Fast minimum variance wavefront reconstruction for extremely large telescopes

Eric Thiébaut[1,2,3,4,*] and Michel Tallon[1,2,3,4]

[1]*Université de Lyon, F-69000 Lyon, France*
[2]*Université de Lyon 1, F-69622 Villeurbanne, France*
[3]*Centre de Recherche Astrophysique de Lyon, Observatoire de Lyon, 9 Avenue Charles André, F-69561 Saint-Genis Laval CEDEX, France*
[4]*CNRS, UMR 5574, Ecole Normale Supérieure de Lyon, F-69007 Lyon, France*
[*]*Corresponding author: thiebaut@obs.univ-lyon1.fr*

Received November 30, 2009; revised February 22, 2010; accepted February 23, 2010; posted March 1, 2010 (Doc. ID 120178); published April 13, 2010

We present what we believe to be a new algorithm, FRactal Iterative Method (FRiM), aiming at the reconstruction of the optical wavefront from measurements provided by a wavefront sensor. As our application is adaptive optics on extremely large telescopes, our algorithm was designed with speed and best quality in mind. The latter is achieved thanks to a regularization that enforces prior statistics. To solve the regularized problem, we use the conjugate gradient method, which takes advantage of the sparsity of the wavefront sensor model matrix and avoids the storage and inversion of a huge matrix. The prior covariance matrix is, however, non-sparse, and we derive a fractal approximation to the Karhunen–Loève basis thanks to which the regularization by Kolmogorov statistics can be computed in $\mathcal{O}(N)$ operations, with N being the number of phase samples to estimate. Finally, we propose an effective preconditioning that also scales as $\mathcal{O}(N)$ and yields the solution in five to ten conjugate gradient iterations for any N. The resulting algorithm is therefore $\mathcal{O}(N)$. As an example, for a 128×128 Shack–Hartmann wavefront sensor, the FRiM appears to be more than 100 times faster than the classical vector-matrix multiplication method.

OCIS codes: 010.7350, 100.3190, 110.1080.

1. INTRODUCTION

The standard and most used method for adaptive optics (AO) control is based on a vector-matrix multiplication (VMM) of the vector of wavefront sensor measurements by the so-called control matrix [1]. This operation gives an update of the commands to be sent to the deformable mirrors to adjust the correction of the corrugated incoming wavefronts. The control matrix is precomputed, generally using modal control optimization [2]. The complexity of computing the control matrix using standard methods scales as $\mathcal{O}(N^3)$, where N is the number of unknowns (phase samples or actuator commands), and applying real time VMM scales as $\mathcal{O}(N^2)$. This computational burden can be reasonably handled on current AO systems where $N\lesssim10^3$.

For future Extremely Large Telescopes (ELTs), the number of actuators being considered is in the range 10^4-10^5. This huge increase is the result of both the larger diameter of the ELTs [3] and the emergence of new architectures for the AO systems, using either a greater density of actuators (Extreme AO) or combining several deformable mirrors and wavefront sensors (multi-conjugate AO, multi-object AO) [4]. The necessary computational power for real time control on such systems is currently unattainable when using standard methods.

More efficient algorithms are thus required and have been developed in recent years. Poyneer *et al.* [5] derived an accurate Fourier transform wavefront reconstructor by solving the boundary problem in circular apertures. This reconstructor scales as $\mathcal{O}(N\log N)$ and is shown to be effective for Extreme AO [6]. MacMartin [7] studied several approximate approaches such as a multiple-layer hierarchic reconstruction, which scales as $\mathcal{O}(N)$.

Although least-squares algorithms give suitable results for single star AO systems (classical on-axis AO or Extreme AO), minimum variance reconstruction is required to minimize the effects of the missing data or unseen modes in the other AO schemes [8]. In the context of minimum variance for multi-conjugate AO, Ellerbroek [9] could apply sparse matrix techniques (Cholesky factorization) using a sparse approximation of the turbulence statistics, and introducing as low-rank adjustments the non-sparse matrix terms arising from the global tip/tilt measurement errors associated with laser guide stars. However the interactions between the layers in their tomographic modeling reduce the efficiency of the sparse direct decomposition methods [10].

Iterative methods are also extensively studied in this context. Their main asset is their ability to iteratively compute the unknowns from the measurements using direct sparse matrices, and so the storage of a precomputed inverse full matrix is not necessary. One major problem with iterative methods is the increase in the number of iterations with the number of unknowns to estimate [11–13]. As an example, Wild *et al.* [14] proposed to use the closed-loop AO system itself as an iterative processor, but the performance of the least-squares reconstruction depends on the loop frequency of the AO system, which should be higher than usual.

The most successful iterative methods in AO are now

based on preconditioned conjugate gradients (PCGs) [15], where some of the previous approximate reconstruction methods are embedded as preconditioners to ensure a small number of iterations (see Section 4). Gilles *et al.* [16] described a multigrid PCG algorithm, mainly aimed at Extreme AO and scaling as $\mathcal{O}(N \log N)$. The multigrid preconditioner is somewhat related to the multiple-layers hierarchic reconstruction [7]. This wavefront reconstruction method has been improved with a faster approximation to the turbulence statistics, scaling as $\mathcal{O}(N)$ [17]. The multigrid PCG algorithm has also been developed for multi-conjugate AO [18]. In this case, the structure of the matrix is more complex and brings some limitations. More recently, a Fourier domain preconditioner was introduced [19,20] in the context of multi-conjugate AO, with a faster reconstruction than a multigrid PCG. In this case, the preconditioner is related to the Fourier transform wavefront reconstructor [5]. Both multigrid and Fourier domain preconditioners were examined for the Thirty Meter Telescope project [21,22].

In this work, we propose novel methods to address the two critical points previously seen in iterative methods for wavefront reconstruction: estimation of the atmospheric phase covariance matrix and preconditioning. We need a sparse representation of the inverse of the atmospheric phase covariance matrix to efficiently introduce priors in the minimum variance estimator. Currently, we can choose between a good representation in the Fourier domain with $\mathcal{O}(N \log N)$ complexity [16,19] and a widely used sparse biharmonic approximation introduced by Ellerbroek [9], less accurate [19], but scaling as $\mathcal{O}(N)$. With the FRactal Iterative Method (FRiM), we introduce a so-called "fractal operator" as a multiscale algorithm with $\mathcal{O}(N)$ complexity. This operator, both accurate and very fast, was inspired by the mid-point method of Lane *et al.* [23] to generate a Kolmogorov phase screen. It can be used for any wavefront structure function. It allows us to very efficiently apply the inverse of the phase covariance matrix to any vector.

We show that this fractal operator is also very efficient when used as a preconditioner. It allows the wavefront reconstruction to be iteratively computed in a space of statistically independent modes. We additionally use a classical Jacobi preconditioner or a new "optimal diagonal preconditioner" to further improve the convergence. The number of iterations is $\lesssim 10$ for a full wavefront reconstruction whatever is the size of the system, with a number of floating point operations $\sim 34 \times N$ per iteration. The method is therefore globally $\mathcal{O}(N)$.

In the following, we first derive the analytical expression for the minimum variance restored wavefront and the equations to be solved. We then introduce the fractal operator allowing a fast computation of the regularization term in an iterative method such as conjugate gradients (CGs). We then propose two fast preconditioners to further speed up the iterative algorithm. We finally use numerical simulations to test the performances of the FRiM.

2. MINIMUM VARIANCE SOLUTION

A. Model of Data

We assume that the wavefront sensor provides measurements of spatial derivatives (slopes or curvatures) of the phase, which are linearly related to the wavefront seen by the sensor,

$$\boldsymbol{d} = \mathbf{S} \cdot \boldsymbol{w} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{d} \in \mathbb{R}^M$ is the *data* vector provided by the sensor, $\boldsymbol{w} \in \mathbb{R}^N$ is the vector of sampled wavefront values, $\mathbf{S} \in \mathbb{R}^{M \times N}$ is the sensor response matrix, and $\boldsymbol{n} \in \mathbb{R}^M$ accounts for the noise and model errors. This equation is general as long as the wavefront sensor is linear. As a typical case, we will however consider a Shack–Hartmann wavefront sensor with Fried geometry [24] in our simulations and for the evaluation of the efficiency of the algorithms.

B. Optimal Wavefront Reconstructor

The estimation of the wavefront $\boldsymbol{w}$ given the data $\boldsymbol{d}$ is an inverse problem which must be solved using proper regularization in order to improve the quality of the solution while avoiding noise amplification or ambiguities due to missing data [25]. In order to keep the problem as simple as possible, we first introduce the requirement that the solution be a linear function of the data, i.e., the restored wavefront satisfies

$$\tilde{\boldsymbol{w}} \overset{\text{def}}{=} \mathbf{R} \cdot \boldsymbol{d}, \tag{2}$$

where $\mathbf{R}$ is the restoration matrix and $\boldsymbol{d}$ is the wavefront sensor measurements. Some quality criterion is needed to derive the expression for the restoration matrix $\mathbf{R}$. For instance, we can require that, on average, the difference between the restored wavefront $\tilde{\boldsymbol{w}}$ and the true wavefront $\boldsymbol{w}$ be as small as possible by minimizing $\langle \|\tilde{\boldsymbol{w}} - \boldsymbol{w}\|^2 \rangle$, where $\langle\ \rangle$ denotes the expected value of its argument. It is interesting to note that minimizing (on average) the variance of the residual wavefront yields the optimal Strehl ratio [26] since

$$\mathrm{SR} \simeq \exp\left(-\frac{1}{\mathcal{A}} \int_{\text{pupil}} [\tilde{w}(\boldsymbol{r}) - w(\boldsymbol{r})]^2 \mathrm{d}\boldsymbol{r} \right), \tag{3}$$

where $\boldsymbol{r}$ is the position in the pupil, $\mathcal{A}$ is the area of the pupil, and $w(\boldsymbol{r})$ is the wavefront phase in radian units. The *best* reconstruction matrix according to our criterion then satisfies

$$\mathbf{R}^\dagger = \arg\min_{\mathbf{R}} \langle \|\mathbf{R} \cdot \boldsymbol{d} - \boldsymbol{w}\|^2 \rangle. \tag{4}$$

Accounting for the facts that the wavefront $\boldsymbol{w}$ and the errors $\boldsymbol{n}$ are uncorrelated and have zero means, i.e., $\langle \boldsymbol{n} \rangle = 0$ and $\langle \boldsymbol{w} \rangle = 0$, the minimum variance reconstructor expands as [27]

$$\mathbf{R}^\dagger = \mathbf{C}_{\boldsymbol{w}} \cdot \mathbf{S}^{\mathrm{T}} \cdot (\mathbf{S} \cdot \mathbf{C}_{\boldsymbol{w}} \cdot \mathbf{S}^{\mathrm{T}} + \mathbf{C}_{\boldsymbol{n}})^{-1}, \tag{5}$$

where $\mathbf{C}_{\boldsymbol{n}} \overset{\text{def}}{=} \langle \boldsymbol{n} \cdot \boldsymbol{n}^{\mathrm{T}} \rangle$ is the covariance matrix of the errors and $\mathbf{C}_{\boldsymbol{w}} \overset{\text{def}}{=} \langle \boldsymbol{w} \cdot \boldsymbol{w}^{\mathrm{T}} \rangle$ is the *a priori* covariance matrix of the wavefront. Applying this reconstructor to the data $\boldsymbol{d}$ requires solving a linear problem with as many equations as there are measurements. Generally, wavefront sensors provide more measurements than wavefront samples (about twice as many for a Shack–Hartmann or a curva-

ture sensor). Fortunately, from the following obvious identities [28]:

$$\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}\cdot\mathbf{C}_w\cdot\mathbf{S}^{\mathrm{T}}+\mathbf{S}^{\mathrm{T}}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot(\mathbf{S}\cdot\mathbf{C}_w\cdot\mathbf{S}^{\mathrm{T}}+\mathbf{C}_n)$$
$$=(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}+\mathbf{C}_w^{-1})\cdot\mathbf{C}_w\cdot\mathbf{S}^{\mathrm{T}},$$

we can rewrite the optimal reconstructor in Eq. (5) as

$$\mathbf{R}^{\dagger}=(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}+\mathbf{C}_w^{-1})^{-1}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}, \tag{6}$$

which involves solving just as many linear equations as there are wavefront samples. The linear reconstructor defined in Eq. (6) is the expression to be preferred in our case.

C. Links with Other Approaches

Using Eq. (6) for the reconstructor, the minimum variance restored wavefront is given by

$$\boldsymbol{w}^{\dagger}\overset{\mathrm{def}}{=}\mathbf{R}^{\dagger}\cdot\boldsymbol{d}=(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}+\mathbf{C}_w^{-1})^{-1}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\boldsymbol{d},$$

which is also the solution of the quadratic problem,

$$\boldsymbol{w}^{\dagger}=\arg\min_{\boldsymbol{w}}\{(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})+\boldsymbol{w}^{\mathrm{T}}\cdot\mathbf{C}_w^{-1}\cdot\boldsymbol{w}\},$$

where $(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})$ is the so-called χ^2 which measures the discrepancy between the data and their model and $\boldsymbol{w}^{\mathrm{T}}\cdot\mathbf{C}_w^{-1}\cdot\boldsymbol{w}$ is a Tikhonov regularization term which enforces *a priori* covariance of the unknowns. Thus Eq. (6) is also the result of the maximum *a posteriori* (MAP) problem. Here, the usual hyper-parameter is hidden in $\mathbf{C}_w$ which is proportional to $(D/r_0)^{5/3}$, where r_0 is the Fried parameter [29]. As already noted by other authors (see, e.g., Rousset [30]), the minimum variance estimator is directly related to Wiener optimal filtering.

Actual AO systems make use of some expansion of the wavefront on a basis of modes, with regularization being achieved by setting the ill-conditioned modes to zero. This technique is similar to truncated singular value decomposition [30]. Since truncation results in aliasing, we expect that the MAP solution will be a better approximation to the wavefront.

D. Iterative Method

The optimal wavefront can be computed in different ways. For instance, the matrix $\mathbf{R}$ can be computed once, using Eq. (5) or Eq. (6), and then applied to every data set $\boldsymbol{d}$. Since it requires the numerical inversion of an $N\times N$ matrix, the direct computation of $\mathbf{R}$ scales as $\mathcal{O}(N^3)$ operations [13]. The reconstructor $\mathbf{R}$ is a $N\times M$ matrix and is not sparse in practice. Hence, the storage of $\mathbf{R}$ requires $MN\approx 2N^2$ floating point numbers and computing $\mathbf{R}\cdot\boldsymbol{d}$ requires $\approx 2MN\approx 4N^2$ floating point operations. For large numbers of degrees of freedom $N\propto(D/r_0)^2$, the computer time spent by the matrix-vector multiplication can be too long for real time applications. Moreover the memory requirement (e.g., for $N\simeq 10^4$, 1.5 Gbits of memory are needed to store $\mathbf{R}$) may be such that memory page faults dominate the computation time of the matrix-vector multiplication.

In order to avoid the direct matrix inversion and the matrix-vector product required by the explicit computation of $\mathbf{R}$, we use an iterative method to solve the linear system

$$(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}+\mathbf{C}_w^{-1})\cdot\boldsymbol{w}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\boldsymbol{d}, \tag{7}$$

which leads to the optimal wavefront $\boldsymbol{w}$ for every data set $\boldsymbol{d}$. For the purpose of the discussion, Eq. (7) can be put in a more generic form,

$$\mathbf{A}\cdot\boldsymbol{x}=\boldsymbol{b}, \tag{8}$$

where, in the case of Eq. (7), $\boldsymbol{x}=\boldsymbol{w}$ and

$$\mathbf{A}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}+\mathbf{C}_w^{-1} \tag{9}$$

is the so-called left hand side matrix, whereas

$$\boldsymbol{b}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\boldsymbol{d} \tag{10}$$

is the so-called right hand side vector.

Barrett *et al.* [15] reviewed a number of iterative algorithms for solving linear systems like Eq. (8). An advantage of these methods is that they do not explicitly require the matrix $\mathbf{A}$; it is sufficient to be able to compute the product of matrix $\mathbf{A}$ (or its transpose) with any given vector. The iterative algorithm therefore fully benefits from the possibility to compute the matrix-vector products in much less than $\mathcal{O}(N^2)$ operations when $\mathbf{A}$ is sparse or has some special structure. This is particularly relevant in our case since applying $\mathbf{A}$ can be achieved by matrix-vector products very fast to compute as shown in Subsections 2.E and 2.F. The drawback of iterative methods is that the computational burden scales as the number of iterations required to approximate the solution with sufficient precision. In the worst case, the number of iterations can theoretically be as high as the number of unknowns N [12,13]. In practice and because of numerical rounding errors, ill-conditioning of the system in Eq. (7) can result in a much higher number of iterations, even on small systems. This problem can, however, be greatly reduced by means of a good preconditioner [12,15].

By construction, $\mathbf{A}$ given by Eq. (9) is a symmetric positive definite matrix and the CG [15] is the iterative method of choice to solve the system in Eq. (8). Figure 1 shows the steps of the CG algorithm to solve the system $\mathbf{A}\cdot\boldsymbol{x}=\boldsymbol{b}$. This method is known to have a super-linear rate of convergence [12] and can be accelerated by using a proper preconditioner $\mathbf{M}\approx\mathbf{A}$ for which solving $\mathbf{M}\cdot\boldsymbol{z}=\boldsymbol{r}$ for $\boldsymbol{z}$ (with $\boldsymbol{r}=\boldsymbol{b}-\mathbf{A}\cdot\boldsymbol{x}$) is much cheaper than solving Eq. (8) for $\boldsymbol{x}$. The preconditioner can also be directly specified by its inverse $\mathbf{Q}=\mathbf{M}^{-1}$ such that $\mathbf{Q}\approx\mathbf{A}^{-1}$ and then $\boldsymbol{z}=\mathbf{Q}\cdot\boldsymbol{r}$ in the CG algorithm. Without a preconditioner, taking $\mathbf{M}=\mathbf{Q}=\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, yields the unpreconditioned version of the CG algorithm. In Section 4 we investigate various means to obtain an effective preconditioner for the wavefront reconstruction problem.

In the remainder of this section, we derive means to quickly compute the dot product with the matrix $\mathbf{A}$ in Eq. (9). To that end, we consider separately the Hessian matrix $\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_n^{-1}\cdot\mathbf{S}$ of the likelihood term and that of the regularization term $\mathbf{C}_w^{-1}$.

initialization:
  compute $\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{x}_0$ for some initial guess $\boldsymbol{x}_0$
  let $k = 0$
until convergence do
  solve $\mathbf{M} \cdot \boldsymbol{z}_k = \boldsymbol{r}_k$ for $\boldsymbol{z}_k$ (apply preconditioner)
  $\rho_k = \boldsymbol{r}_k^{\mathrm{T}} \cdot \boldsymbol{z}_k$
  if $k = 0$, then
    $\boldsymbol{p}_k = \boldsymbol{z}_k$
  else
    $\boldsymbol{p}_k = \boldsymbol{z}_k + (\rho_k/\rho_{k-1})\, \boldsymbol{p}_{k-1}$
  endif
  $\boldsymbol{q}_k = \mathbf{A} \cdot \boldsymbol{p}_k$
  $\alpha_k = \rho_k/(\boldsymbol{p}_k^{\mathrm{T}} \cdot \boldsymbol{q}_k)$ (optimal step size)
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\, \boldsymbol{p}_k$
  $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k\, \boldsymbol{q}_k$
  $k \leftarrow k + 1$
done

Fig. 1. PCG algorithm for solving $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$, where $\mathbf{A}$ is a symmetric positive definite matrix and $\mathbf{M}$ is a preconditioner. The unpreconditioned version of the algorithm is simply obtained by taking $\mathbf{M} = \mathbf{I}$; hence $\boldsymbol{z}_k = \boldsymbol{r}_k$.

E. Computation of the Likelihood Term

Most AO systems use either a Shack–Hartmann sensor which provides measurements of the local gradient of the wavefront or a curvature sensor which measures the local curvature of the wavefront [1]. Since such sensors probe local spatial derivatives of the wavefront, their response can be approximated by local finite differences which yields a very sparse linear operator $\mathbf{S}$. Although some non-sparse matrix terms can appear due to tilt indetermination with laser guide stars or to take account of natural guide star tip/tilt sensors. Owing to the low rank of these modes, sparse matrix models can still be applied [9]. Thus, denoting N_{dif} the number of wavefront samples required to compute the local finite differences, only $\approx M \times N_{\mathrm{dif}}$ out of $M \times N$ coefficients of $\mathbf{S}$ are nonzero. For instance, Fig. 2 shows the Fried geometry of the Shack–Hartmann sensor model [24] which we used in our numerical simulations. The error free slopes are related to the wavefront by

$$d_x(x,y) = \tfrac{1}{2}[w(x+a,y+a) + w(x+a,y) - w(x,y+a) - w(x,y)],$$

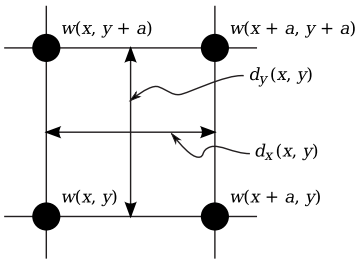

Fig. 2. Wavefront sensor with Fried geometry as used for our simulations. The black circles stand for phase samples $w(x,y)$, at the corners of the square subapertures of size a. This model is exact if we assume that the wavefront at any point in the pupil is obtained from a bilinear interpolation of phase samples at the corner of the subapertures.

$$d_y(x,y) = \tfrac{1}{2}[w(x+a,y+a) - w(x+a,y) + w(x,y+a) - w(x,y)], \tag{11}$$

where (x,y) are the pupil coordinates, d_x and d_y are the slopes along the x and y directions, and a is the sampling step. Hence $N_{\mathrm{dif}}=4$, in our case, whatever is the number of degrees of freedom. Besides, to a good approximation, wavefront sensors provide uncorrelated measurements [1]; hence the covariance matrix $\mathbf{C}_{\boldsymbol{n}}$ of the errors can be taken as a diagonal matrix,

$$\mathbf{C}_{\boldsymbol{n}} \approx \mathrm{diag}(\mathrm{Var}(\boldsymbol{n})), \tag{12}$$

where $\mathrm{Var}(\boldsymbol{n})$ is the vector of noise and error variances. Since $\mathbf{C}_{\boldsymbol{n}}$ is diagonal, its inverse $\mathbf{C}_{\boldsymbol{n}}^{-1}$ is diagonal and trivial to compute. Finally, the matrices $\mathbf{S}$ and $\mathbf{C}_{\boldsymbol{n}}^{-1}$ are sparse and the dot product by $\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \mathbf{S}$ can be therefore computed in $\mathcal{O}(N)$ operations.

F. Fast Estimation of the Regularization Term

Unlike $\mathbf{C}_{\boldsymbol{n}}$ and $\mathbf{C}_{\boldsymbol{n}}^{-1}$, neither $\mathbf{C}_{\boldsymbol{w}}$ nor $\mathbf{C}_{\boldsymbol{w}}^{-1}$ is sparse. We introduce here a way to derive an approximation for $\mathbf{C}_{\boldsymbol{w}}^{-1}$ so that it can be applied to a vector with a small number of operations.

We first consider the following decomposition of $\mathbf{C}_{\boldsymbol{w}}$:

$$\mathbf{C}_{\boldsymbol{w}} = \mathbf{K} \cdot \mathbf{K}^{\mathrm{T}}, \tag{13}$$

where $\mathbf{K}$ is a square invertible matrix. Since $\mathbf{C}_{\boldsymbol{w}}$ is positive definite, there exist a number of possibilities for such a factorization: Cholesky decomposition [9,13,17], spectral factorization, etc. We then use this decomposition to define new variables $\boldsymbol{u}$ based on the wavefront $\boldsymbol{w}$,

$$\boldsymbol{u} \overset{\mathrm{def}}{=} \mathbf{K}^{-1} \cdot \boldsymbol{w}. \tag{14}$$

The expected value of $\boldsymbol{u}$ is $\langle \boldsymbol{u} \rangle = \mathbf{K}^{-1} \cdot \langle \boldsymbol{w} \rangle = \mathbf{0}$, and its covariance matrix therefore satisfies

$$\mathbf{C}_{\boldsymbol{u}} = \langle \boldsymbol{u} \cdot \boldsymbol{u}^{\mathrm{T}} \rangle = \mathbf{K}^{-1} \cdot \langle \boldsymbol{w} \cdot \boldsymbol{w}^{\mathrm{T}} \rangle \cdot \mathbf{K}^{-\mathrm{T}} = \mathbf{K}^{-1} \cdot \mathbf{C}_{\boldsymbol{w}} \cdot \mathbf{K}^{-\mathrm{T}} = \mathbf{I},$$

which shows that the new variables are independent and identically distributed following a normal law: $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. This gives rise to a method for generating wavefronts since, from a set $\boldsymbol{u}$ of N independent random values following a normal law, taking $\boldsymbol{w} = \mathbf{K} \cdot \boldsymbol{u}$ yields a random wavefront with the expected covariance. Finally, using this re-parameterization, it is possible to rewrite the regularization term as

$$\boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{w}}^{-1} \cdot \boldsymbol{w} = \boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1} \cdot \boldsymbol{w} = \|\mathbf{K}^{-1} \cdot \boldsymbol{w}\|_2^2 = \|\boldsymbol{u}\|_2^2. \tag{15}$$

Then, depending on whether the problem is solved for the wavefront samples $\boldsymbol{w}$ or for the so-called *wavefront generators* $\boldsymbol{u}$ [cf. Eqs. (38) and (39) in Section 5], each CG iteration would be cheap to compute provided either (i) operators $\mathbf{K}^{-1}$ and $\mathbf{K}^{-\mathrm{T}}$ are fast to apply or (ii) operators $\mathbf{K}$ and $\mathbf{K}^{\mathrm{T}}$ are fast to apply.

A comparable re-parameterization has been introduced by Roddier [31] for generating turbulent wavefronts using a Zernike expansion of randomly weighted Karhunen–Loève functions. This however requires one to diagonalize a huge $N \times N$ matrix $\mathbf{C}_{\boldsymbol{w}}$, a procedure that costs at least

$\mathcal{O}(N^3)$ operations, and would give a *slow* operator $\mathbf{K}$ (or $\mathbf{K}^{-1}$) taking $\mathcal{O}(N^2)$ operations to apply.

Exploiting the fractal structure of turbulent wavefronts, Lane *et al.* [23] derived a fast method to generate wavefronts by a mid-point algorithm. In what follows, we show that their method amounts to approximating the effect of operator $\mathbf{K}$ in $\mathcal{O}(N)$ operations and we derive algorithms to apply the corresponding $\mathbf{K}^{-1}$, $\mathbf{K}^{\mathrm{T}}$, and $\mathbf{K}^{-\mathrm{T}}$ operators that also take $\mathcal{O}(N)$ operations. We propose to use these so-called *fractal operators* for a fast computation of the regularization and also as effective pre-conditioners to speed up the CG iterations.

3. FRACTAL OPERATORS

A. Principle and Structure Function

The mid-point algorithm [23] starts at the largest scales of the wavefront and step-by-step builds smaller scales by interpolating the wavefront values at the previous scale and by adding a random value with a standard deviation computed so that the new wavefront values and their neighbors have the expected structure function. Using $\mathbf{K}_j$ to denote the linear operator which generates the wavefront values at the jth scale, the linear operator $\mathbf{K}$ can be factorized as

$$\mathbf{K} = \mathbf{K}_1 \cdot \mathbf{K}_2 \cdot \cdots \cdot \mathbf{K}_p, \tag{16}$$

where p is the number of scales, $\mathbf{K}_p$ generates the four outermost wavefront values, and $\mathbf{K}_1$ generates the wavefront values at the finest scale. The original mid-point algorithm cannot be used directly for our needs because it is not invertible. In this section, we reconsider the mid-point algorithm to derive new expressions for the $\mathbf{K}_j$'s such that they are sparse, invertible, and such that their inverses are also sparse.

The structure function of the wavefront is the expected value of the quadratic difference between two phases of a turbulent wavefront,

$$\langle [w(\boldsymbol{r}_i) - w(\boldsymbol{r}_j)]^2 \rangle = f(|\boldsymbol{r}_i - \boldsymbol{r}_j|), \tag{17}$$

where, e.g.,

$$f(r) = 6.88 \times (r/r_0)^{5/3}, \tag{18}$$

for a turbulent wavefront obeying Kolmogorov's law. The structure function is stationary (shift-invariant) and isotropic since it only depends on the distance $|\boldsymbol{r}_i - \boldsymbol{r}_j|$ between the considered positions $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ in the wavefront. From the structure function, we can deduce the covariance of the wavefront between two positions in the pupil,

$$C_{i,j} = \langle w_i w_j \rangle = \tfrac{1}{2}(\sigma_i^2 + \sigma_j^2 - f_{i,j}), \tag{19}$$

with $w_i = w(\boldsymbol{r}_i)$ being the wavefront phase at position $\boldsymbol{r}_i$, $\sigma_i^2 = \mathrm{Var}(w_i)$, and $f_{i,j} = f(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$ being the structure function between wavefront samples i and j. The wavefront variances (thus the covariance) are not defined for pure Kolmogorov statistics but can be defined by other models of the turbulence such as the von Kármán model. Nevertheless, any structure function f can be used by our algorithm: in case the variance is undefined, we will show that the σ_i^2's appear as free parameters and that choosing suitable variance values is not a problem.

B. Generation of Outermost Values

The first point to address is the initialization of the mid-point recursion, that is, the generation of the four outermost corner values. Lane *et al.* [23] used six random values to generate the four initial corners. It is however required to use exactly the same number of random values $\boldsymbol{u}$ as there are wavefront samples in $\boldsymbol{w}$; otherwise the corresponding linear operator $\mathbf{K}$ cannot be invertible. This is possible by slightly modifying their original algorithm.

The four initial wavefront values (Fig. 3) have the following covariance matrix:

$$\mathbf{C}_{\mathrm{out}} = \begin{pmatrix} c_0 & c_1 & c_2 & c_1 \\ c_1 & c_0 & c_1 & c_2 \\ c_2 & c_1 & c_0 & c_1 \\ c_1 & c_2 & c_1 & c_0 \end{pmatrix}, \quad \text{with} \quad \begin{cases} c_0 = \sigma^2, \\ c_1 = \sigma^2 - f(D)/2, \\ c_2 = \sigma^2 - f(\sqrt{2}D)/2, \end{cases}$$

where σ^2 is the variance (assumed to be the same) of the four initial phases and where D is the distance between points 1 and 2 (see Fig. 3). Having the same variances σ^2 for the four outermost wavefront phases seems natural since none of these points play a particular role. For the four outer wavefront samples, the matrix of eigenvectors of $\mathbf{C}_{\mathrm{out}}$ is

$$\mathbf{Z}_{\mathrm{out}} = \begin{pmatrix} 1/2 & -1/2 & 0 & 1/\sqrt{2} \\ 1/2 & 1/2 & -1/\sqrt{2} & 0 \\ 1/2 & -1/2 & 0 & -1/\sqrt{2} \\ 1/2 & 1/2 & 1/\sqrt{2} & 0 \end{pmatrix}.$$

Note that the eigenvectors (columns) defined on these four samples are (in order) *piston*, *waffle* [7], *tip*, and *tilt*. The eigenvalues are

$$\boldsymbol{\lambda}_{\mathrm{out}} = \begin{pmatrix} c_0 + 2c_1 + c_2 \\ c_0 - 2c_1 + c_2 \\ c_0 - c_2 \\ c_0 - c_2 \end{pmatrix} = \begin{pmatrix} 4\sigma^2 - f(D) - f(\sqrt{2}D)/2 \\ f(D) - f(\sqrt{2}D)/2 \\ f(\sqrt{2}D)/2 \\ f(\sqrt{2}D)/2 \end{pmatrix}.$$

In the case of pure Kolmogorov statistics, σ^2 must be chosen so that $\mathbf{K}$ is invertible. This is achieved if the eigenvalue of the *piston*-like mode is strictly positive; hence

$$\sigma^2 > f(D)/4 + f(\sqrt{2}D)/8.$$

We have chosen σ^2 so that the smallest covariance, which is $c(\sqrt{2}D)$ between the most remote points, is exactly zero,

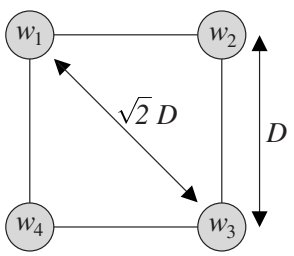

Fig. 3. The four initial values for wavefront generation, at the corners of the support.

$$\sigma^2 = \tfrac{1}{2} f(\sqrt{2}D). \tag{20}$$

Of course, when a von Kármán model of turbulence is chosen, both σ^2 and f are fixed by the model; Eq. (20) is to be used only for the Kolmogorov case.

A possible expression for the operator $\mathbf{K}_{\text{out}}$, such that $\mathbf{C}_{\text{out}} = \mathbf{K}_{\text{out}} \cdot \mathbf{K}_{\text{out}}^{\mathrm{T}}$, is

$$\mathbf{K}_{\text{out}} = \frac{1}{2}\begin{pmatrix} a & -b & -c & 0 \\ a & b & 0 & -c \\ a & -b & c & 0 \\ a & b & 0 & c \end{pmatrix}, \tag{21}$$

with

$$a = \sqrt{4\sigma^2 - f(D) - f(\sqrt{2}D)/2},$$

$$b = \sqrt{f(D) - f(\sqrt{2}D)/2},$$

$$c = \sqrt{f(\sqrt{2}D)},$$

from which $\mathbf{K}_{\text{out}}^{-1}$ is

$$\mathbf{K}_{\text{out}}^{-1} = \frac{1}{2}\begin{pmatrix} 1/a & 1/a & 1/a & 1/a \\ -1/b & 1/b & -1/b & 1/b \\ -2/c & 0 & 2/c & 0 \\ 0 & -2/c & 0 & 2/c \end{pmatrix}. \tag{22}$$

The operator $\mathbf{K}_p$ in Eq. (16) is obtained simply from $\mathbf{K}_{\text{out}}$. Indeed, $\mathbf{K}_p$ is essentially the identity matrix except for 16 nonzero coefficients corresponding to the outermost corners and which are given by $\mathbf{K}_{\text{out}}$. The same rules yield $\mathbf{K}_p^{-1}$ from $\mathbf{K}_{\text{out}}^{-1}$.

C. Generation of Wavefront Samples at Smaller Scales

Given the wavefront with a sampling step r, the mid-point algorithm generates a refined wavefront with a sampling of $r/2$ using a perturbed interpolation,

$$w_0 = \alpha_0 u_0 + \sum_{j=1}^{N_{\text{int}}} \alpha_j w_j, \tag{23}$$

where w_0 is the wavefront value at the mid-point position, $u_0 \sim \mathcal{N}(0,1)$ is a normally distributed random value, and N_{int} is the number of wavefront samples from the previous scale which are used to generate the new sample (see Fig. 4). Equation (23) comes from a generalization of the principle of the original mid-point algorithm. Since we proceed from the largest scale to smaller ones, all the operations can be done *in-place*: the value of w_0 computed according to Eq. (23) replacing that of u_0. In other words, the input and output vectors, $\boldsymbol{u}$ and $\boldsymbol{w}$, can share the same area of the computer memory. It is then immediately apparent that a random wavefront computed by this algorithm scales as $\mathcal{O}(N_{\text{int}} \times N) = \mathcal{O}(N)$ since the number of neighbors $N_{\text{int}} \sim 4$ does not depend on the number of wavefront samples N.

The $N_{\text{int}}+1$ scalars α_j have to be adjusted so that the structure function between w_0 and any of the $w_{i=1,\ldots,N_{\text{int}}}$ matches the turbulence statistics,

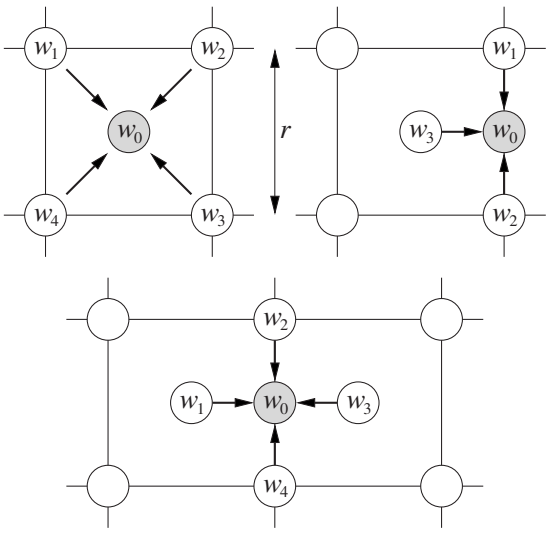

Fig. 4. Wavefront refinement. To generate a grid with cell size $r/2$, new values (in gray) are generated from wavefront values (in white) of a grid with cell size equal to r. Top left: new value from four values $r/\sqrt{2}$ apart. Top right: new edge value from three values $r/2$ apart. Bottom: new value from four values $r/2$ apart.

$$f_{i,0} \overset{\text{def}}{=} \langle (w_0 - w_i)^2 \rangle = \alpha_0^2 + \sum_{j=1}^{N_{\text{int}}} \alpha_j f_{i,j} - \sum_{1 \le j < k \le N_{\text{int}}} \alpha_j \alpha_k f_{j,k} + \left(1 - \sum_{k=1}^{N_{\text{int}}} \alpha_k\right)\left(\sigma_i^2 - \sum_{j=1}^{N_{\text{int}}} \alpha_j \sigma_j^2\right). \tag{24}$$

Note that, to obtain this equation, we have accounted for the fact that since $u_0 \sim \mathcal{N}(0,1)$ and $w_{j=1,\ldots,N_{\text{int}}}$ are uncorrelated, then $\langle u_0^2 \rangle = 1$ and $\langle u_0 w_{j=1,\ldots,N_{\text{int}}} \rangle = 0$. System (24) gives N_{int} equations, whereas there are $N_{\text{int}}+1$ unknown parameters $\{\alpha_0, \ldots, \alpha_{N_{\text{int}}}\}$: an additional constraint is needed.

In the original mid-point algorithm, Lane *et al.* [23] chose to normalize the sum of the interpolation coefficients and use the constraint that $\Sigma_{j=1}^{N_{\text{int}}} \alpha_j = 1$. In that case, Eq. (24) simplifies and the coefficients are obtained by solving

$$f_{i,0} = \alpha_0^2 + \sum_{j=1}^{N_{\text{int}}} \alpha_j f_{i,j} - \sum_{1 \le j < k \le N_{\text{int}}} \alpha_j \alpha_k f_{j,k} \quad \text{s.t.} \quad \sum_{j=1}^{N_{\text{int}}} \alpha_j = 1. \tag{25}$$

Note that all the variances σ_j^2 are implicit with this constraint.

We consider here another constraint which is to have the same variance, say σ^2, for all the wavefront samples. In other words, we consider a wavefront with stationary (shift-invariant) statistical properties. This is justified by our objective to reconstruct phase corrugations in several layers for atmospheric tomography. Indeed, since the beams coming from different directions in the field of view are not superimposed in the layers, this condition allows the wavefront statistics to remain the same for all the beams. With this choice, the additional equation is provided by $\langle w_0^2 \rangle = \sigma^2$ and the interpolation coefficients $\{\alpha_0, \ldots, \alpha_{N_{\text{int}}}\}$ are obtained by solving the system of $N_{\text{int}}+1$ equations,

$$f_{i,0} = \alpha_0^2 + \sum_{j=1}^{N_{\text{int}}} \alpha_j f_{i,j} - \sum_{1\le j<k\le N_{\text{int}}} \alpha_j \alpha_k f_{j,k} + \sigma^2 \left(1 - \sum_{j=1}^{N_{\text{int}}} \alpha_j\right)^2 \quad \text{for } i = 1, \ldots, N_{\text{int}},$$

$$\sigma^2 = \alpha_0^2 + \sigma^2 \left(\sum_{j=1}^{N_{\text{int}}} \alpha_j\right)^2 - \sum_{1\le j<k\le N_{\text{int}}} \alpha_j \alpha_k f_{j,k}.$$

The system can be further simplified to

$$\sum_{j=1}^{N_{\text{int}}} (2\sigma^2 - f_{i,j})\alpha_j = 2\sigma^2 - f_{i,0} \quad \text{for } i = 1, \ldots, N_{\text{int}},$$

$$\alpha_0^2 = \left[1 - \left(\sum_{j=1}^{N_{\text{int}}} \alpha_j\right)^2\right]\sigma^2 + \sum_{1\le j<k\le N_{\text{int}}} \alpha_j \alpha_k f_{j,k}, \tag{26}$$

where the first N_{int} equations form a linear system which must be solved to obtain $\alpha_{j=1,\ldots,N_{\text{int}}}$ and where substituting these values in the last equation yields the value of α_0. It is worth noting that by using the covariances instead of the structure function, the system in Eq. (26) is equivalent to

$$\sum_{j=1}^{N_{\text{int}}} C_{i,j}\alpha_j = C_{0,i} \quad \text{for } i = 1, \ldots, N_{\text{int}},$$

$$\alpha_0^2 = \sigma^2 - \sum_{j=1}^{N_{\text{int}}} C_{0,j}\alpha_j. \tag{27}$$

The expressions for the interpolations coefficients for the different cases illustrated by Fig. 4 are derived in Appendix A. To assess the accuracy of the statistics approximated by the fractal operator, we have computed the structure function of phase screens $\boldsymbol{w}$ computed by our implementation of the mid-point algorithm, i.e., as $\boldsymbol{w} = \mathbf{K}\cdot\boldsymbol{u}$ with $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Figure 5 shows that the two-dimensional (2D) structure function is almost isotropic and demonstrates good agreement of our approximation to the theoretical law.

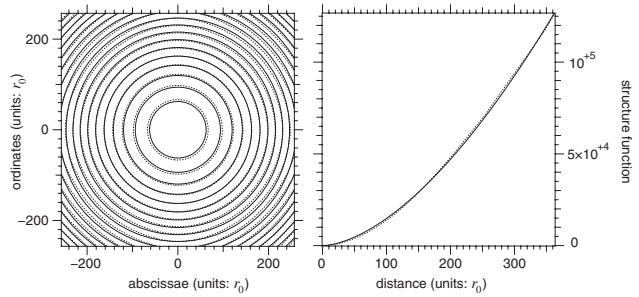

Fig. 5. Structure function. *Left:* 2D isocontours; *right:* one-dimensional profile computed by radial averaging. *Solid lines:* Kolmogorov law $6.88\times(r/r_0)^{5/3}$; *dotted lines:* average of 1000 structure functions generated with the mid-point method.

D. Inverse Operator

According to the factorization in Eq. (16), the inverse of $\mathbf{K}$ is

$$\mathbf{K}^{-1} = \mathbf{K}_p^{-1} \cdot \cdots \cdot \mathbf{K}_2^{-1} \cdot \mathbf{K}_1^{-1}. \tag{28}$$

In Subsection 3.B, the inverse of the outermost operator $\mathbf{K}_p$ has been derived and shown to be sparse—see Eq. (22). To compute the $\mathbf{K}_j^{-1}$'s for the inner scales ($j<p$), it is sufficient to solve Eq. (23) for u_0, which trivially yields

$$u_0 = \frac{1}{\alpha_0}\left(w_0 - \sum_{j=1}^{N_{\text{int}}} \alpha_j w_j\right), \tag{29}$$

where $\{w_1, \ldots, w_{N_{\text{int}}}\}$ are the neighbors of w_0 (Fig. 4). Since in Eq. (29), the u_j's only depend on the w_j's, the $\mathbf{K}_j^{-1}$'s can be applied in any order. However, by proceeding from the smallest scales toward the largest ones as in Eq. (28), the operator $\mathbf{K}^{-1}$ can be performed *in-place*. This property may be important to avoid memory page faults and to speed up the computation. Finally, from Eqs. (21) and (29), it is clear that applying the $\mathbf{K}_j^{-1}$'s requires exactly as many operations as for the $\mathbf{K}_j$'s and that computing $\mathbf{K}^{-1}\cdot\boldsymbol{u}$ requires $\mathcal{O}(N)$ operations.

E. Transpose Operator

Iterating from the smallest scale to the largest one, it is easy to derive an algorithm to apply the transpose operator $\mathbf{K}^{\mathrm{T}} = \mathbf{K}_p^{\mathrm{T}} \cdot \cdots \cdot \mathbf{K}_2^{\mathrm{T}} \cdot \mathbf{K}_1^{\mathrm{T}}$ to a given vector. The following algorithm computes $\boldsymbol{z} = \mathbf{K}^{\mathrm{T}}\cdot\boldsymbol{v}$ for any input vector $\boldsymbol{v}$:

copy input vector: $\boldsymbol{z} \leftarrow \boldsymbol{v}$
from the smallest scale to the largest scale, do
  for $j = 1, \ldots, N_{\text{int}}$ do
    $z_j \leftarrow z_j + \alpha_j z_0$
  done
  $z_0 \leftarrow \alpha_0 z_0$
done
apply $\mathbf{K}_{\text{out}}^{\mathrm{T}}$ at the largest scale of $\boldsymbol{z}$
return $\boldsymbol{z}$.

It is important to note that the loop must be performed *in-place* for the algorithm to work. From the structure of this algorithm, it is clear that the multiplication of a vector by the transpose operator is performed in $\mathcal{O}(N)$ operations.

F. Inverse Transpose Operator

The operator $\mathbf{K}^{\mathrm{T}} = \mathbf{K}_1^{-\mathrm{T}} \cdot \mathbf{K}_2^{-\mathrm{T}} \cdot \cdots \cdot \mathbf{K}_p^{-\mathrm{T}}$ works from the largest scale to the smallest one. The following algorithm computes $\boldsymbol{z} = \mathbf{K}^{-\mathrm{T}}\cdot\boldsymbol{v}$ for any input vector $\boldsymbol{v}$:

copy input vector: $\boldsymbol{z} \leftarrow \boldsymbol{v}$
apply $\mathbf{K}_{\text{out}}^{-\mathrm{T}}$ at the largest scale of $\boldsymbol{z}$
from the largest scale to the smallest scale, do
  $z_0 \leftarrow z_0/\alpha_0$
  for $j = 1, \ldots, N_{\text{int}}$ do
    $z_j \leftarrow z_j - \alpha_j z_0$
  done
done
return $\boldsymbol{z}$.

Again, the operation can be done *in-place* (the copy of the input vector $\boldsymbol{v}$ is only required to preserve its contents if needed), and the number of operations is $\mathcal{O}(N)$.

4. PRECONDITIONING

Preconditioning is a general means to speed up the convergence of iterative optimization methods [15] such as the PCG algorithm described in Fig. 1. Preconditioning is generally introduced as finding an invertible matrix $\mathbf{M}$ such that the spectral properties of $\mathbf{M}^{-1}\cdot\mathbf{A}$ are more favorable than that of $\mathbf{A}$ (i.e., lower condition number and/or more clustered eigenvalues), and then the transformed system,

$$\mathbf{M}^{-1}\cdot\mathbf{A}\cdot\boldsymbol{x}=\mathbf{M}^{-1}\cdot\boldsymbol{b}, \tag{30}$$

which has the same solution as the original system $\mathbf{A}\cdot\boldsymbol{x}=\boldsymbol{b}$ can be solved in a much fewer iterations. In this section, we consider different means for preconditioning the phase restoration problem: explicit change of variables and diagonal preconditioners.

A. Fractal Operator as a Preconditioner

Preconditioning is also equivalent to an *implicit* linear change of variables [12]: using the preconditioner $\mathbf{M}=\mathbf{C}^{\mathrm{T}}\cdot\mathbf{C}$ in the algorithm in Fig. 1 is the same as using the (unpreconditioned) CG algorithm to solve the optimization problem with respect to $\hat{\boldsymbol{x}}=\mathbf{C}\cdot\boldsymbol{x}$. Following this we have considered using our statistically independent modes to solve the problem with respect to the variables $\boldsymbol{u}=\mathbf{K}^{-1}\cdot\boldsymbol{w}$. In this case, it is however advantageous in terms of the number of floating points operations to use an *explicit* change of variables and to directly solve the problem for $\boldsymbol{u}$ rather than for $\boldsymbol{w}$ with a preconditioner $\mathbf{M}=\mathbf{K}^{-\mathrm{T}}\cdot\mathbf{K}^{-1}$. Introducing this change of variable in Eq. (7) and using Eq. (15), the system to solve becomes

$$(\mathbf{K}^{\mathrm{T}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}\cdot\mathbf{K}+\mathbf{I})\cdot\boldsymbol{u}=(\mathbf{K}^{\mathrm{T}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\boldsymbol{d}). \tag{31}$$

After $\boldsymbol{u}$ is found by the iterative algorithm, the restored wavefront is given $\boldsymbol{w}=\mathbf{K}\cdot\boldsymbol{u}$. We expect improvements in the convergence of the iterative method by using $\boldsymbol{u}$ instead of $\boldsymbol{w}$ because this yields an *a priori* covariance matrix equal to the identity matrix [32]. Improved speedup may be still possible by using a preconditioner on $\boldsymbol{u}$ as we discuss in the following.

B. Diagonal Preconditioners

Diagonal preconditioners may not be the most efficient ones but are very cheap to use [15] and are thus considered here. When the variable $\boldsymbol{x}$ in Eq. (8) follows known statistics, an optimal preconditioner $\mathbf{M}$ can be computed so that $\mathbf{M}^{-1}\cdot\mathbf{A}$ is, on average, as close as possible to the identity matrix. This *closeness* can be measured in two different spaces: in the data space or in the parameter space.

In the *data space*, this criterion is written as

$$\mathbf{M}=\arg\min_{\mathbf{M}}\langle\|\mathbf{A}\cdot\boldsymbol{x}-\mathbf{M}\cdot\boldsymbol{x}\|^2\rangle\Leftrightarrow 0=\frac{\partial\langle\|(\mathbf{A}-\mathbf{M})\cdot\boldsymbol{x}\|^2\rangle}{\partial\mathbf{M}}$$
$$=2(\mathbf{M}-\mathbf{A})\cdot\langle\boldsymbol{x}\cdot\boldsymbol{x}^{\mathrm{T}}\rangle\Leftrightarrow\mathbf{M}\cdot\mathbf{C}_{\boldsymbol{x}}=\mathbf{A}\cdot\mathbf{C}_{\boldsymbol{x}}, \tag{32}$$

where $\mathbf{C}_{\boldsymbol{x}}\overset{\mathrm{def}}{=}\langle\boldsymbol{x}\cdot\boldsymbol{x}^{\mathrm{T}}\rangle$ is the covariance matrix of $\boldsymbol{x}$. Of course, if $\mathbf{M}$ is allowed to be any matrix and since $\mathbf{C}_{\boldsymbol{x}}$ has a full rank, the solution to Eq. (32) is $\mathbf{M}=\mathbf{A}$. However, for a diagonal preconditioner, $\mathbf{M}=\mathrm{diag}(\boldsymbol{m})$, only the diagonal terms of Eq. (32) have to be considered; this yields

$$\mathbf{M}=\mathrm{diag}(\boldsymbol{m})=\mathrm{diag}(\mathbf{A}\cdot\mathbf{C}_{\boldsymbol{x}})\cdot\mathrm{diag}(\mathbf{C}_{\boldsymbol{x}})^{-1}. \tag{33}$$

For $\boldsymbol{x}=\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ then $\mathbf{C}_{\boldsymbol{x}}=\mathbf{I}$, and Eq. (32) simplifies to

$$\mathbf{M}=\mathrm{diag}(\mathbf{A}), \tag{34}$$

which is the well known Jacobi preconditioner [15].

Taking $\mathbf{Q}\overset{\mathrm{def}}{=}\mathbf{M}^{-1}$ and minimizing the statistical distance in the *parameter space* yields

$$\mathbf{Q}=\arg\min_{\mathbf{Q}}\langle\|\mathbf{Q}\cdot\mathbf{A}\cdot\boldsymbol{x}-\boldsymbol{x}\|^2\rangle\Leftrightarrow 0=\frac{\partial\langle\|\mathbf{Q}\cdot\mathbf{A}\cdot\boldsymbol{x}-\boldsymbol{x}\|^2\rangle}{\partial\mathbf{Q}}$$
$$=2(\mathbf{Q}\cdot\mathbf{A}-\mathbf{I})\cdot\mathbf{C}_{\boldsymbol{x}}\cdot\mathbf{A}^{\mathrm{T}}\Leftrightarrow\mathbf{Q}\cdot\mathbf{A}\cdot\mathbf{C}_{\boldsymbol{x}}\cdot\mathbf{A}^{\mathrm{T}}=\mathbf{C}_{\boldsymbol{x}}\cdot\mathbf{A}^{\mathrm{T}}. \tag{35}$$

For a diagonal preconditioner, $\mathbf{Q}=\mathrm{diag}(\boldsymbol{q})$, only the diagonal terms of Eq. (35) have to be considered; hence

$$\mathbf{Q}=\mathrm{diag}(\boldsymbol{q})=\mathrm{diag}(\mathbf{C}_{\boldsymbol{x}}\cdot\mathbf{A}^{\mathrm{T}})\cdot\mathrm{diag}(\mathbf{A}\cdot\mathbf{C}_{\boldsymbol{x}}\cdot\mathbf{A}^{\mathrm{T}})^{-1}. \tag{36}$$

Finally, when $\boldsymbol{x}=\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$,

$$Q_{i,i}=\frac{A_{i,i}}{\sum_j A_{i,j}^2},\quad Q_{i,j\neq i}=0. \tag{37}$$

In contrast to the Jacobi preconditioner, the *optimal* preconditioner $\mathbf{Q}$ is expensive to compute since every element of matrix $\mathbf{A}$ must be evaluated to evaluate the denominator. This however has to be done only once and for all for a given left hand side matrix $\mathbf{A}$. The improvements given by the diagonal preconditioners in Eqs. (34) and (37) are compared in the next section.

5. SIMULATIONS AND RESULTS

A. Summary of the Various Possibilities

Our previous study gives rise to six different possibilities to solve Eq. (8). The first method is based on Eq. (15) to iteratively solve for $\boldsymbol{w}$,

$$(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}+\mathbf{K}^{-\mathrm{T}}\cdot\mathbf{K}^{-1})\cdot\boldsymbol{w}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\boldsymbol{d}, \tag{38}$$

using the sparse model matrix $\mathbf{S}$ and the fractal operators $\mathbf{K}^{-1}$ and $\mathbf{K}^{-\mathrm{T}}$ introduced in Subsection 2.E and Sec. 3. Although the *a priori* covariance matrix of $\boldsymbol{w}$ is not the identity, we have tried two other methods by assessing the speedup brought by each of the two diagonal preconditioners defined in Eqs. (34) and (37), with $\mathbf{A}=\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}+\mathbf{K}^{-\mathrm{T}}\cdot\mathbf{K}^{-1}$.

Solving the problem in our statistically independent modes corresponds to a fourth method, requiring one to iteratively solve

$$(\mathbf{K}^{\mathrm{T}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}\cdot\mathbf{K}+\mathbf{I})\cdot\boldsymbol{u}=\mathbf{K}^{\mathrm{T}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\boldsymbol{d} \quad (39)$$

for $\boldsymbol{u}$ and then do $\boldsymbol{w}=\mathbf{K}\cdot\boldsymbol{u}$. For the two last methods, we use with Eq. (39), one of the two preconditioners defined in Eqs. (34) and (37) with $\mathbf{A}=\mathbf{K}^{\mathrm{T}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}\cdot\mathbf{K}+\mathbf{I}$. In this case, $\mathbf{C}_{\boldsymbol{u}}=\mathbf{I}$ so we expect somewhat faster convergence.

B. Comparison of the Rates of Convergence

When comparing the efficiency of the six different possibilities, we need to take into account that the number of floating point operations may be different for each of them. The aim is not to derive an accurate number of operations, which would depend on the specific implementation of the algorithms, but rather to get a general estimate. For instance, the dependence of $\mathbf{K}$ on r_0 can be factorized out and included in operator $\mathbf{C}_{\boldsymbol{n}}$ with no extra computational cost. This kind of optimization was not considered here. As detailed in Appendix B, the number of operations is marginally increased by the preconditioning and does not depend on which variables ($\boldsymbol{w}$ or $\boldsymbol{u}$) are used when starting from an arbitrary initial vector. A small difference appears only when starting the algorithms with an initial zero vector, as summarized in Table 1. For wavefront reconstruction, when comparing the total number of operations, N_{ops}, for a given number of (P)CG iterations, N_{iter}, such that $N_{\mathrm{iter}}\geq 1$, we will use the following equations:

$$N_{\mathrm{CG}}^{\mathrm{ops}}\sim(N_{\mathrm{overhead}}+33N_{\mathrm{CG}}^{\mathrm{iter}})N,$$

$$N_{\mathrm{PCG}}^{\mathrm{ops}}\sim(N_{\mathrm{overhead}}+34N_{\mathrm{PCG}}^{\mathrm{iter}})N, \quad (40)$$

where $N_{\mathrm{overhead}}=4$ when working with variable $\boldsymbol{w}$, and $N_{\mathrm{overhead}}=10$ when explicitly working with variable $\boldsymbol{u}$.

In order to assess the speed of the reconstruction, we have tested the different wavefront reconstruction algorithms on a number of different conditions. For every simulation, the wavefront sensor sampling is such that the size of the Shack–Hartmann subaperture is equal to the Fried parameter r_0. A wavefront is first generated by applying the fractal operator $\mathbf{K}$ to a vector of normally distributed random values as in Subsection 3.C. The measurements are then estimated using the current wavefront sensor model, $\mathbf{S}$, and a stationary uncorrelated random noise $\boldsymbol{n}$ is added to the simulated slopes in accordance with Eq. (1). The noise level is given by its standard deviation σ_{noise} in radians per subaperture, where the radians here correspond to phase differences between the edges of the subapertures. At each iteration of the algorithm, the residual wavefront is computed as the difference between the current solution and the initial wavefront. The root mean squared error of the residual wavefront is computed over the pupil, with the piston removed. The piston mode is the only removed mode. A central obscuration is always introduced, with a diameter 1/3 the diameter of the pupil.

The graphs presented are for two AO systems of sizes 65×65 (cf. Figs. 6–9) and 257×257 (cf. Figs. 10 and 11). Several levels of noise from 1 rad/subaperture down to 0.05 rad/subaperture are examined. They correspond to the levels of photon noise obtained with ~7 to ~3000 detected photons per subaperture. On each curve, the six algorithms are compared. All the curves plot the median value obtained for 100 simulations under the same conditions. The different algorithms were applied to the same simulated wavefronts and sensor data. The graphs have been plotted assuming a number of floating point operations given by Eq. (40), where here the numbers of unknowns are $N=4225$ and 66,049 for AO systems 65×65 and 257×257, respectively. Various observations can be drawn from these curves as discussed in what follows.

Solving by using $\boldsymbol{w}$'s as unknowns is much slower than using $\boldsymbol{u}$'s by more than 1 order of magnitude for a 65×65 system, and 2 orders of magnitude for 257×257. This demonstrates a stunning efficiency for the fractal operator used as a preconditioner. With $\boldsymbol{w}$, the algorithm

Table 1. Number of Operations Involved in CG and PCGs Applied to the Wavefront Restoration Problem Solved by Our Algorithm[a]

Algorithm Step	Floating Point Operation
Initialization: general case	~25N
Zero initial vector in $\boldsymbol{u}$ space	~12N
Zero initial vector in $\boldsymbol{w}$ space	~6N
First CG iteration	~31N
Any subsequent CG iteration	~33N
Total after $N_{\mathrm{iter}}\geq1$ iterations	~$(23+33N_{\mathrm{iter}})N$
First PCG iteration	~32N
Any subsequent PCG iteration	~34N
Total after $N_{\mathrm{iter}}\geq1$ iterations	~$(23+34N_{\mathrm{iter}})N$

[a]The integers N and N_{iter} are, respectively, the number of unknowns and number of iterations. For a reconstruction, we assume an initial null guess in the initialization step: in this case the number of operations at this step is reduced to ~6N or ~12N when, respectively, $\boldsymbol{w}$ or $\boldsymbol{u}$ is used as unknown.

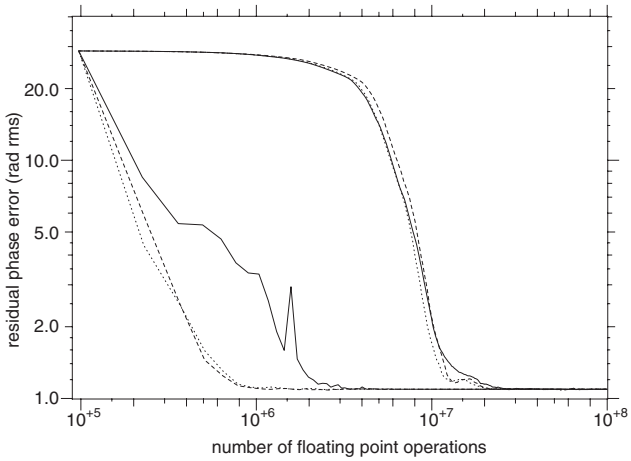

Fig. 6. Phase error as a function of the number of operations. Curves are the median value of 100 simulations with $D/r_0=65$, $\sigma_{\mathrm{noise}}=1$ rad/subaperture and r_0 has the same size as one subaperture. Solid curves are for CG, dashed curves are for PCG with Jacobi preconditioner, and dotted curves are for PCG with optimal diagonal preconditioner. Thin curves are for (P)CG onto the wavefront samples $\boldsymbol{w}$, whereas thick curves are for (P)CG onto the wavefront generator $\boldsymbol{u}$.

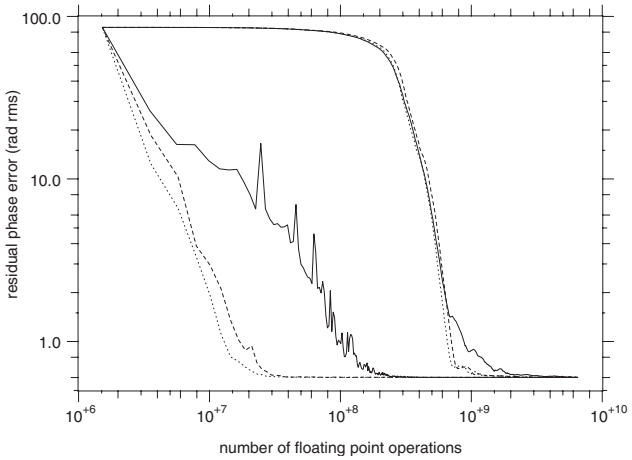

Fig. 7. Same as Fig. 6 but for $\sigma_{\text{noise}}=0.5$ rad/subaperture.

does not show any improvement of the residual error for a long time before finding its way toward the solution. In contrast, the very first steps with $\boldsymbol{u}$ already show a tremendous reduction in the residual error. For instance, this feature is critical if the number of iterations is to be limited to a fixed value as could be the case in a closed-loop.

Using the Jacobi or optimal diagonal preconditioners has no same effect when working in the $\boldsymbol{w}$ or $\boldsymbol{u}$ space.

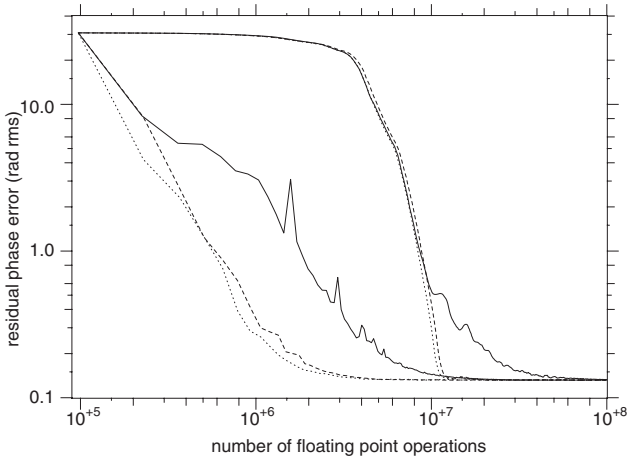

Fig. 8. Same as Fig. 6 but for $\sigma_{\text{noise}}=0.1$ rad/subaperture.

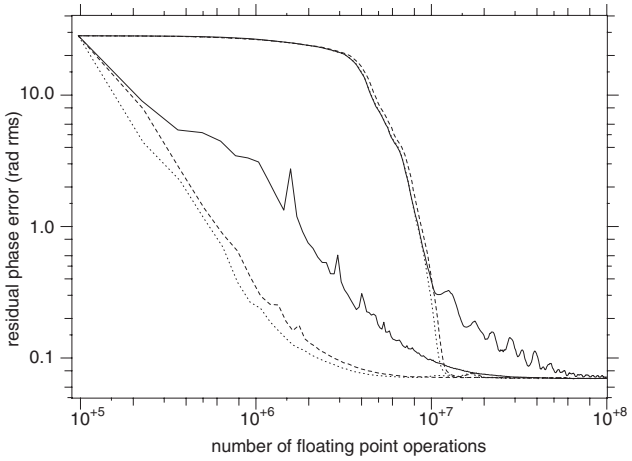

Fig. 9. Same as Fig. 6 but for $\sigma_{\text{noise}}=0.05$ rad/subaperture.

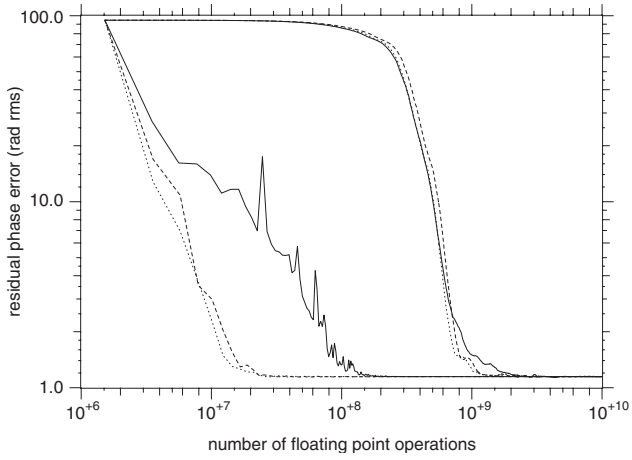

Fig. 10. Same as Fig. 6 but for $D/r_0=257$ and $\sigma_{\text{noise}}=1$ rad/subaperture.

When solving for $\boldsymbol{w}$, the preconditioners are only useful at the very end of the convergence, mainly in the case of a high signal-to-noise ratio. Thus they are not very helpful to reduce the computational load. In contrast, the effect of the diagonal preconditioners is very effective from the beginning when working with $\boldsymbol{u}$. We may notice that the difference between the two diagonal preconditioners is significant but not critical. The optimal diagonal preconditioner yields slightly faster convergence.

When σ_{noise} decreases, the convergence of the two fastest methods takes longer to reach a lower level of residual errors but the rate of convergence remains steady. This is analyzed in more detail in the next section.

C. Number of Iterations

From the previous section, we now consider only the fastest method, using $\boldsymbol{u}$ as both unknowns and the optimal diagonal preconditioner. The aim here is to assess the number of iterations needed to restore the wavefront. As in the previous section, we consider one subaperture per r_0, so the variance of the incoming wavefronts increases with the size of the system. Figure 12 shows how the residual phase variance decreases at each iteration for various configurations of the system, in sizes (33×33, 65

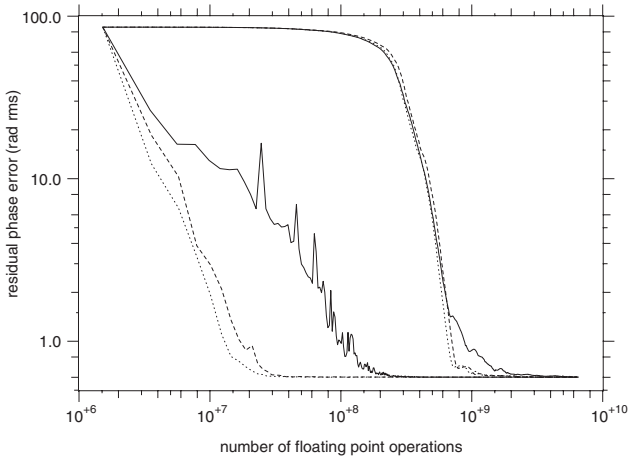

Fig. 11. Same as Fig. 6 but for $D/r_0=257$ and $\sigma_{\text{noise}}=0.5$ rad/subaperture.

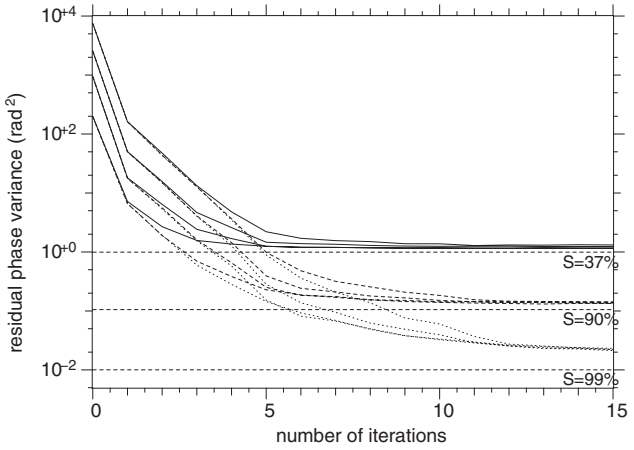

Fig. 12. Decrease in the residual phase variance as a function of the number of iterations when using $\boldsymbol{u}$ as unknowns and optimal diagonal preconditioner. Each curve is the median value of 100 simulations. Three sets of curves are plotted for different values of σ^2_{noise}: 1 (solid), 0.09 (dashed), and 0.01 (dotted) rad$^2/r_0$. In each set of curves, the size of the system increases from bottom to top: 32, 64, 128, and 256 subapertures along the diameter of the pupil. Levels of Strehl ratios are indicated. The curves show that five to ten iterations are enough in most cases for a full reconstruction.

$\times 65$, 129×129, 257×257) and in noise levels ($\sigma^2_{\text{noise}}=1$, 0.09, and 0.01 rad$^2/r_0$). In the first iterations, we can see that the behavior of the algorithm does not depend on the signal-to-noise ratio. In contrast, the final value obtained does not depend on the size of the system. Strehl levels corresponding to the residual phase variance are indicated. The curves show that, whatever is the size of the system, only five to ten iterations are enough for a reconstruction starting from zero.

In order to remove the effect of starting from different initial phase variances, the same curves have been normalized by the initial variance in Fig. 13. We can see that the descent of the FRiM follows the same path for all the simulations and is stopped at different values of the final

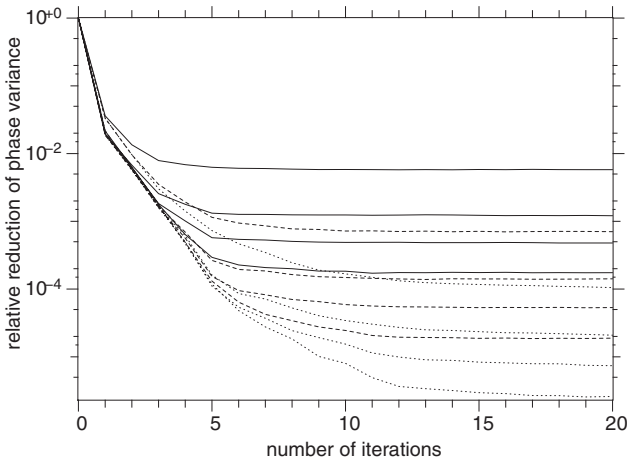

Fig. 13. The same curves as those in Fig. 12 are plotted here, normalized by the initial variance of the phase. This shows a high relative attenuation ($\sim 1/40$) after the first iteration, in any configuration. In each set of curves: $\sigma^2_{\text{noise}}=1$ (solid), 0.09 (dashed), and 0.01 (dotted) rad$^2/r_0$, the size of the system increases from top to bottom: 32, 64, 128, and 256 subapertures along the diameter of the pupil.

variance, which depends on the signal-to-noise ratio. Along this path, the variance is already reduced by a factor of $\sim 1/50$ at the first iteration, $\sim 1/170$ at the second iteration, and more than $\sim 10^{-4}$ at iteration 6. This steep descent will be an asset in a closed loop.

6. CONCLUSION

We have introduced the FRiM, a new minimum variance iterative algorithm for fast wavefront reconstruction and fast control of an adaptive optics (AO) system. Combining fast regularization and efficient preconditioning, regularized wavefront reconstruction by the FRiM is an $\mathcal{O}(N)$ process, where N is the number of wavefront samples.

The FRiM takes advantage of the sparsity of the model matrix $\mathbf{S}$ of wavefront sensors (or interaction matrices) and makes use of a fractal operator $\mathbf{K}$ for a fast computation of the priors. Based on a generalization of the midpoint algorithm [23], $\mathbf{K}$ is not sparse but is implemented so that it only requires $\mathcal{O}(N)\simeq 6N$ operations. Our modifications with respect to the original algorithm allow the operator to be invertible and the generated wavefront to be stationary. We have derived algorithms for computing $\mathbf{K}^{-1}$, $\mathbf{K}^{\mathrm{T}}$, and $\mathbf{K}^{-\mathrm{T}}$ in the same number of floating point operations. In our simulations, we consider a modified Kolmogorov law but any stationary structure function or covariance can be implemented in our approach. The property of stationarity is expected to be helpful for turbulence tomography.

Another breakthrough comes from the efficiency of the fractal operator when used as a preconditioner. Combining a fractal change of variables and an optimal diagonal preconditioner, we were able to reduce the number of iterations in the range of 5–10 for a full wavefront reconstruction whatever is the size of the AO system. The exact number of iterations mainly depends on the signal-to-noise ratio of the measurements.

It is beyond this work to compare with all the other methods currently studied in response to the huge increase in the number of degrees of freedom for the AO system on ELTs. Nevertheless, we can easily compare to standard vector-matrix multiplication (VMM). Assuming uncorrelated noise, the simulations show that the number of operations with the FRiM is $N_{\text{ops}}\sim(23+34N_{\text{iter}})N$, where the number of PCG iterations is $N_{\text{iter}}\lesssim 10$ for any number of degrees of freedom N. For up to $N=1.3\times 10^4$ degrees of freedom (i.e., $D/r_0\leq 128$), one wavefront estimation (from scratch) involves $\lesssim 6\times 10^6$ operations, that is, a bandwidth of ~ 500 Hz for a machine capable of 3 Gflops which is typical of current workstations. Conversely, conventional (non-sparse) matrix multiplication would require $\sim 4N^2\sim 7\times 10^8$ operations to compute the wavefront: our method is more than 100 times faster. Furthermore, since the operations can be done *in-place*, it is expected that the computation with the FRiM could all be done in cache memory.

For simulating very large AO systems (e.g., atmospheric tomography on ELTs), the speed of the current version of the FRiM is already an asset. For real time control of AO systems, the FRiM algorithm can be parallelized to run on several central processing units. Being an iterative method (unlike Fourier methods), the FRiM

could be used to improve the estimation of the wavefront from any pieces of new data as soon as it becomes available. Hence, the FRiM does not need all the measurements in a closed-loop system. A fast iterative method that gives intermediate results with only a part of the measurements opens the way to new control approaches for reducing the effect of the delay. A further advantage of the FRiM is that it accounts for the statistics of the turbulence which not only yields a better estimation of the residual phase [33] but also helps one to disentangle ambiguities such as unseen modes in atmospheric tomography. In this paper, we assume that the structure function is perfectly known. Béchet [34] showed that it is sufficient to not overestimate r_0 by more than a factor of ~ 2 to benefit from the advantages of taking into the priors.

The next step of this work is to extend the theory to a closed loop and to assess the performances and the properties of the algorithm in this regime. Since the wavefront is not allowed to change a lot from one step of the AO loop to the other, the algorithm will always starts close to the solution: the number of iterations is expected to be yet lower. This study is not yet completed but preliminary results have proved the efficiency of the FRiM in the case of closed-loop AO [35,36].

APPENDIX A: DERIVATION OF THE INTERPOLATION COEFFICIENTS

In this appendix, we detail the computation of the interpolation coefficients involved in the different configurations shown in Fig. 4. Denoting by r the step size in the grid before the refinement, the distances between the points considered in this refinement step are $\sqrt{2}r$, r, $r/\sqrt{2}$, and $r/2$ (Fig. 4). Hence the only covariances required in our computations are

$$c_0 = c(0) = \sigma^2,$$

$$c_1 = c(r/2) = \sigma^2 - f(r/2)/2,$$

$$c_2 = c(r/\sqrt{2}) = \sigma^2 - f(r/\sqrt{2})/2,$$

$$c_3 = c(r) = \sigma^2 - f(r)/2,$$

$$c_4 = c(\sqrt{2}r) = \sigma^2 - f(\sqrt{2}r)/2, \tag{A1}$$

where $c(r)$ and $f(r)$ are, respectively, the covariance and the structure function for a separation r.

1. Square Configuration

For the interpolation stage illustrated in the top-left part of Fig. 4 and according to Eq. (27), the interpolation coefficients $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are obtained by solving

$$\begin{pmatrix} c_0 & c_3 & c_4 & c_3 \\ c_3 & c_0 & c_3 & c_4 \\ c_4 & c_3 & c_0 & c_3 \\ c_3 & c_4 & c_3 & c_0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} c_2 \\ c_2 \\ c_2 \\ c_2 \end{pmatrix}.$$

Solving this linear system and plugging the solution into Eq. (27) leads to

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{c_2}{c_0 + 2c_3 + c_4},$$

$$\alpha_0 = \pm \sqrt{c_0 - \frac{4c_2^2}{c_0 + 2c_3 + c_4}}. \tag{A2}$$

Note that the sign of α_0 is irrelevant.

2. Triangle Configuration

In the original mid-point algorithm [23], the values at the edges of the support (top-right part of Fig. 4) were generated from only the two neighbors on the edge, ignoring the third interior neighbor (denoted by w_3 in the figure). Here, according to Eq. (27), the interpolation coefficients $\{\alpha_1, \alpha_2, \alpha_3\}$ for this stage are obtained by solving

$$\begin{pmatrix} c_0 & c_3 & c_2 \\ c_3 & c_0 & c_2 \\ c_2 & c_2 & c_0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_1 \\ c_1 \end{pmatrix}.$$

Solving this linear system and plugging the solution into Eq. (27) leads to

$$\alpha_1 = \alpha_2 = \frac{c_1(c_0 - c_2)}{c_0(c_0 + c_3) - 2c_2^2},$$

$$\alpha_3 = \frac{c_1(c_0 - 2c_2 + c_3)}{c_0(c_0 + c_3) - 2c_2^2},$$

$$\alpha_0 = \pm \sqrt{c_0 - \frac{c_1^2(3c_0 - 4c_2 + c_3)}{c_0(c_0 + c_3) - 2c_2^2}}. \tag{A3}$$

3. Diamond Configuration

The interpolation coefficients for the stage in the bottom part of Fig. 4 can be deduced from Eq. (A2) by replacing r with $r/\sqrt{2}$, then

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{c_1}{c_0 + 2c_2 + c_3},$$

$$\alpha_0 = \pm \sqrt{c_0 - \frac{4c_1^2}{c_0 + 2c_2 + c_3}}. \tag{A4}$$

APPENDIX B: COMPUTATIONAL BURDEN

In order to estimate the number of floating point operations, we need to carefully detail the steps of the CG method and count the number of operations involved in the multiplication by the different linear operators $\mathbf{S}$, $\mathbf{K}$, etc. Figure 1 summarizes the steps of the (PCG) algorithm [15] to solve Eq. (8).

If the unknowns are the wavefront samples, then $\boldsymbol{x} = \boldsymbol{w}$ and

$$\mathbf{A} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \mathbf{S} + \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1},$$

$$\boldsymbol{b} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \boldsymbol{d}.$$

Starting the algorithm with $\boldsymbol{w}_0$, the initial residuals write

$$\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{w}_0 = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot (\boldsymbol{d} - \mathbf{S} \cdot \boldsymbol{w}_0) - \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1} \cdot \boldsymbol{w}_0. \tag{B1}$$

If the unknowns are the wavefront generators, then $\boldsymbol{x} = \boldsymbol{u}$ and

$$\mathbf{A} = \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \mathbf{S} \cdot \mathbf{K} + \mathbf{I},$$

$$\boldsymbol{b} = \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \boldsymbol{d},$$

where $\mathbf{I}$ is the identity matrix. Starting the algorithm with $\boldsymbol{u}_0$, the initial residuals are

$$\boldsymbol{r}_0 = \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{u}_0 = \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot (\boldsymbol{d} - \mathbf{S} \cdot \mathbf{K} \cdot \boldsymbol{u}_0) - \boldsymbol{u}_0. \tag{B2}$$

Making use of possible factorizations (some of the α_i's have the same values), applying any one of the operators $\mathbf{K}$, $\mathbf{K}^{\mathrm{T}}$, $\mathbf{K}^{-1}$, or $\mathbf{K}^{-\mathrm{T}}$, involves the same number of floating point operations,

$$N_{\mathrm{ops}}(\mathbf{K}) = N_{\mathrm{ops}}(\mathbf{K}^{\mathrm{T}}) = N_{\mathrm{ops}}(\mathbf{K}^{-1}) = N_{\mathrm{ops}}(\mathbf{K}^{-\mathrm{T}}) = 6N_{\boldsymbol{u}} - 14 \sim 6N,$$

where N is the number of degrees of freedom of the system, $N_{\boldsymbol{u}} \sim N$ is the number of elements in vector $\boldsymbol{u}$, and, in our notation, $N_{\mathrm{ops}}(\mathbf{L})$ is the number of floating point operations required to apply a linear operator $\mathbf{L}$ to a vector.

Since we consider uncorrelated data noise, $\mathbf{C}_{\boldsymbol{n}}^{-1}$ is diagonal and

$$N_{\mathrm{ops}}(\mathbf{C}_{\boldsymbol{n}}^{-1}) = M \sim 2N.$$

However note that these $\sim 2N$ floating point operations per iteration can be saved for stationary noise (i.e., $\mathbf{C}_{\boldsymbol{n}}^{-1} \propto \mathbf{I}$).

For the Fried model of wavefront sensor and after proper factorization,

$$N_{\mathrm{ops}}(\mathbf{S}) = N_{\mathrm{ops}}(\mathbf{S}^{\mathrm{T}}) \sim 4N.$$

This assumes, in particular, that the data were premultiplied by 2 [see Eq. (11)].

Finally, whatever is the unknown ($\boldsymbol{w}$ or $\boldsymbol{u}$), the total number of floating point operations required to apply the left hand side matrix $\mathbf{A}$ to a given vector is

$$N_{\mathrm{ops}}(\mathbf{A}) \sim 2N_{\mathrm{ops}}(\mathbf{K}) + 2N_{\mathrm{ops}}(\mathbf{S}) + N_{\mathrm{ops}}(\mathbf{C}_{\boldsymbol{n}}^{-1}) + N \sim 23N.$$

The last N comes from the addition of likelihood and regularization terms.

From Eqs. (B1) and (B2), using either $\boldsymbol{w}$'s or $\boldsymbol{u}$'s as the unknowns, initialization of the CG, i.e., computation of the initial residuals $\boldsymbol{r}_0$, involves

$$N_{\mathrm{ops}}(\boldsymbol{r}_0) \sim 2N_{\mathrm{ops}}(\mathbf{K}) + 2N_{\mathrm{ops}}(\mathbf{S}) + N_{\mathrm{ops}}(\mathbf{C}_{\boldsymbol{n}}^{-1}) + M + N \sim 25N$$

operations. Note that, if the algorithm is initialized with $\boldsymbol{x}_0 = \mathbf{0}$ (a vector of zeroes), this number of operations is significantly reduced down to $\sim 6N$ and $\sim 12N$ when, respectively, $\boldsymbol{w}$ and $\boldsymbol{u}$ are used as unknowns. Also note that there may be additional $\sim 6N$ operations to compute $\boldsymbol{w}$ from $\boldsymbol{u}$ when necessary.

Whatever are the considered variables, the number of unknowns is $\sim N$; hence any dot product in the CG algorithm involves $2N - 1 \sim 2N$ floating point operations. The first CG iteration (Fig. 1) requires two dot products ($2N - 1 \sim 2N$ floating point operations each) to compute ρ_k and α_k, applying $\mathbf{A}$ once and two vector updates (involving $\sim 2N$ operations each), hence a total of $\sim 31N$ operations. Any subsequent iteration requires an additional vector update to compute the CG direction, hence $\sim 33N$ operations. Finally, preconditioning by a diagonal preconditioner simply adds $\sim N$ operations per iteration.

The numbers of floating operations required by the different versions of the reconstruction algorithm are summarized in Table 1 and by Eq. (40). Note that, in the general case, the number of operations does not depend on which variables, $\boldsymbol{w}$ or $\boldsymbol{u}$, are used. There is a difference of $\sim 6N$ operations in the initialization step only when the algorithm is started with a zero initial vector (see Table 1).

ACKNOWLEDGMENTS

This project forms part of the "ELT Design Study" and is supported by the European Commission, within its Framework Programme 6, under contract No. 011863. This work was also supported by contract No. 0712729 with The European Southern Observatory (ESO). The authors would like to thank Clémentine Béchet and Nicholas Devaney for their fruitful comments. The algorithms and the simulations presented in this article have been implemented in Yorick, a free data processing language [37].

REFERENCES

1. F. Roddier, *Adaptive Optics in Astronomy* (Cambridge U. Press, 1999).
2. E. Gendron and P. Léna, "Astronomical adaptive optics. I. Modal control optimization," Astron. Astrophys. **291**, 337–347 (1994).
3. M. Le Louarn, N. Hubin, M. Sarazin, and A. Tokovinin, "New challenges for adaptive optics: extremely large telescopes," Mon. Not. R. Astron. Soc. **317**, 535–544 (2000).
4. N. Hubin, B. L. Ellerbroek, R. Arsenault, R. M. Clare, R. Dekany, L. Gilles, M. Kasper, G. Herriot, M. Le Louarn, E. Marchetti, S. Oberti, J. Stoesz, J.-P. Véran, and C. Vérinaud, "Adaptive optics for extremely large telescopes," in *Scientific Requirements for Extremely Large Telescopes*, Vol. 232 of IAU Symposium, P. A. Whitelock, M. Dennefeld, and B. Leibundgut, eds. (Cambridge U. Press, 2005), pp. 60–85.
5. L. A. Poyneer, D. T. Gavel, and J. M. Brase, "Fast wavefront reconstruction in large adaptive optics systems with use of the Fourier transform," J. Opt. Soc. Am. A **19**, 2100–2111 (2002).
6. L. A. Poyneer, D. Dillon, S. Thomas, and B. A. Macintosh, "Laboratory demonstration of accurate and efficient nanometer-level wavefront control for extreme adaptive optics," Appl. Opt. **47**, 1317–1326 (2008).
7. D. G. MacMartin, "Local, hierarchic, and iterative reconstructors for adaptive optics," J. Opt. Soc. Am. A **20**, 1084–1093 (2003).
8. B. Le Roux, J.-M. Conan, C. Kulcsár, H.-F. Raynaud, L. M. Mugnier, and T. Fusco, "Optimal control law for classical and multiconjugate adaptive optics," J. Opt. Soc. Am. A **21**, 1261–1276 (2004).
9. B. L. Ellerbroek, "Efficient computation of minimum-

variance wave-front reconstructors with sparse matrix techniques," J. Opt. Soc. Am. A **19**, 1803–1816 (2002).
10. C. R. Vogel, "Sparse matrix methods for wavefront reconstruction revisited," Proc. SPIE **5490**, 1327–1335 (2004).
11. W. H. Southwell, "Wave-front estimation from wave-front slope measurements," J. Opt. Soc. Am. **70**, 998–1006 (1980).
12. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. (Springer Verlag, 2006).
13. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. (Cambridge U. Press, 1992).
14. W. J. Wild, E. J. Kibblewhite, and R. Vuilleumier, "Sparse matrix wave-front estimators for adaptive-optics systems for large ground-based telescopes," Opt. Lett. **20**, 955–957 (1995).
15. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* (SIAM, 1994).
16. 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).
17. L. Gilles, "Order-*N* sparse minimum-variance open-loop reconstructor for extreme adaptive optics," Opt. Lett. **28**, 1927–1929 (2003).
18. L. Gilles, B. L. Ellerbroek, and C. R. Vogel, "Preconditioned conjugate gradient wave-front reconstructors for multiconjugate adaptive optics," Appl. Opt. **42**, 5233–5250 (2003).
19. Q. Yang, C. R. Vogel, and B. L. Ellerbroek, "Fourier domain preconditioned conjugate gradient algorithm for atmospheric tomography," Appl. Opt. **45**, 5281–5293 (2006).
20. C. R. Vogel and Q. Yang, "Fast optimal wavefront reconstruction for multi-conjugate adaptive optics using the Fourier domain preconditioned conjugate gradient algorithm," Opt. Express **14**, 7487–7498 (2006).
21. L. Gilles, B. Ellerbroek, and C. Vogel, "A comparison of multigrid *V*-cycle versus Fourier domain preconditioning for laser guide star atmospheric tomography," in *Signal Recovery and Synthesis*, OSA Topical Meetings, B. L. Ellerbroek and J. C. Christou, eds. (Optical Society of America, 2007), paper JTuA1.
22. L. Gilles and B. L. Ellerbroek, "Split atmospheric tomography using laser and natural guide stars," J. Opt. Soc. Am. A **25**, 2427–2435 (2008).
23. R. G. Lane, A. Glindemann, and J. C. Dainty, "Simulation of a Kolmogorov phase screen," Waves Random Media **2**, 209–224 (1992).
24. D. L. Fried, "Least-squares fitting a wave-front distortion estimate to an array of phase-difference measurements," J. Opt. Soc. Am. **67**, 370–375 (1977).
25. E. Thiébaut, "Introduction to image reconstruction and inverse problems," in NATO Science Series II., *Mathematics, Physics and Chemistry*, *Vol. 198,* R. Foy and F.-C. Foy, eds. (Springer, 2005), p. 397.
26. J. Herrmann, "Phase variance and Strehl ratio in adaptive optics," J. Opt. Soc. Am. A **9**, 2257–2258 (1992).
27. A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation* (SIAM, 2005).
28. A. Tarantola and B. Valette, "Inverse problems = quest for information for information," J. Geophys. **50**, 159–170 (1982).
29. D. L. Fried, "Statistics of a geometric representation of wavefront distortion," J. Opt. Soc. Am. **55**, 1427–1435 (1965).
30. G. Rousset, "Wavefront sensing," in *Adaptive Optics for Astronomy*, Vol. 423 of Proceedings of NATO ASI Series C, D. M. Alloin and J.-M. Mariotti, eds. (Kluwer, 1993), pp. 115–137.
31. N. Roddier, "Atmospheric wavefront simulation using zernike polynomials," Opt. Eng. (Bellingham) **29**, 1174–1180 (1990).
32. J. Skilling and R. K. Bryan, "Maximum entropy image reconstruction: general algorithm," Mon. Not. R. Astron. Soc. **211**, 111–124 (1984).
33. C. Béchet, M. Tallon, and E. Thiébaut, "Comparison of minimum-norm maximum likelihood and maximum a posteriori wavefront reconstructions for large adaptive optics systems," J. Opt. Soc. Am. A **26**, 497–508 (2009).
34. C. Béchet, "Commande optimale rapide pour l'optique adaptative des futurs télescopes hectométriques," Ph.D. dissertation (Ecole Centrale de Lyon, 2008).
35. C. Béchet, M. Tallon, and E. Thiébaut, "FRIM: minimum-variance reconstructor with a fractal iterative method," in *Advances in Adaptive Optics II*, Vol. 6272 of SPIE Conference, D. B. C. B. L. Ellerbroek, ed. (2006), p. 62722U.
36. C. Béchet, M. Tallon, and E. Thiébaut, "Closed-loop AO performance with FrIM," in *Adaptive Optics: Analysis and Methods*, Conference of the Optical Society of America (Optical Society of America, 2007), paper JTuA4.
37. D. Munro, Yorick, http://yorick.sourceforge.net/http://yorick.sourceforge.net/.