)}(m)$  combines all  $G_i$  phase offsets between partition  $i$  and partitions  $m$  for iteration  $l$ :

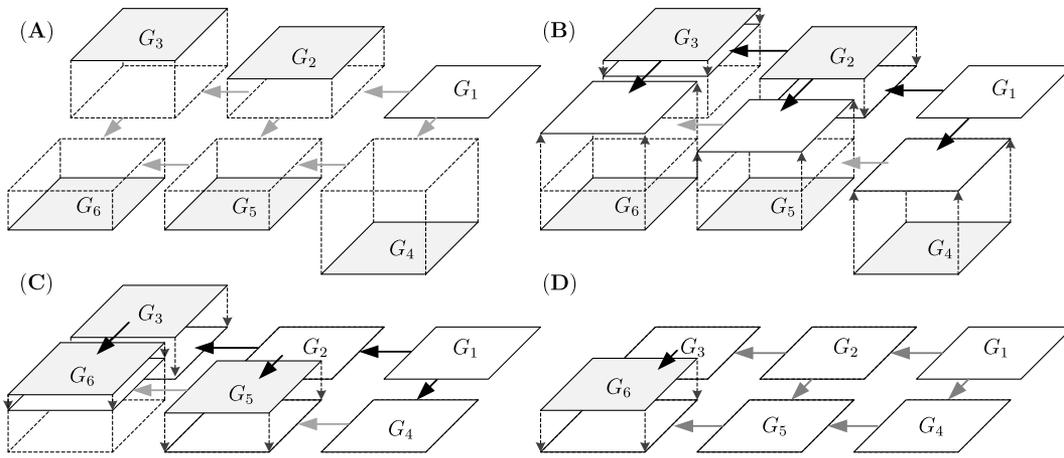
$$\mathcal{K}_i^{(l)}(m) = \{\hat{k}_{i,m}(l)\}, \quad 1 \leq i \leq G, \quad \forall m \in \mathcal{M}_i. \quad (26)$$

This set will prove to be instrumental during the distributed dual ascent stage of D-SABRE.

In Fig. 5, the DPME method has been applied to the discontinuous reconstructed wavefront from Fig. 3.

### C. Distributed Post-Smoothing

DPME minimizes the unknown phase offsets between local D-SABRE partitions. In general, this does not result in a smooth reconstruction of the wavefront, especially in the presence of (sensor) noise. Smoothness can be improved by increasing the overlap level as discussed in Section 3.A but only up to some point and at the cost of reduced computational efficiency. In order to obtain a smooth reconstruction, a distributed dual ascent (DDA) method is used to enforce smoothness between local D-SABRE partitions. The derivation of the DDA



**Fig. 4.** (A)–(D) Four-step DPME operation on 6 partitions, with  $G_1$  as the master partition. Gray and black arrows indicate information flow, with black arrows as the actual information used in a DMPE step.

smoother starts with the decomposition of the global Lagrangian from Eq. (9) into sub-Lagrangians according to [12]

$$\mathcal{L}(\mathbf{c}, \mathbf{y}) = \sum_{i=1}^G \mathcal{L}_i(\mathbf{c}_i, \mathbf{y}) = \sum_{i=1}^G \left( \frac{1}{2} \|\sigma_i - \mathbf{D}_i \mathbf{c}_i\|_2^2 + \mathbf{y}^\top \mathbf{G}_i \mathbf{c}_i \right), \quad (27)$$

with  $\mathbf{y}$  as the *global* vector of Lagrange duals,  $\mathbf{c}_i$  as a subvector of local B-coefficients, and  $\mathbf{G}_i$  as a naive partitioning of the global constraint matrix  $\mathbf{A}$ :

$$\mathbf{A} = [\mathbf{G}_1 \quad \mathbf{G}_2 \quad \cdots \quad \mathbf{G}_G]. \quad (28)$$

Dual decomposition of Eq. (27) leads to the following algorithm [12]:

$$\mathbf{c}_i(l+1) = \operatorname{argmin}_{\mathbf{c}_i} \mathcal{L}_i(\mathbf{c}_i, \mathbf{y}(l)), \quad (29a)$$

$$\mathbf{y}(l+1) = \mathbf{y}(l) + \alpha(l) \mathbf{A} \mathbf{c}(l), \quad (29b)$$

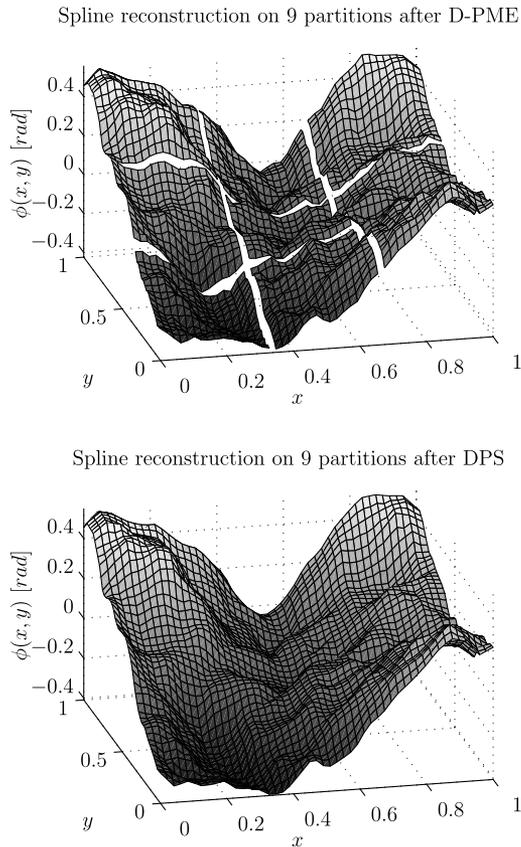
with  $\alpha(l)$  as an iteration-dependent constant and  $l$  as the dual decomposition iterator.

The dual update step in Eq. (29b) clearly is a centralized operation as it requires a gathering and then broadcasting of global variables (i.e.,  $\mathbf{y}(l)$  and  $\mathbf{c}(l)$ ).

By exploiting the sparseness structure of the global constraint matrix  $\mathbf{A}$ , a more efficient formulation can be obtained. The structure of  $\mathbf{A}$  is as follows [18,19]:

$$\begin{bmatrix} \mathbf{A}_{1,1} & 0 & 0 & \cdots & \mathbf{A}_{1,*} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{2,2} & 0 & \cdots & \mathbf{A}_{2,*} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \mathbf{A}_{3,1} & 0 & 0 & \cdots & 0 & \mathbf{A}_{3,*} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & & \ddots & & & & & & & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \mathbf{A}_{EQ,*} & 0 & \cdots & \mathbf{A}_{EQ,J} \\ \mathbf{h} & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \vdots \\ \mathbf{c}_J \end{bmatrix} = 0, \quad (30)$$

with each block  $\mathbf{A}_{qj} \in \mathbb{R}^{Q \times \hat{d}}$  containing all the smoothness conditions for a single edge  $q$  and triangle  $j$ . Note that for each



**Fig. 5.** D-SABRE model after completion of the DPME stage, with remaining discontinuities exaggerated (top); see also Visualization 1. D-SABRE model after completion of the distributed post-smoothing stage (bottom); see also Visualization 2.

edge  $q$ , there are exactly two blocks  $\mathbf{A}_{qj}$ : one for each triangle  $j$  containing the mutual edge  $q$ .

By exploiting the sparseness of  $\mathbf{A}$ , we can derive two new submatrices  $\mathbf{H}_i$  and  $\mathbf{H}_{i,M}$  for each partition  $i$ . The submatrix  $\mathbf{H}_i$  contains all smoothness conditions that define continuity inside partition  $i$ . The submatrix  $\mathbf{H}_{i,M}$ , on the other hand, contains all smoothness conditions that govern not only continuity inside partition  $i$  but also all continuity between partition  $i$  and its neighbors. In Appendix A, the details of both decompositions are presented.

Using the submatrices  $\mathbf{H}_i$  and  $\mathbf{H}_{i,M}$ , a local distributed version of Eq. (29) is obtained as

$$\mathbf{c}_i(l+1) = \hat{\mathbf{c}}_i^{\text{loc}} + \mathbf{R}_{N_i} \mathbf{H}_i^T \mathbf{y}_i(l), \quad (31a)$$

$$\mathbf{y}_i(l+1) = \mathbf{y}_i(l) + \alpha(l) \mathbf{H}_{i,M} \mathbf{c}_{i,M}(l+1), \quad (31b)$$

with  $\hat{\mathbf{c}}_i^{\text{loc}}$  as the locally estimated B-coefficients from Eq. (21), which during the iteration are constant,  $\mathbf{R}_{N_i}$  as the matrix from Eq. (21), and  $\mathbf{y}_i$  as the local dual vector. The dual update step in Eq. (31b) uses  $\mathbf{c}_{i,M}$ , which is the vector of all B-coefficients in  $\mathcal{S}_{r_m}^d$  as well as all B-coefficients in  $\mathcal{S}_{r_m}^d$  that are subject to continuity conditions on  $\mathcal{T}_i \cap \mathcal{T}_m \forall m \in \mathcal{M}_i$ .

The resulting method is fully distributed in the sense that each model partition only shares information with its direct neighboring model partitions.

### D. D-SABRE Stage-2: Merging DPME with DDA Smoothing

Both the DPME and DDA operations introduced in the previous sections are iterative. These operations could be executed sequentially, starting with DPME to remove the unknown phase offsets, and ending with DDA to post-smooth the wavefront. This would, however, defeat the purpose of D-SABRE as a distributed WFR method because it would require all partitions to complete the DPME operation before starting the DDA operation. The reason for this is that if DDA is naively merged with DPME, the DDA smoother will blend neighboring partitions regardless of convergence of the DPME operation. This results in significant undesirable and unphysical artifacts in the reconstructed wavefront.

The solution to this is a new concept indicated as virtual DPME (V-DPME). The idea is that even though DPME has not yet converged, each partition has at all times exact information on the phase offset between it and its direct neighbors. In V-DPME, this information is used to virtually minimize all piston modes between a partition  $i$  and its  $G_i$  neighbors. After V-DPME, the DDA algorithm can safely smooth the locally piston-mode-equalized partitions without introducing artifacts into the reconstructed wavefront.

The information required for V-DPME is obtained during the DPME operation in the form of the set  $\mathcal{K}_i^{(l)}(m)$  from Eq. (26) which contains all phase offsets between partition  $i$  and its neighbors. For V-DPME, a new vector of virtual offsets is created from elements of  $\mathcal{K}_i^{(l)}(m_j)$ :

$$\mu_{i,M}(l) = \left[ 0_{1 \times J_{\Omega_i}} \hat{\mathcal{K}}_i^{(l)}(m_1) \cdot 1_{1 \times J_{m_1}} \hat{\mathcal{K}}_i^{(l)}(m_2) \cdot 1_{1 \times J_{m_2}} \hat{\mathcal{K}}_i^{(l)} \cdots \right]^T$$

$$1 \leq i \leq G, \quad \forall m_j \in \mathcal{M}_i, \quad (32)$$

with  $J_{m_j}$  as the total number of triangles in  $\Omega_i \cap \Omega_{m_j}$ .

Substitution of Eq. (32) into Eq. (31b), together with the DPME from Eq. (25b), results in the final formulation of Stage-2 of the D-SABRE:

$$\mathbf{c}_i(l+1) = \hat{\mathbf{c}}_i^{\text{loc}} + \mathbf{R}_{N_i} \mathbf{H}_i^T \mathbf{y}_i(l), \quad (33a)$$

$$\mathbf{y}_i(l+1) = \mathbf{y}_i(l) + \alpha(l) \mathbf{H}_{i,M} (\mathbf{c}_{i,M}(l+1) + \mu_{i,M}(l)), \quad (33b)$$

which converges when  $l = L$ , after which time the actual PME value is added, resulting in the final B-coefficient estimate:

$$\mathbf{c}_i = \mathbf{c}_i(L) + k_i(L). \quad (34)$$

### E. Algorithm Convergence

The DPME and DDA steps of D-SABRE Stage-2 are both iterative in nature. In this section, we provide an analysis of convergence of Stage-2 as a whole. It is important to note that all parallel operations in Stage-2 are allowed to run asynchronously, i.e., the parallel processors do not require a global clock-tick.

However, optimal convergence of the DPME operation does depend on synchronicity between neighboring partitions in the sense that failure of one parallel processor to calculate an initial local WF reconstruction in Stage-1 or an accurate PME update in Stage-2 at a specific global clock-tick will influence all partitions that through interpartition continuity depend on these results. The DDA step, on the other hand, can be run completely asynchronously, because V-DPME negates any interpartition offsets. For this analysis, we will assume that local wavefront reconstruction and DPME iterations run synchronously on the parallel processors.

DPME convergence depends solely on the maximum distance, counted in number of partitions, between the master partition and any other partition. If this distance is  $R$ , then DPME converges in exactly  $R$  iterations because it takes at most  $R$  steps for information from the master partition to reach the partition furthest away from the master partition. Each step requires a single DPME iteration leading to exactly  $R$  iterations. Note that the “max” operation in Eq. (25b) only influences the path the information follows and not the total number of steps.

We base our analysis of convergence of the DDA step on the theory presented in [12]. First, let  $\varepsilon^*$  be the optimal value of the optimization problem in Eq. (18):

$$\varepsilon_i^* = \min \|\sigma_i - \mathbf{D}_i \mathbf{c}_i\|_2^2 \quad \text{subject to } \mathbf{A}_i \mathbf{c}_i = 0, \quad 1 \leq i \leq G. \quad (35)$$

Also, let

$$r_i(l+1) = \mathbf{A}_i \mathbf{c}_i(l+1) \quad (36)$$

be the primal residual at iteration  $l+1$ .

Then [12] proves that

$$\varepsilon_i^* - \varepsilon_i(l+1) \leq (y^*)^T r_i(l+1), \quad l \rightarrow \infty, \quad (37)$$

with  $\varepsilon_i(l+1)$  as the optimal value of Eq. (18) at iteration  $l+1$  and  $y^*$  as the optimal Lagrange dual. The right-hand side of Eq. (37) approaches zero as  $l \rightarrow \infty$  since  $r_i(l)$  approaches zero when the constraints are met. Therefore, we have  $\lim_{l \rightarrow \infty} \varepsilon_i(l+1) = \varepsilon_i^*$ , implying convergence.

In practice, we find that our DDA algorithm converges in at most a few tens of iterations, with adequate continuity achieved

in  $L = 10$  iterations, which agrees with the rules of thumb provided in [12]. In this case, the norm of the residual of the smoothness constraints [see Eq. (7)] is  $\|\mathbf{H}\mathbf{c}\|_2^2 < 1e-3$  for  $\text{SNR} \geq 0$  dB and reaches  $\|\mathbf{H}\mathbf{c}\|_2^2 < 1e-8$  for  $\text{SNR} \geq 20$  dB.

We also find that initializing the DDA iteration with a smooth initial estimate for  $\hat{\mathbf{c}}_i^{\text{loc}}$  significantly improves convergence of the DDA step. Finally, the rate of convergence of the DDA stage depends strongly on the value of  $\alpha(L)$  in Eq. (33). In our algorithm, a value of  $\alpha(L) = 0.5$  leads to fast convergence.

### F. Computational Complexity Analysis

Key to the utility of the D-SABRE method is its computational performance. In this section, the results from a theoretical analysis of computational complexity of the D-SABRE method are presented. The scope of this analysis is limited to theoretically required computed performance per parallel processor. Specific hardware-dependent issues such as transport latency, cache size, and available instruction sets are not included in this analysis. In our analysis, we focus on the real-time reconstruction operation and not on algorithm initialization. In the analysis, we indicate  $R$  and  $L$  as the total number of iterations in, respectively, the DPME and DDA steps of Stage-2. We compare our results to that of the global SABRE method. The results are summarized in Table 1.

For the global SABRE, we find for the complexity of the matrix–vector operation in Eq. (12):

$$U_{\text{SABRE}} = \mathcal{O}(2N\hat{d}), \quad (38)$$

with  $N$  as the total number of WFS, and  $\hat{d}$  and  $J$ , respectively, as the total number of B-coefficients and triangles in the global SABRE model.

For Stage-1 of the D-SABRE, we find for the complexity of the matrix–vector operation in Eq. (21):

$$U_{\text{stage-1}} = \mathcal{O}(2N_i J_i \hat{d}), \quad (39)$$

with  $N_i$  as the total number of WFS in partition  $i$  and  $J_i = J_{\Omega_i} + J_{\Xi_i}$  as the total number of triangles (including the overlap triangles) in partition  $i$ .

Determining the complexity of Stage-2 is somewhat more involved because it is an iterative algorithm consisting of a number of operations. In the following, we shall assume that

**Table 1. Comparison of Theoretical Computational Complexity of Global SABRE and D-SABRE<sup>a,b,c</sup>**

	Global SABRE	D-SABRE (per core)	Speedup (per core)
WFR	$\mathcal{O}(J^2 \hat{d})$	$\mathcal{O}(\rho^2 J_{\Omega_i}^2 \hat{d})$	$\mathcal{O}(G^2 / \rho^2)$
DPME	0	$\mathcal{O}(R J_{\Omega_i} \hat{d})$	$\mathcal{O}(1 / (R J_{\Omega_i} \hat{d}))$
DDA	0	$\mathcal{O}(L(1 + \rho)(J_{\Omega_i} \hat{d})^2)$	$\mathcal{O}(1 / (L(1 + \rho)(J_{\Omega_i} \hat{d})^2))$
Total	$\mathcal{O}(J^2 \hat{d})$	$\mathcal{O}(J_{\Omega_i} \hat{d}(\rho^2 J_{\Omega_i} + R + L \hat{d}(1 + \rho) J_{\Omega_i}))$	$\mathcal{O}(\frac{G^2}{\rho^2 + R/J_{\Omega_i} + L \hat{d}(1 + \rho)})$

<sup>a</sup>Simplex Type-I WFS geometry (i.e.,  $N = J/2$ ) is assumed. All partitions run on a separate (perfect) parallel processor.

<sup>b</sup> $J$ , number of triangles in global triangulation;  $J_i$ , number of triangles per partition.

<sup>c</sup> $G$ , total number of partitions;  $\hat{d}$ , B-coefficients per triangle ( $\hat{d} = 3$  for linear D-SABRE);  $R$ , DPME iterations ( $R \ll \hat{d} J_i$ );  $L$ , DDA iterations ( $\rho = J_i / J_{\Omega_i}$ ).

all necessary simplifications in terms of the static matrix pre-calculation have been made. We then find for the complexity of the matrix–vector and vector–vector operations associated with DPME and DDA in Eq. (33):

$$U_{\text{stage-2}} = \mathcal{O}(R J_{\Omega_i} \hat{d} + L(J_{\Omega_i} \hat{d}(1 + r_H) + c_H(1 + r_H))), \quad (40)$$

with  $R$  as the total number of DPME iterations,  $J_{\Omega_i}$  as the total number of triangles in the core triangulation of partition  $i$  [see Eq. (14)],  $r_H = \text{row}(\mathbf{H}_{i,M})$ , and  $c_H = \text{col}(\mathbf{H}_{i,M})$ .

In general (for  $\text{OL} > 0$ ), we have  $r_H \gg 1$ ,  $c_H \gg 1$ ,  $r_H < J_{\Omega_i} \hat{d} < c_H < J_i \hat{d} = \rho J_{\Omega_i} \hat{d}$ , with  $\rho$  from Eq. (15). Additionally, we can assume that  $2N_i = J_i = \rho J_{\Omega_i}$  for a simplex Type-I WFS geometry [11]. With these (conservative) assumptions and under the introduction of  $\rho$ , we can simplify and combine Eqs. (39) and (40) as

$$\begin{aligned} U_{\text{DSABRE}} &= \mathcal{O}(\rho^2 J_{\Omega_i}^2 \hat{d} + R J_{\Omega_i} \hat{d} + L((J_{\Omega_i} \hat{d})^2 + \rho(J_{\Omega_i} \hat{d})^2)) \\ &= \mathcal{O}(\rho^2 J_{\Omega_i}^2 \hat{d} + R J_{\Omega_i} \hat{d} + L(1 + \rho)(J_{\Omega_i} \hat{d})^2) \\ &= \mathcal{O}(J_{\Omega_i} \hat{d}(\rho^2 J_{\Omega_i} + R + L \hat{d}(1 + \rho) J_{\Omega_i})), \end{aligned} \quad (41)$$

from which it can be concluded that the contribution of DPME to the total computational complexity can be neglected for as long as  $R$  is small compared to the other terms in brackets, which holds for most realistic partitionings. If all parallel processors operate synchronously and no transport latency is present, the speedup factor over the global SABRE is obtained by combining Eq. (38) with Eq. (41) as

$$S_{\text{DSABRE}} = \mathcal{O}\left(\frac{2N \cdot J \hat{d}}{J_{\Omega_i} \hat{d}(\rho^2 J_{\Omega_i} + R + L \hat{d}(1 + \rho) J_{\Omega_i})}\right). \quad (42)$$

If all partitions have the same number of triangles in the core of the partition (which is desirable), we have  $G = J/J_{\Omega_i}$  with  $G$  as the total number of partitions.

In addition, if we assume that use is made of a simplex Type-I geometry (in which case  $N = J/2$ ), then using the result  $J = G J_{\Omega_i}$  reduces Eq. (42) to

$$\begin{aligned} \tilde{S}_{\text{DSABRE}} &= \mathcal{O}\left(\frac{2J/2GJ_{\Omega_i} \hat{d}}{J_{\Omega_i} \hat{d}(\rho^2 J_{\Omega_i} + R + L \hat{d}(1 + \rho) J_{\Omega_i})}\right) \\ &= \mathcal{O}\left(\frac{JG}{\rho^2 J_{\Omega_i} + R + L \hat{d}(1 + \rho) J_{\Omega_i}}\right) \\ &= \mathcal{O}\left(\frac{G^2}{\rho^2 + R/J_{\Omega_i} + L \hat{d}(1 + \rho)}\right). \end{aligned} \quad (43)$$

From Eq. (43), it can be concluded that (1) the influence of the DPME step on the computational complexity is negligible as long as  $R/J_i \ll \rho^2$ ; (2) the overlap overhead reduces the speedup by a factor  $\rho^2$ ; and (3) the DDA step is the dominant term in the computational efficiency if  $L \hat{d} > \rho$ .

As a rule of thumb, a partitioning can (and should) be designed such that the overlap overhead  $1 < \rho < 2$ . For that overlap level,  $L = 10$  produces adequate results in most cases, showing that the DDA step in the form of the variable  $L$  indeed has a dominant influence on the speedup factor. In fact, when the DDA step is not used, and if we (reasonably) assume  $R/J_i \ll \rho^2$ , we obtain a speedup factor of  $\frac{1}{\rho^2} G^2$  for the linear

D-SABRE. In Table 1, the results of the complexity analysis are summarized.

Finally, it must be noted here that the efficiency of the DDA step can be improved significantly when only interpartition smoothness is enforced in Eq. (33) instead of full-partition smoothness. In that case,  $r_H$  and  $c_H$  will be significantly smaller to the point that the overlap overhead will have a dominant contribution to the complexity.

As an example of Table 1, consider first a  $64 \times 64$  WFS array such that  $N = 4096$ , and assume that a linear spline model (i.e.,  $\hat{d} = 3$ ) is used on a Type-I sensor geometry. In this case, the global SABRE would require  $\mathcal{O}(J^2 \hat{d}) = \mathcal{O}(2 \cdot 10^8)$  flops for a single reconstruction. For the D-SABRE with 64 partitions,  $OL = 2$ ,  $R = 4$ , and no DDA ( $L = 0$ ), we have  $J_{\Omega_i} = 128$  and subsequently  $1.53 \leq \rho \leq 2.19$  with Eq. (16). In this case, the contribution of DPME can be neglected ( $R/J_{\Omega_i} \ll 1$ ), and we have as a worst case  $\mathcal{O}(\rho^2 J_{\Omega_i}^2 \hat{d}) = \mathcal{O}(2 \cdot 10^5) = \mathcal{O}(57N)$  flops per D-SABRE core per reconstruction. For a  $200 \times 200$  array and a  $20 \times 20$  partitioning ( $N = 40000$ ),  $OL = 2$ ,  $R = 10$ , and no DDA ( $L = 0$ ), we have  $J_{\Omega_i} = 200$  and subsequently  $1.42 \leq \rho \leq 1.92$ . Again, the contribution of DPME can be neglected, and we find for the worst case  $\mathcal{O}(4 \cdot 10^5) = \mathcal{O}(11N)$  flops per D-SABRE core per reconstruction.

Even though hardware-dependent issues are not in the scope of this paper, we want to mention that communication latency between the partitions for the iterative, distributed Stage-2 of the D-SABRE method is not considered to be an issue in a planned parallel GPU implementation. For the reference case of a  $200 \times 200$  SH array and a  $20 \times 20$  partitioning and a standard of the shelf GPU, all necessary data, e.g., reconstruction matrices, of the D-SABRE method can be stored directly on the GPU memory and no communication over the slow GPU to CPU connection will be necessary.

#### 4. SIMULATIONS WITH THE D-SABRE

In this section, the results from numerical experiments with the D-SABRE are presented. These experiments are aimed at validating the D-SABRE method by a simulation study.

First, the reconstruction accuracy and noise resilience of the D-SABRE are compared to that of the centralized SABRE method in Sections 4.A and 4.B. As in [11], a von Karman turbulence model was used to simulate phase screens which are then measured by a Fourier-optics-based Shack–Hartmann (SH) WFS simulator which generates wavefront phase slopes for a square grid of varying numbers of subapertures. Slope measurements of different signal-to-noise ratio (SNR) levels are created by adding Gaussian-distributed white noise.

The second part of this section contains a comparison of D-SABRE to the distributed cumulative reconstructor (CuRe-D) method which also provides parallelizable wavefront reconstruction of linear complexity and was published in [10]. Both methods were implemented for YAO [20], a Monte Carlo simulation tool for astronomical AO systems, to test their reconstruction performance on a reference simulator. The main goal of this study was to obtain a comparison of the two methods under equal conditions to understand if the D-SABRE in its most basic form reaches the same level of performance as the CuRe-D method for different decomposition levels. This

baseline version of the D-SABRE applies a spline model of polynomial degree  $d = 1$  and continuity order  $r = 0$ ; it further performs only the DPME step in Stage-2. After an open-loop analysis of reconstruction accuracy and noise resilience under the influence of additive white noise, we tested both methods in a closed-loop environment for a SCAO setting considering different levels of photon shot noise. Therefore, an additional projection step of the reconstructed phase onto the deformable mirror was implemented to allow the correction in combination with a simple integrator control provided by YAO.

#### A. Validation of D-SABRE Accuracy

The first experiment is aimed at comparing the D-SABRE with the global SABRE method from [11]. For this, a Fourier-optics-based SH lenslet array is used to obtain wavefront slopes from a set of 100 simulated wavefronts observing at a wavelength of 699 nm. For the von Karman turbulence model, a Fried coherence length of 0.2 m was used with a turbulence outer scale of 50 m and the telescope diameter is assumed to be 2 m. The SH lenslet array consists of 2500 lenslets, laid out in a  $50 \times 50$  grid with full illumination of all subapertures. Noise affecting the SH wavefront sensor measurements is simulated through white noise added to the slope measurements, with the SNR provided by the ratio of the slope and noise variance in the logarithmic decibel scale, where a SNR of 0 dB implies that the magnitude of the noise is equal to that of the signal.

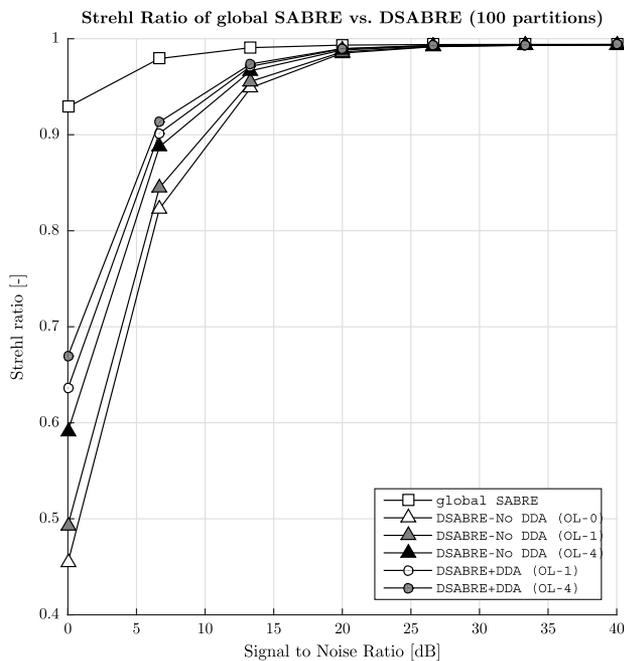
In Fig. 6, the average Strehl ratio is plotted as a function of SNR for different variants of the D-SABRE and compared to that of the global SABRE method. For the D-SABRE, the global WFS domain is decomposed into 25 partitions. For each SNR, setting a total of 100 reconstructions is performed. The results show that the D-SABRE with OL-4 and DDA approximates the global SABRE within 1% in terms of the Strehl ratio for  $SNR \geq 20$  dB, within 5% for  $SNR \geq 10$  dB, and within 10% for  $SNR \geq 5$  dB. In addition, Fig. 6 shows that including the DDA step significantly improves D-SABRE accuracy for higher noise cases.

#### B. Investigation of D-SABRE PME Error Propagation

An important issue with current state-of-the-art distributed WFR methods is the increase of the reconstruction error as the total size of the WFS array increases [10]. In contrast to these methods, D-SABRE is not subject to noise propagation within partitions because an optimal least-squares estimator is used. However, D-SABRE is subject to inaccuracies in the least-squares estimates of the piston mode offsets that are propagated between partitions. In principle, these inaccuracies will only occur if the model residue is nonwhite or if the sample of the piston offset on shared vertices is too small. In fact, as the absolute size of the partitions increases (i.e., more triangles per partition), the piston mode offset estimation will become more accurate as the total number of shared vertices used to calculate the offset will increase.

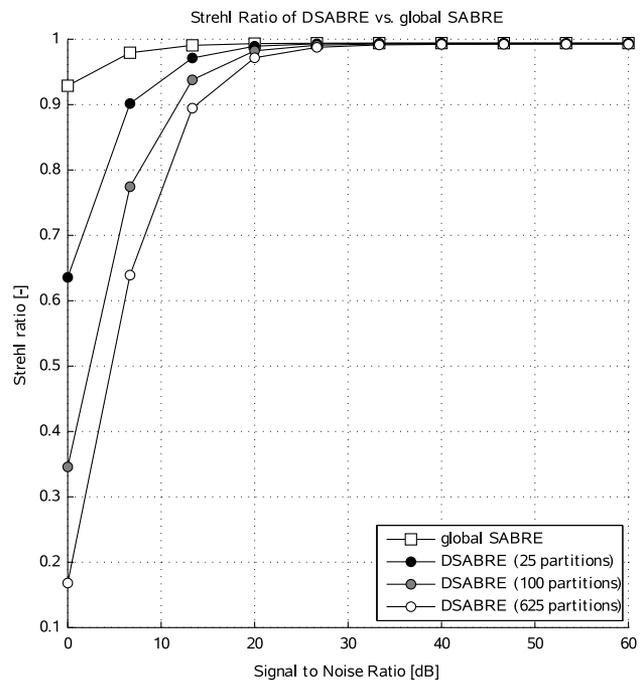
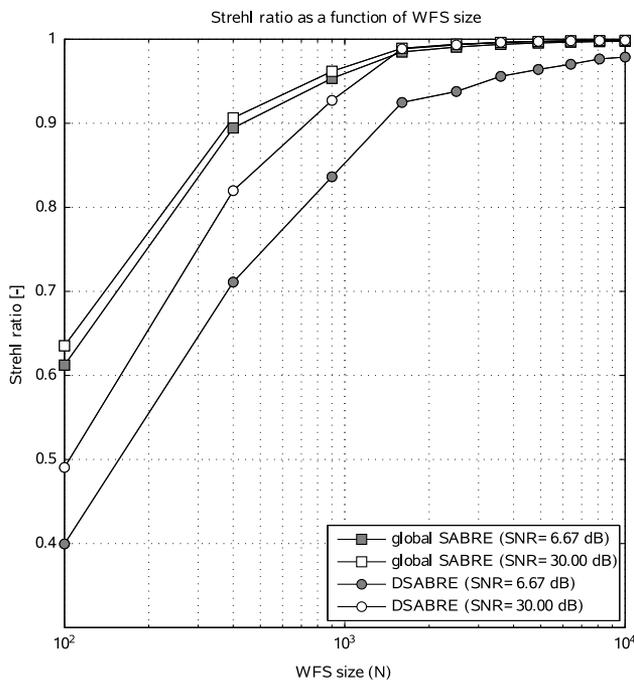
Here, we present the results of an investigation into PME error propagation issues with the D-SABRE using numerical simulations.

In the left plot of Fig. 7, the results from the simulations of WFS array sizing are shown. In this case, the WFS array size



**Fig. 6.** Comparison of the average Strehl ratio of the global SABRE method with that of various D-SABRE variants as a function of the signal-to-noise ratio, where 0 dB corresponds with equal magnitudes of noise and signal. All D-SABRE variants use a decomposition of the global  $50 \times 50$  WFS grid (2500 WFS) into 25 partitions.

is increased from  $10^2$  to  $10^4$  sensors, while the total number of partitions is fixed at 100. For each WFS array, a set of 100 reconstructions are conducted at a low SNR of 6.67 dB and a high SNR of 30 dB. From Fig. 7 it can be observed that



**Fig. 7.** Left: average Strehl ratio as a function of the total number of WFS and a fixed number of 100 partitions. Right: average Strehl ratio as a function of signal-to-noise ratio for a varying partition counts on a fixed  $50 \times 50$  WFS grid, where 0 dB corresponds with equal magnitudes of noise and signal. In both cases, an OL-1 overlap with DDA smoothing is used.

the Strehl ratio actually increases with increasing WFS array size for both the SABRE and D-SABRE method for all SNR values. Also, it demonstrates that when the absolute size of partitions increases, the propagation of PME errors *decreases*.

In the right plot of Fig. 7, the results from simulations with different partitionings of a  $50 \times 50$  grid are shown. In this case, the global WFS array is decomposed into 25, 100, and 625 local partitions. At each SNR level, 100 reconstructions are performed after which the result is averaged. This time, there is a clear positive influence on the Strehl ratio of *reducing* the total number of partitions on the fixed  $50 \times 50$  WFS array.

From Fig. 7 it can be concluded that the D-SABRE is subject to propagation of PME errors between partitions but is not subject to noise propagation. Also, the PME error actually reduces in size as the total number of shared vertices is increased.

### C. Comparison with CuRe-D in YAO Open-Loop Simulations

The linear D-SABRE method is compared with the CuRe-D in terms of reconstruction accuracy and noise propagation for an open-loop configuration.

#### 1. Open-Loop Configurations

The AO simulation tool YAO was used to create 100 wavefront realizations with a Fried parameter  $r_0 = 0.18$  cm and simulate the corresponding diffraction-based slope measurements of a Shack–Hartmann wavefront sensor with a  $64 \times 64$  subaperture array on a circular telescope pupil of 30 m diameter. In YAO, the slope measurements are computed, after full propagation and subaperture image formation, on a  $10 \times 10$  pixel array using the center of mass algorithm. Subapertures of illumination higher or equal than 50% were taken into account. In this

section, zero noise is assumed in the simulation of the slope measurements and an observation wavelength of 650 nm is considered.

Due to its hierarchical structure, the CuRe-D method is limited to partition arrays of size  $N_p \times N_p$  with  $N_p$  as powers of two, and it provides wavefront estimates at the corners of the SH subapertures. Though such partitioning is not necessary for D-SABRE, in order to compare the two methods, the D-SABRE was configured accordingly and the obtained spline estimate of the wavefront is evaluated at the subaperture corner positions located within the pupil domain. Note that the D-SABRE method computes an analytical solution of the wavefront reconstruction problem providing phase estimates at any point in the triangulation which covers the telescope pupil. In Fig. 8, we show the example of a Type-II triangulation for a  $32 \times 32$  SH subaperture array. The two graphics on the right depict a decomposition of the global partition on the left into a  $4 \times 4$  partitioning with an overlap level of  $OL = 2$ .

Analogous to Sections 4.A and 4.C, noise effects in the sensing process are simulated with additive white noise on the slope measurements provided by YAO. Low signal-to-noise ratios of  $SNR < 12$  dB are considered in this section to investigate noise resilience as well as reconstruction accuracy of the two distributed methods. To do so, the relative RMS values of the residual wavefronts are compared for varying signal-to-noise ratios. The data provided is the mean and its standard deviation computed from RMS errors of wavefront reconstructions for 100 phase realizations.

2. Open-Loop Results

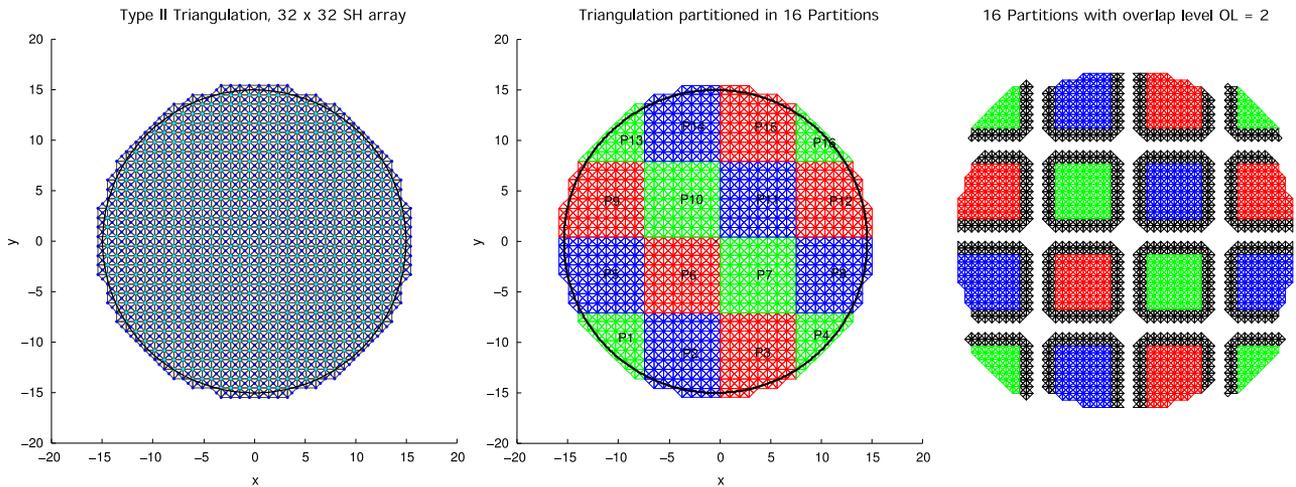
In Fig. 9, the results from simulations with different partitionings of the  $64 \times 64$  SH array are shown. For both methods, we consider  $N_p \times N_p$  partition arrays for  $N_p = 2, 4,$  and  $8$  resulting in square partitions each containing  $32 \times 32, 16 \times 16,$  and  $8 \times 8$  subapertures, respectively. In the case of D-SABRE, a

Type-II triangulation was used and overlap levels of 1, 2,  $64/2N_p,$  and  $64/N_p$  were applied.

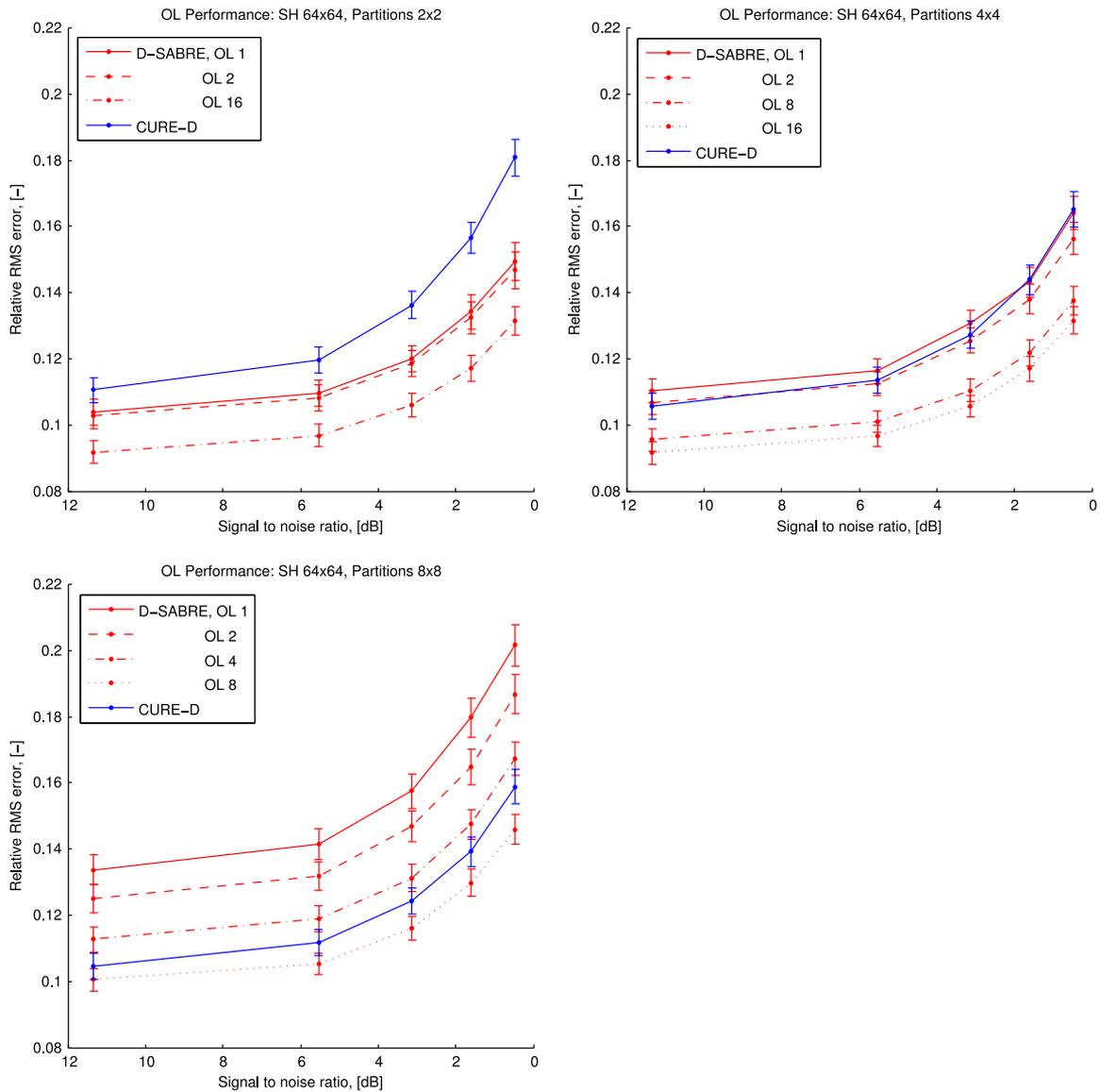
In comparison with CuRe-D, the D-SABRE shows an advantage in reconstruction accuracy for lower levels of decomposition. For the example of a  $2 \times 2$  partitioning of the domain, D-SABRE clearly outperforms CuRe-D even for a minimal amount of partition overlap. For a higher level of decomposition, equal or better accuracy can only be achieved by increasing the overlap level. For the  $4 \times 4$  partitioning,  $OL = 2$  leads to performance comparable to that of CuRe-D, whereas for  $8 \times 8$  partitioning an overlap of  $OL = 64/N_p$  is necessary to achieve better results than CuRe-D.

In terms of noise resilience, a similar trend can be observed. For lower levels of decomposition, D-SABRE proves more resilient to an increase in the amount of additive white noise than CuRe-D. We considered the intervals between SNR values at which relative RMS errors are provided in Fig. 9 and computed the average rate of change of the relative RMS error over these SNR intervals for both methods through finite differences. Averaged over the different SNR intervals, the rate with which the accuracy of D-SABRE for the  $2 \times 2$  partitioning decreases for increasing noise is only 56% of the rate observed for CuRe-D. For the  $4 \times 4$  partitioning, the rate of accuracy decay for D-SABRE increases to 70%, for the  $8 \times 8$  partitioning to 98% of the rate observed for CuRe-D. These values were obtained for an overlap level of  $OL = 64/2N_p$  in the case of the D-SABRE method.

The results presented in this section can be explained with the very different noise propagation behaviors of the two methods. It has been shown in [5] and [10] that the cumulative approach of the CuRe-D algorithm leads to an accumulation of noise for larger numbers of subapertures per partition. Hence, increasing the level of decomposition for the distributed CuRe-D improves the noise propagation properties of the method. It can be seen in Fig. 9 that this reduction of noise propagation within partitions outweighs the negative effect



**Fig. 8.** Triangulation and partitioning example for D-SABRE. Left: global Type-II triangulation for a  $32 \times 32$  SH array constructed with vertices located on SH center locations (cyan) and added vertices (blue) to form the regular Type-II triangulation and to cover the entire pupil domain. Center: partitioning of the global triangulation in 16 partitions on a  $4 \times 4$  partition array. Right: partitioning in 16 partitions with applied overlap level of  $OL = 2$ .



**Fig. 9.** Reconstruction accuracy and noise resilience for different levels of decomposition on a  $64 \times 64$  SH array of D-SABRE and CuRe-D. In the D-SABRE case, different levels of partition overlap are considered. The average relative residual RMS is plotted as a function of the signal-to-noise ratio.

of piston mode estimation errors introduced through the decomposition. In contrast, as discussed in Section 4.C, the D-SABRE does not suffer from noise propagation within partitions which contain a large number of subapertures. However, D-SABRE is subject to PME error propagation between partitions, which results in loss of performance for an increased level of decomposition given a fixed WFS array size. This effect can be counteracted by increasing the overlap level OL, or by increasing the size of the WFS array. For strong decomposition, the D-SABRE in the tested baseline version meets the performance of CuRe-D only for a sufficient amount of overlap OL, which decreases the speed up, as shown in Eq. (43).

As discussed in Section 2.B, more advanced sensor models can further improve the quality of the local D-SABREs and reduce the inaccuracies in the piston mode estimate which are currently propagated between partitions. This would allow

higher levels of decomposition and less partition overlap. Current work on a C-implementation of the D-SABRE method will make it feasible to test the method for larger WFS arrays of  $\approx 10^4 - 10^5$  subapertures as will be found in XAO systems.

#### D. Comparison with CuRe-D in YAO Closed-Loop Simulations

To conclude, D-SABRE and CuRe-D are compared in a closed-loop environment regarding their response to photon shot noise. Since D-SABRE and CuRe-D provide estimates of the residual phase and not directly actuator commands for its correction, an additional step mapping the phase onto the deformable mirror had to be implemented which was then combined with the simple integrator control law provided by YAO.

### 1. Closed-Loop Configurations

As in Section 4.C, we evaluate the D-SABRE phase estimates at the subaperture corners, i.e., the locations where phase values are provided by CuRe-D. We want to stress at this point that D-SABRE provides an analytical solution to the wavefront reconstruction problem and therefore phase estimates over the whole pupil plane. Misalignments between the subaperture and the actuator grid can be taken into account without additional approximation error, e.g., introduced through further interpolation. Further, the B-spline basis function matrix introduced in Eq. (4), which evaluates the D-SABRE phase estimates, can be precomputed. This allows us to apply the deformable mirror fit as a direct mapping between the B-coefficients and the actuator commands.

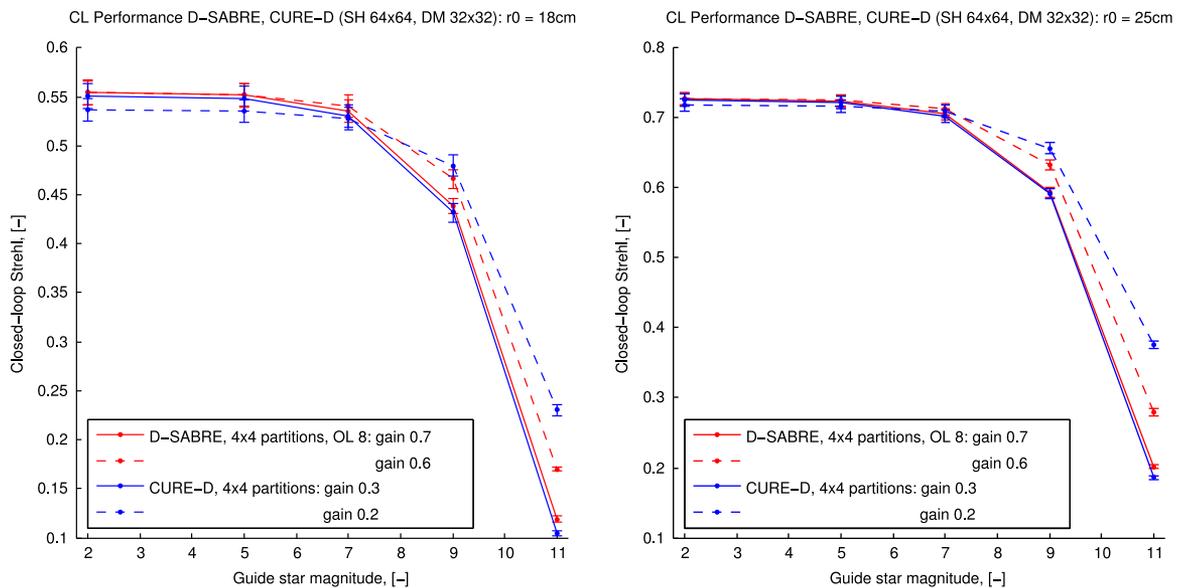
We consider a SCAO system with a stacked array deformable mirror defined through the shape of the actuator influence functions which are provided by YAO. Based on the influence function values at the subaperture corners, the actuator commands were computed in order to fit the deformable mirror optimally in a least-squares sense to the phase values. This fitting step is performed in a centralized manner and was not optimized for speed. To allow the computation of meaningful results which require a certain number of closed-loop iterations for several turbulence realizations, the density of the actuators is set lower than the density of the  $64 \times 64$  subaperture array. We opted for a  $32 \times 32$  actuator grid where the actuator positions are conjugated to every second subaperture corner in the SH array. Again, the focus was to compare the behavior of the methods for changes in parameters, like photon shot noise and loop gain, under identical circumstances, rather than aiming for a simulation of a XAO-dimensional system. The coupling between the actuators is set to 20% and the normalized

response threshold (in WFS signal) below which an actuator will not be kept as valid to 30%.

To avoid an initial transient, the long-exposure Strehl ratio is accumulated starting with the 50th iteration and the loop is evaluated for 1 s with a sampling time of 2 ms. The results are given for a wavelength of 650 nm and 10 turbulence realizations. The loop gain of the integrator control was tuned for both methods to perform optimally averaged for the same set of turbulence realizations used in the following experiment. Pairs of optimal loop gains for D-SABRE and CuRe-D were computed for two different noise levels: for a low-noise scenario simulated with a high flux guide star of magnitude 5 and for a high-noise scenario simulated with a low flux guide star of magnitude 11. The D-SABRE and CuRe-D were applied for a  $4 \times 4$  partitioning of the  $64 \times 64$  SH array; in the case of D-SABRE, an overlap level  $OL = 8$  was chosen.

### 2. Closed-Loop Results

In Fig. 10, the mean of the closed-loop Strehl ratios is provided with the standard deviation of the mean for different levels of photon shot noise, indicated by the magnitude of the guide star. The solid lines show the Strehl ratios obtained by the respective method if applying loop gains optimized for guide star magnitude 5 (i.e., high flux), and the dashed lines show the results obtained for loop gains optimized for guide star magnitude 11 (i.e., low flux). The closed-loop experiment was performed for the set of turbulence realizations with Fried parameter  $r_0 = 18$  cm used for the open-loop tests in Section 4.C. For the loop gains 0.7 and 0.3 obtained by tuning D-SABRE and CuRe-D, respectively, for high flux, the D-SABRE outperforms CuRe-D for all considered guide star magnitudes. The advantage which the applied configuration of D-SABRE has



**Fig. 10.** Long-exposure Strehl ratio comparing closed-loop performance and noise resilience of D-SABRE and CuRe-D for different levels of photon shot noise in a system with a  $64 \times 64$  SH array and a  $32 \times 32$  actuator array. Left: turbulence with Fried parameter  $r_0 = 18$  cm; right:  $r_0 = 25$  cm. A  $4 \times 4$  partitioning is constructed for both distributed methods. Two loop gains are considered for each method respectively optimized for low (solid) and high (dashed) noise levels. In the D-SABRE case,  $OL = 64 / (2N_p)$  is used. The Strehl ratio is averaged for 10 turbulence realizations and plotted as a function of the guide star magnitude.

shown for the open-loop environment with additive white noise (see Fig. 9) is preserved in the closed-loop experiment under exposure of photon shot noise, albeit not with the same margin. Decreasing the loop gain for CuRe-D to 0.2, the optimal loop gain for low flux, boosts its performance for guide star magnitudes  $>9$ . The analog adjustment of the D-SABRE loop gain to 0.6 results in a less significant improvement for the low flux range and D-SABRE does not reach the performance of CuRe-D for the resulting levels of photon shot noise. However, in the high flux range, D-SABRE proves in this experiment greater robustness to the variation of the loop gain. D-SABRE preserves its performance for guide star magnitudes  $<7$ , whereas CuRe-D is more sensitive to changes of the loop gain and suffers a significant drop in Strehl ratio. The experiment was repeated for a set of 10 turbulence realizations with Fried parameter  $r_0 = 25$  cm. The results are depicted in the right plot of Fig. 10 and confirm the findings made in this section. For a sufficient amount of overlap and a moderate level of decomposition, the baseline D-SABRE meets the performance of the CuRe-D method also in a closed-loop setting and under the impact of photon shot noise.

## 5. CONCLUSIONS

In this paper, a new distributed-multivariate-spline-based wavefront reconstruction method is introduced. This new method, the D-SABRE, is an extension of the SABRE method for wavefront reconstruction. The D-SABRE is aimed at large-scale wavefront reconstruction problems (e.g.,  $200 \times 200$  WFS grids) where global reconstructions are not realistic.

The D-SABRE method partitions the global WFS domain into any number of partitions, and then solves the local reconstruction problem in two distributed stages. In the first stage, the local wavefront reconstruction problem is solved for every partition; this stage is completely distributed in the sense that no communication between partitions is necessary. In the second stage, the local partitions are smoothed into a single continuous global wavefront. This stage requires only communication between direct partition neighbors. The blending consists of two operations: the DPME and the DDA operation. DPME is aimed at equalizing the unknown piston modes between partitions, while DDA is aimed at post-smoothing the resulting partitions. Both operations are iterative, and convergence of the operations is proved. Without DPME, no accurate global wavefront can be obtained. The DDA operation, on the other hand, can be considered optional and should only be applied when a continuous global wavefront reconstruction is desired because of its significant negative influence on computational complexity.

An analysis of computational complexity showed that a speedup factor over the SABRE can be obtained that is quadratically proportional to the total number of partitions. The most dominant factor for the complexity is the local reconstruction stage (Stage-1) and the optional distributed post-smoothing (DDA) step in Stage-2. The DPME step in Stage-2 is cheap in terms of computational complexity.

Numerical experiments show that the D-SABRE approximates the global wavefront reconstruction within 1% in terms of the Strehl ratio for  $\text{SNR} \geq 20$  dB, within 5% for

$\text{SNR} \geq 10$  dB, and within 10% for very low  $\text{SNR} \geq 5$  dB. The most important factors determining reconstruction accuracy are the overlap level between partitions and the absolute partition size, where a larger partition leads to more accurate results.

The D-SABRE is not subject to noise propagation in the same sense as the CuRe and CuRe-D methods. Instead, it is subject to propagation of piston mode offset estimations between partitions. The magnitude of this DPME error propagation decreases as the absolute size of the partitions increases. In that sense, the D-SABRE favors very large WFS arrays, that are partitioned into partitions that are as large as possible given a particular hardware setup.

Finally, a comparison of the D-SABRE with the CuRe-D method is given through open- and closed-loop simulations which are performed with the AO simulation tool YAO. The open-loop analysis of the performance under the influence of additive white noise shows a clear advantage of D-SABRE for low levels of domain partitioning. If the number of partitions is increased and the size of the partitions decreases, overlap between the partitions has to be applied in order to outperform CuRe-D. In the tested closed-loop SCAO configurations, D-SABRE and CuRe-D showed comparable behavior for increasing levels of photon shot noise. D-SABRE has proven to be more robust to variations in the gain of the control loop.

We conclude that the D-SABRE will be most useful on very large-scale ( $>200 \times 200$ ) WFS arrays, where the computational hardware favors low levels of domain partitioning resulting in partitions that are as large as possible. This is in contrast with CuRe-D, which is useful on large-scale WFS arrays but which requires an as high as possible domain partitioning to prevent noise propagation.

Future work on the D-SABRE method will be focused on extending the approach (1) to higher degree polynomials (i.e.,  $d \geq 3$ ), which requires imposing explicit boundary conditions on the local models; (2) by implementing more advanced sensor models such as that introduced in [13] that exploit the integrative nature of the SH sensors; (3) by implementing more advanced estimation schemes that exploit knowledge of turbulence statistics; and finally (4) by implementing D-SABRE on a real-world parallel hardware (GPU) setup.

## APPENDIX A

By exploiting the sparseness of  $\mathbf{A}$  in Eq. (30), we can derive two new submatrices  $\mathbf{H}_i$  and  $\mathbf{H}_{i,m}$  for each partition  $i$ . For this, we first introduce  $\mathcal{E} = 1, 2, \dots, E$  as the global index set of *all* triangle edges in the global triangulation  $\mathcal{T}$  and  $\mathcal{C} = 1, 2, \dots, J \cdot \hat{d}$  as the global index set of *all* B-coefficients in the global spline model  $s_r^d$ .

The first submatrix of  $\mathbf{A}$ , indicated as the *inner constraint matrix*  $\mathbf{H}_i$ , contains blocks that only influence B-coefficients inside the partition  $i$ :

$$\mathbf{H}_i := \mathbf{A}(\mathcal{E}_{\Omega_i}, \mathcal{C}_{\Omega_i}), \quad (\text{A1})$$

with  $\mathcal{E}_{\Omega_i} \subset \mathcal{E}$  as the set of indices of all triangle edges inside the core part  $\Omega_i$  of the subtriangulation  $\mathcal{T}_i$  from Eq. (13) and with  $\mathcal{C}_{\Omega_i} \subset \mathcal{C}$  as the set of all B-coefficient indices in  $\Omega_i$ .

The second submatrix  $\mathbf{H}_{i,m}$  contains all blocks in  $\mathbf{A}$  that influence B-coefficients inside the partition  $i$  as well as B-coefficients in neighboring partitions  $m$  through the action of the continuity conditions:

$$\mathbf{H}_{i,m} := \mathbf{A}(\mathcal{E}_{i,m}, \mathcal{C}_{i,m}), \quad (\text{A2})$$

with  $\mathcal{E}_{i,m} = \mathcal{E}_{\Omega_i} \cup \mathcal{E}_{i \rightarrow m}$  as the indices of all edges in  $\mathcal{T}_i$  that influence  $s_{r_i}^d$ . Finally, we define  $\mathcal{C}_{i,m} \subset \mathcal{C}$  as the indices of all B-coefficients in  $s_{r_i}^d$  as well as all B-coefficients in  $s_{r_m}^d$  that are subject to continuity conditions on  $\mathcal{T}_i \cap \mathcal{T}_m$ .

We define  $\mathbf{y}_i = \mathbf{y}(\mathcal{E}_{\Omega_i})$  as the local dual vector,  $\mathbf{c}_i = \mathbf{c}(\mathcal{C}_{\Omega_i})$  as the local B-coefficient vector, and  $\mathbf{c}_{i,m} = \mathbf{c}(\mathcal{C}_{i,m})$  all B-coefficients in  $s_{r_i}^d$  combined with all B-coefficients in  $s_{r_m}^d$  that are subject to continuity conditions on  $\Omega_i \cap \Omega_m$ .

## REFERENCES

1. F. Roddier, "Curvature sensing and compensation: a new concept in adaptive optics," *Appl. Opt.* **27**, 1223–1225 (1988).
2. D. L. Fried, "Least-square fitting a wave-front distortion estimate to an array of phase-difference measurements," *J. Opt. Soc. Am. A* **67**, 370–375 (1977).
3. J. Herrmann, "Least-squares wave front errors of minimum norm," *J. Opt. Soc. Am. A* **70**, 28–35 (1980).
4. W. H. Southwell, "Wave-front estimation from wave-front slope measurements," *J. Opt. Soc. Am. A* **70**, 998–1006 (1980).
5. M. Rosensteiner, "Cumulative reconstructor: fast wavefront reconstruction algorithm for extremely large telescopes," *J. Opt. Soc. Am. A* **28**, 2132–2138 (2011).
6. L. A. Poyneer, D. T. Gavel, and J. M. Brase, "Fast wave-front reconstruction in large adaptive optics systems with use of the Fourier transform," *J. Opt. Soc. Am. A* **19**, 2100–2111 (2002).
7. P. J. Hampton, P. Agathoklis, and C. Bradley, "A new wave-front reconstruction method for adaptive optics systems using wavelets," *IEEE J. Sel. Top. Signal Proc.* **2**, 781–792 (2008).
8. L. Gilles, C. R. Vogel, and B. L. Ellerbroek, "Multigrid preconditioned conjugate-gradient method for large-scale wave-front reconstruction," *J. Opt. Soc. Am. A* **19**, 1817–1822 (2002).
9. E. Thiébaud and M. Tallon, "Fast minimum variance wavefront reconstruction for extremely large telescopes," *J. Opt. Soc. Am. A* **27**, 1046–1059 (2010).
10. M. Rosensteiner, "Wavefront reconstruction for extremely large telescopes via cure with domain decomposition," *J. Opt. Soc. Am. A* **29**, 2328–2336 (2012).
11. C. C. de Visser and M. Verhaegen, "Wavefront reconstruction in adaptive optics systems using nonlinear multivariate splines," *J. Opt. Soc. Am. A* **30**, 82–95 (2013).
12. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.* **3**, 1–122 (2010).
13. S. Guo, R. Zhan, J. Li, J. Zou, R. Xu, and C. Liu, "Sequential deconvolution from wave-front sensing using bivariate simplex splines," *Opt. Commun.* **342**, 73–78 (2015).
14. C. C. de Visser, Q. P. Chu, and J. A. Mulder, "Differential constraints for bounded recursive identification with multivariate splines," *Automatica* **47**, 2059–2066 (2011).
15. R. Rannacher, S. Turek, and J. G. Heywood, "Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations," *Int. J. Numer. Methods Fluids* **22**, 325–352 (1996).
16. M. J. Lai and L. L. Schumaker, "A domain decomposition method for computing bivariate spline fits of scattered data," *SIAM J. Numer. Anal.* **47**, 911–928 (2009).
17. M. J. Lai and L. L. Schumaker, *Spline Functions on Triangulations* (Cambridge University, 2007).
18. C. C. de Visser, "Global nonlinear model identification with multivariate splines," Ph.D. thesis (Delft University of Technology, 2011).
19. C. C. de Visser, Q. P. Chu, and J. A. Mulder, "A new approach to linear regression with multivariate splines," *Automatica* **45**, 2903–2909 (2009).
20. F. Rigaut and M. van Dam, "Simulating astronomical adaptive optics systems using YAO," in *Third AO4ELT Conference—Adaptive Optics for Extremely Large Telescopes*, Florence, Italy, May 2013.