European Community's Framework Programme 6

EUROPEAN EXTREMELY LARGE TELESCOPE DESIGN STUDY

WP9600 - Algorithms for Reconstruction and Control

Document title:	The new *Fractal Iterative Method* for Fast Wavefront Reconstruction and Control
Document number:	ELT-INS-TRE-09600-0008
Issue No	1
Date	20 March 2007

Prepared by Michel Tallon, Eric Thiébaut, and Clémentine Béchet

Approved by Michel Tallon

Released by

CHANGE RECORD

Issue	Date	Section / Paragraph affected	Reason / Initiation / Remarks

Contents

Abbreviations

AO	Adaptive Optics
CG	Conjugate Gradient
ELT	Extremely Large Telescope
MAP	Maximum a priori
PCG	Preconditioned conjugate gradient
SVD	Singular value decomposition
VMM	Vector-matrix multiply

1 Scope

We describe a novel algorithm, FrIM (for Fractal reconstruction Iterative Method), aiming at the reconstruction of the optical wavefront from measurements provided by a wavefront sensor. Since our objective is Extremely Large Telescopes and regularized reconstruction, our algorithm was designed with speed and quality in mind. We show that the maximum *a posteriori* solution of the phase restoration problem yields the maximum Strehl ratio. A critical point of iterative methods is the evaluation of the priors (for regularization), which makes use of a non-sparse covariance matrix. We introduce a new multiscale algorithm, and show that a proper regularization following Kolmogorov statistics can be obtained in $O(N)$ operations, N being the number of phase samples to estimate. We then show that an iterative algorithm such as the conjugate gradient does not require the computation of any inverse matrix to solve the regularized problem. This allows one to benefit from the sparsity of the model matrix. Finally we propose an effective preconditioning which scales as $O(N)$ and yields the solution in few (at most about 10) conjugate gradient iterations for any N. The resulting algorithm being therefore $O(N)$.

2 Introduction

The aim of this report is to describe a new iterative method for reconstruction and control, named *Fractal Iterative Method* (FrIM).

The classical and most used method for AO control is based on vector-matrix multiply (VMM), where the so-called control matrix in precomputed, generally using modal control optimization (Gendron & Léna, 1994). But Extremely Large Telescopes yield to a very high number of degrees of freedom, owing to the larger diameter (Le Louarn et al., 2000), as well as to the needs of the new architectures of AO systems (Extreme AO, Multiconjugate AO, Multi-objects AO, etc.). More efficient algorithms are thus required. Furthermore, although least-square algorithms give suitable results for Single Star AO systems, minimum variance reconstruction is required to minimize the effect of the missing data in the other AO schemes.

As a new minimum variance method, we are here considering an iterative algorithm. Why ?

- We do not need to invert any matrix. As we will see, only the direct model is necessary. A 100×100 Single Star AO system yields an interaction matrix of $\sim 8000\times16000$ elements, *i.e.* $\sim$ 1 Gbytes to store, since its inverse is a full matrix.

- By saving the huge amount of computation for inverting, we can consider to adapt in real time to the evolving conditions of the AO system. This involves turbulence conditions of course, but also variations of measurements noise (*e.g.* scintillation), correlations in this noise (elongations of laser guide stars), of change in the geometry of the AO system (*e.g.* Multiconjugate or laser tomography AO).

- We can expect to take advantage of the sparsity of the direct model (interaction matrix) which is fast to apply. On the large matrices we are considering, this can save sufficient computations to allow many iterations of an iterative algorithm.

- The sparse direct model should be much easier to store in memory so we can consider to store it in cache memory and speed up the computation.

- While the method is considered here for reconstruction, we can notice that when running the AO system, the iterative algorithm always starts close to the solution. Indeed, the wavefront is not allowed to change a lot from one step of the AO loop to the other. This should reduce the number of iterations to reach a given level of quality.
- The first iterations of an iterative algorithm already deliver approximates of the solution. These approximates can be used immediately in the AO control loop. This could allow a reduction of the effect of the delay in the loop.

But two critical points must be solved in order to take full advantage of these nice properties of iterative methods: estimation of the covariance matrix of the wavefronts and preconditioning.

The covariance matrix is not sparse, neither is its inverse, so introducing priors would yield dramatic loss in efficiency as we will see. Nevertheless, priors are mandatory for recovering from missing measurements as it is the case for turbulence tomography because of central obscuration or of the so-called "unseen" spatial modes. Priors are also very useful since they allow the algorithm not to fit the noise. Using FrIM, we propose here to estimate the covariance matrix with a multi-scale algorithm in $O(N)$ operations, where N is the number of unknowns (phase samples or actuator commands).

The other major problem with iterative methods is the increase of the number of iterations with the number of unknowns to estimate (Southwell, 1980). This effect can be reduced by the use of preconditioning. Various schemes have been proposed, *e.g.* multigrid preconditioning (Barchers, 2004) or Fourier preconditioning (Vogel & Yang, 2006). By using our multi-scale operator, we can work directly in the Karhunen-Loève space and we can obtain fast convergence. This operator is then an efficient preconditioner.

Finally, the use of a classical Jacobi preconditioner, or of a new "optimal diagonal preconditioner" allows us to further improve the convergence: the number of iterations is finally ~ 10 whatever the size of the system, with a number of floating point operations $\sim 33 \times N$ per iteration. So the method is $O(N)$.

The report is organized as follows. We first derive the analytical expression of the minimum variance restored wavefront and the equations to be solved. We then introduce the fractal operator allowing fast computation of the regularization term in an iterative method such as the conjugate gradients which must be used to solve the problem in practice (i.e. for large numbers of degrees of freedom N). We propose different, cheap but efficient, preconditioners to yet speed up the iterative algorithm. Results are finally analyzed through different curves.

3 Minimum variance solution

3.1 Wavefront sensor

We assume that the wavefront sensor provides measurements of phase spatial derivatives (slopes or curvatures) linearily 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.

Although this equation is general as long as the wavefront sensor is linear, as a typical case, we will assume a Shack-Hartmann wavefront sensor with the Fried's geometry (Fried, 1977) for the simulations and for the evaluation of the efficiency of the algorithm.

3.2 Optimal wavefront reconstructor

The estimation of the wavefront $\boldsymbol{w}$ given the data $\boldsymbol{d}$ is an inverse problem which must be solved by using proper regularization in order to improve the quality of the solution whereas avoiding noise amplification (Thiébaut, 2005).

In order to keep the problem as simple as possible, we first introduce the requirement that the solution is a linear function of the data, in other words, the restored wavefront writes:

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

where $\mathbf{R}$ is the restoration matrix and $\boldsymbol{d}$ the wavefront sensor measurements. Some quality criterion is needed to derive the expression of 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 $\langle \cdot \rangle$ denotes the expected value of its argument. Interestingly, minimizing (on average) the variance of the residual wavefront yields the optimal Strehl ratio (Herrmann, 1992) 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 the wavefront $w(\boldsymbol{r})$ is in radian units (*i.e.* phase).

All these requirements lead to the following optimization problem:

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

where $\mathbf{R}^\dagger$ is the *best* reconstruction matrix according to our criterion. The *distance* between the true wavefront and its estimation writes:

$$\begin{aligned} \langle \|\tilde{\boldsymbol{w}} - \boldsymbol{w}\|^2 \rangle &= \langle \|\mathbf{R} \cdot \boldsymbol{d} - \boldsymbol{w}\|^2 \rangle \\ &= \langle \|\mathbf{R} \cdot (\mathbf{S} \cdot \boldsymbol{w} + \boldsymbol{n}) - \boldsymbol{w}\|^2 \rangle \, . \end{aligned}$$

This expression is quadratic with respect to the coefficients of $\mathbf{R}$ and its minimum is obtained by solving:

$$\left. \frac{\partial \langle \|\tilde{\boldsymbol{w}} - \boldsymbol{w}\|^2 \rangle}{\partial \mathbf{R}} \right|_{\mathbf{R}=\mathbf{R}^\dagger} = 0 \, , \tag{5}$$

where partial derivative of a scalar quantity $\varepsilon = \langle \|\tilde{\boldsymbol{w}} - \boldsymbol{w}\|^2 \rangle$ with respect to a matrix $\mathbf{R}$ is a matrix defined as:

$$\left[\frac{\partial \varepsilon}{\partial \mathbf{R}} \right]_{i,j} \stackrel{\text{def}}{=} \frac{\partial \varepsilon}{\partial R_{i,j}} \, .$$

By linearity of the expectation and of the derivation:

$$\begin{aligned}\frac{\partial\langle\|\tilde{\boldsymbol{w}}-\boldsymbol{w}\|^2\rangle}{\partial\mathbf{R}} &= \left\langle\frac{\partial\,\|\tilde{\boldsymbol{w}}-\boldsymbol{w}\|^2}{\partial\mathbf{R}}\right\rangle\\ &= \left\langle\frac{\partial\,\|\mathbf{R}\cdot(\mathbf{S}\cdot\boldsymbol{w}+\boldsymbol{n})-\boldsymbol{w}\|^2}{\partial\mathbf{R}}\right\rangle\\ &= 2\left\langle[\mathbf{R}\cdot(\mathbf{S}\cdot\boldsymbol{w}+\boldsymbol{n})-\boldsymbol{w}]\cdot(\mathbf{S}\cdot\boldsymbol{w}+\boldsymbol{n})^{\mathrm{T}}\right\rangle\\ &= 2\,(\mathbf{R}\cdot\mathbf{S}-\mathbf{I})\cdot\langle\boldsymbol{w}\cdot\boldsymbol{w}^{\mathrm{T}}\rangle\cdot\mathbf{S}^{\mathrm{T}}+2\,\mathbf{R}\cdot\langle\boldsymbol{n}\cdot\boldsymbol{n}^{\mathrm{T}}\rangle\\ &\quad+2\,(\mathbf{R}\cdot\mathbf{S}-\mathbf{I})\cdot\langle\boldsymbol{w}\cdot\boldsymbol{n}^{\mathrm{T}}\rangle\\ &\quad+2\,\mathbf{R}\cdot\langle\boldsymbol{n}\cdot\boldsymbol{w}^{\mathrm{T}}\rangle\cdot\mathbf{S}^{\mathrm{T}},\end{aligned}$$

where $\mathbf{I}$ is the identity matrix. Assuming that the wavefront $\boldsymbol{w}$ and the errors $\boldsymbol{n}$ are uncorrelated, and that the data are unbiased, *i.e.* $\langle\boldsymbol{n}\rangle = 0$, we have $\langle\boldsymbol{w}\cdot\boldsymbol{n}^{\mathrm{T}}\rangle = \langle\boldsymbol{w}\rangle\cdot\langle\boldsymbol{n}\rangle^{\mathrm{T}} = 0$ and similarly $\langle\boldsymbol{n}\cdot\boldsymbol{w}^{\mathrm{T}}\rangle = 0$. We can simplify the previous expression:

$$\frac{\partial\langle\|\tilde{\boldsymbol{w}}-\boldsymbol{w}\|^2\rangle}{\partial\mathbf{R}} = 2\,(\mathbf{R}\cdot\mathbf{S}-\mathbf{I})\cdot\mathbf{C}_{\boldsymbol{w}}\cdot\mathbf{S}^{\mathrm{T}}+2\,\mathbf{R}\cdot\mathbf{C}_{\boldsymbol{n}}$$

where, since $\langle\boldsymbol{n}\rangle = 0$, $\mathbf{C}_{\boldsymbol{n}} \overset{\text{def}}{=} \langle\boldsymbol{n}\cdot\boldsymbol{n}^{\mathrm{T}}\rangle$ is the covariance matrix of the errors and where, since a turbulent wavefront has zero mean, *i.e.* $\langle\boldsymbol{w}\rangle = 0$, $\mathbf{C}_{\boldsymbol{w}} \overset{\text{def}}{=} \langle\boldsymbol{w}\cdot\boldsymbol{w}^{\mathrm{T}}\rangle$ is the *a priori* covariance matrix of the wavefront. From that latter equation, the optimal wavefront reconstructor is:

$$\mathbf{R}^{\dagger} = \mathbf{C}_{\boldsymbol{w}}\cdot\mathbf{S}^{\mathrm{T}}\cdot\left(\mathbf{S}\cdot\mathbf{C}_{\boldsymbol{w}}\cdot\mathbf{S}^{\mathrm{T}}+\mathbf{C}_{\boldsymbol{n}}\right)^{-1}. \tag{6}$$

Applying this reconstructor to the data $\boldsymbol{d}$ however requires to solve a linear problem with as many equations as there are measurements. Generally, wavefront sensors provides more measurements than wavefront samples (about twice as many for a Shack-Hartmann or a curvature sensor). Fortunately, by use of Sherman-Morrison-Woodbury formula (Tarentola & Valette, 1982), Eq. (6) can be rewritten as:

$$\mathbf{R}^{\dagger} = \left(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}+\mathbf{C}_{\boldsymbol{w}}^{-1}\right)^{-1}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1} \tag{7}$$

which involves solving just as many linear equations as there are wavefront samples. The linear recontructor defined in Eq. (7) is the expression to be preferred in our case.

3.3 Links with other approaches

Using expression in Eq. (7) for the reconstructor, the minimum variance restored wavefront writes:

$$\begin{aligned}\boldsymbol{w}^{\dagger} &\overset{\text{def}}{=} \mathbf{R}^{\dagger}\cdot\boldsymbol{d}\\ &= \left(\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\mathbf{S}+\mathbf{C}_{\boldsymbol{w}}^{-1}\right)^{-1}\cdot\mathbf{S}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot\boldsymbol{d}\end{aligned}$$

which is also the solution of the quadratic problem:

$$\boldsymbol{w}^{\dagger} = \arg\min_{\boldsymbol{w}}\left\{(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{n}}^{-1}\cdot(\mathbf{S}\cdot\boldsymbol{w}-\boldsymbol{d})+\boldsymbol{w}^{\mathrm{T}}\cdot\mathbf{C}_{\boldsymbol{w}}^{-1}\cdot\boldsymbol{w}\right\}$$

initialisation:
  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
  $\boldsymbol{z}_k = \mathbf{Q} \cdot \boldsymbol{r}_k$ (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

Figure 1: Preconditioned conjugate gradient (PCG) algorithm for solving $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ where $\mathbf{A}$ is a symmetric positive definite matrix and $\mathbf{Q}$ is a preconditioner. The unpreconditioned version of the algorithm is simply obtained by taking $\mathbf{Q} = \mathbf{I}$, hence $z_k = \boldsymbol{r}_k$.

where $(\mathbf{S} \cdot \boldsymbol{w} - \boldsymbol{d})^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot (\mathbf{S} \cdot \boldsymbol{w} - \boldsymbol{d})$ is the so-called χ^2 which measures the discrepancy between the data and their model and $\boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{w}}^{-1} \cdot \boldsymbol{w}$ is a Tikhonov regularization term which enforces *a priori* covariance of the unknowns. Thus Eq. (7) is also the result of the maximum a posteriori solution. The usual hyper-parameter is here hidden in $\mathbf{C}_{\boldsymbol{w}}$ which is proportional to $(D/r_0)^{5/3}$.

As already noticed (Rousset, 1993), minimum variance estimator is somehow related to a Wiener-type filtering, but here with an optimal weight for the noise.

We can point out another link with the usual truncated SVD technique (Rousset, 1993). Actual adaptive optics systems make use of some expansion of the wavefront on a basis of modes, regularization being achieved by truncature of the ill-conditionned modes. Since this results in aliasing, we expect the MAP solution to be a better approximation of the wavefront. A somewhat smoother effect than crude mode truncature is obtained by writting:

$$\boldsymbol{w}_\mu = \arg\min_{\boldsymbol{w}} \left\{ (\mathbf{S} \cdot \boldsymbol{w} - \boldsymbol{d})^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot (\mathbf{S} \cdot \boldsymbol{w} - \boldsymbol{d}) + \mu \, \|\boldsymbol{w}\|^2 \right\}$$

where hyper-parameter μ allows to tune the regularization level required to avoid noise amplification (Thiébaut, 2005).

4 Iterative Method

Computing the optimal wavefront can be done in different ways. For instance, the reconstructor matrix $\mathbf{R}$ can be computed once, according to Eq. (6) or Eq. (7), and then applied to every data set $\boldsymbol{d}$. The direct computation of $\mathbf{R}$ scales as $O(N^3)$ operations, N being the number of wavefront samples. The reconstructor $\mathbf{R}$ is a $N \times M$ matrix and is not sparse in practice. Hence, the storage of $\mathbf{R}$ requires $M\,N \approx 2\,N^2$ 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 vector-matrix multiplication can be too long for real time applications. Moreover the memory requirement (*e.g.* for $N \simeq 10^4$, 1.5 Gb of memory are needed to store $\mathbf{R}$) may be such that memory page faults dominate the computation time of vector-matrix multiplication.

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

$$\left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} + \mathbf{C}_w^{-1}\right) \cdot \boldsymbol{w} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d} \tag{8}$$

which leads to the optimal wavefront $\boldsymbol{w}$ for every data set $\boldsymbol{d}$. As we will show in the following, the advantage of such an approach is that it allows us to fully exploit the particular structure of the linear equations to speed up the computations involved in one iteration. Iterative methods have however the drawback that the time spent depends on the number of required iterations which, in the worst cases, scales as the number of unknown N. In practice, numerical rounding errors can make the number of iterations still much higher, even on small systems (small N). We will nevertheless show that it is possible to bring the number of iterations down by proper preconditioning.

Equation (8) can be put in the more generic form:

$$\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b} \tag{9}$$

where, in the case of Eq. (8), $\boldsymbol{x} = \boldsymbol{w}$ and:

$$\mathbf{A} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} + \mathbf{C}_w^{-1} \tag{10}$$

is the so-called left hand side matrix, whereas

$$\boldsymbol{b} = \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d} \tag{11}$$

is the so-called right hand side vector. Barrett et al. (1994) have reviewed a number of iterative algorithms for solving such a linear system. The first 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. If $\mathbf{A}$ is sparse or if it has some special structure it is possible to compute the vector-matrix product in much less than $O(N^2)$ operations.

These iterative algorithms are therefore not only easy to implement but they are also easy to optimize by accounting for the particular structure of the matrix $\mathbf{A}$ of interest. This is specially relevant in our case since applying $\mathbf{A}$ (or its transpose) can be achieved by product of very sparse matrices providing multiplication by $\mathbf{C}_w^{-1}$ can be (approximated) by a small number of operations.

In our case, $\mathbf{A}$ given by Eq. (10) is, by construction, a symmetric positive definite matrix, the conjugate gradients (CG) (Barrett et al., 1994) is therefore the iterative method of choice to solve the system in Eq. (9). Figure 1 shows the steps of the conjugate gradients method to solve the system $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$. This method is known to have a quadratic convergence rate, and it is possible to speed up its convergence by using a proper preconditioner $\mathbf{Q}$. Ideally the preconditioner should be a good estimate of the inverse of the matrix $\mathbf{A}$, *i.e.* $\mathbf{Q} \approx \mathbf{A}^{-1}$. However if no preconditioner is available, taking $\mathbf{Q} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, yields the unpreconditioned version of the conjugate gradients algorithm. Note that, *e.g.* in Barrett

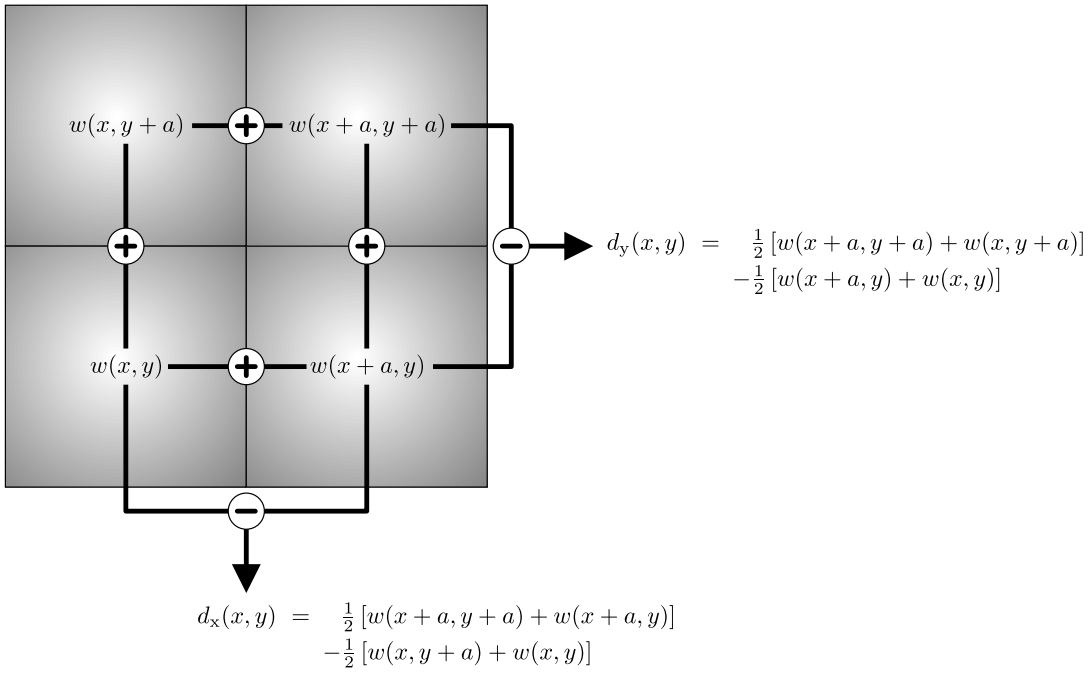

Figure 2: Wavefront sensor with Fried's geometry as used for our simulations. The gray squares stand for phase pixels $w(x, y)$, at the corners of the square subapertures of size a (not represented). This model is exact if we assume that the wavefront at any point in the pupil is obtained from a bilinear interpolation of phase samples at the corner of the subapertures.

et al. (1994) review, the preconditioner may also be given by the matrix $\mathbf{M} = \mathbf{Q}^{-1}$ such that $\mathbf{M} \approx \mathbf{A}$ but for which solving $\mathbf{M} \cdot \boldsymbol{z}_k = \boldsymbol{r}_k$ to obtain $\boldsymbol{z}_k$ is cheaper then solving $\mathbf{A} \cdot \boldsymbol{x} = \boldsymbol{b}$. The main advantage of this approach is that it is not necessary to know exactly the solution. This is well illustrated by the hierarchic iterative method (MacMartin, 2003).

The drawback of iterative methods is that the computational burden scales as the number of iterations required to approximate the solution with a sufficient precision. If a cheap (in the sense of the number of operations involved by the transform) approximation of the inverse of $\mathbf{A}$ (or a cheap to inverse linear transform which approximates $\mathbf{A}$) is known the number of iterations required by an iterative method can be greatly reduced by means of preconditioning. This is investigated latter in this report.

Let us examine now the two terms of Eq. (10), the so-called likelihood term, $\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \mathbf{S}$, which tells us how close we are from the data, and the regularization term $\mathbf{C}_{\boldsymbol{w}}^{-1}$ which tells us how close we are from our priors.

4.1 Computation of the likelihood term

Most adaptive optics systems use either a Shack-Hartmann sensor which prodives measurements of the local gradient of the wavefront or a curvature sensor which measures the local curvature of the wavefront. 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 $\mathbf{S}$. Indeed, 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 non-zero (M are the number of measurements and N are the number of wavefront samples). For instance, Fig. 2

shows the Fried's geometry of the Shack-Hartmann sensor model (Fried, 1977) which we used in our numerical simulations. The error free slopes are related to the wavefront by:

$$\begin{aligned} d_{\mathrm{x}}(x,y) &= \frac{1}{2}\left[w(x+a,y+a)+w(x+a,y)\right. \\ &\quad \left. -w(x,y+a)-w(x,y)\right] \\ d_{\mathrm{y}}(x,y) &= \frac{1}{2}\left[w(x+a,y+a)-w(x+a,y)\right. \\ &\quad \left. +w(x,y+a)-w(x,y)\right] \end{aligned} \tag{12}$$

where (x, y) are the pupil coordimates, 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 the number of degree of freedoms is. Besides, to a good approximation, wavefront sensors provide uncorrelated measurements (Roddier, 1999), hence the covariance matrix $\mathbf{C}_{\boldsymbol{n}}$ of the errors can be taken as a diagonal matrix:

$$\mathbf{C}_{\boldsymbol{n}} \approx \mathrm{diag}(\mathrm{Var}(\boldsymbol{n})) \tag{13}$$

where $\mathrm{Var}(\boldsymbol{n}) = \mathrm{tr}(\langle \boldsymbol{n} \cdot \boldsymbol{n}^{\mathrm{T}} \rangle)$ 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.

Computing the matrix multiplication of $\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{n}}^{-1} \cdot \mathbf{S}$ with any vector can be done in $2\,M\,N \sim 4N^2$ operations with VMM, the matrix product being precomputed.

With the benefit of the sparsity of Eq. (12), this number can be reduced, in different ways, depending on the exact writting of the equations. We can consider two extreme cases:

- In the most usual case, we have real Fried's geometry with uniform noise on the measurements. We can first rotate the measurements in M operations. This operation is done once for the right hand side (Eq. (11)) and it can be shown that the noise of the rotated measurements is still uncorrelated. The uniform weight $1/2$ in Eq. (12) can be included once in the uniform noise level, or in the right hand side. Applying $\mathbf{S}$ or $\mathbf{S}^{\mathrm{T}}$ is now done with one subtraction per measurement. Since the weighting (noise level and $1/2$ factor) can be applied afterwards in the unknowns space, the total number of floating point operations is $2\,M + N \approx 5\,N$.

- In the worst case, the measurements depends on the surrounding phase points with arbitrary weights, so no factorization is possible. It is no more Fried's geometry, from which we only keep that each measurement is linked to 4 phase points. In this case, we need 4 multiplications and 3 additions between the coefficients for each measurement: applying $\mathbf{S}$ then yieds $7\,M$ operations. If we assume that the diagonal of $\mathbf{C}_{\boldsymbol{n}}$ was included in the coefficients of $\mathbf{S}$, the number of floating point operations is $14\,M \approx 28\,N$.

4.1.1 Fast estimation of the regularization term

The sensor response matrix $\mathbf{S}$ is sparse for a Schack-Hartman and, for (partially) uncorrelated measurements, $\mathbf{C}_{\boldsymbol{n}}$ and $\mathbf{C}_{\boldsymbol{n}}^{-1}$ are sparse, the problem is that neither $\mathbf{C}_{\boldsymbol{w}}$ nor $\mathbf{C}_{\boldsymbol{w}}^{-1}$ are 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.

In an iterative method, the regularization term must be evaluated at every iteration and an efficient way to estimate $\boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{C}_{\boldsymbol{w}}^{-1} \cdot \boldsymbol{w}$ is highly desirable. Since $\mathbf{C}_{\boldsymbol{w}}$ is a real symmetric matrix,

by spectral factorization of $\mathbf{C}_w$, the covariance of the wavefront can be written:

$$\mathbf{C}_w = \mathbf{Z} \cdot \mathbf{\Lambda} \cdot \mathbf{Z}^{\mathrm{T}}$$

where $\mathbf{Z}$ is an orthogonal matrix, *i.e.* $\mathbf{Z} \cdot \mathbf{Z}^{\mathrm{T}} = \mathbf{Z}^{\mathrm{T}} \cdot \mathbf{Z} = \mathbf{I}$, of which the columns form an orthonormal basis of eigenvectors of $\mathbf{C}_w$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix of which the diagonal entries are the eigenvalues of $\mathbf{C}_w$. If none of the eigenvalues is zero, then $\mathbf{C}_w$ is invertible and

$$\mathbf{C}_w^{-1} = \mathbf{Z} \cdot \mathbf{\Lambda}^{-1} \cdot \mathbf{Z}^{\mathrm{T}},$$

where $\mathbf{\Lambda}^{-1} = \mathrm{diag}\,(1/\lambda_1, \ldots, 1/\lambda_N)$ therefore:

$$\begin{aligned} \boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{C}_w^{-1} \cdot \boldsymbol{w} &= \boldsymbol{w}^{\mathrm{T}} \cdot \mathbf{Z} \cdot \mathbf{\Lambda}^{-1} \cdot \mathbf{Z}^{\mathrm{T}} \cdot \boldsymbol{w} \\ &= \|\mathbf{\Lambda}^{-\frac{1}{2}} \cdot \mathbf{Z}^{\mathrm{T}} \cdot \boldsymbol{w}\|_2^2, \end{aligned}$$

where $\mathbf{\Lambda}^{-\frac{1}{2}} = \mathrm{diag}\left(1/\sqrt{\lambda_1}, \ldots, 1/\sqrt{\lambda_N}\right)$ — being an invertible covariance matrix, $\mathbf{C}_w$ is positive definite, all its eigenvalues are therefore strictly positive. Introducing the new variables:

$$\boldsymbol{u} \overset{\mathrm{def}}{=} \mathbf{K}^{-1} \cdot \boldsymbol{w} \quad \text{with} \quad \mathbf{K}^{-1} = \mathbf{\Lambda}^{-\frac{1}{2}} \cdot \mathbf{Z}^{\mathrm{T}}, \tag{14}$$

the regularization term simply writes:

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

Reciprocally, given the variables $\boldsymbol{u}$, the wavefront is obtained by:

$$\boldsymbol{w} = \mathbf{K} \cdot \boldsymbol{u} \quad \text{with} \quad \mathbf{K} \overset{\mathrm{def}}{=} \mathbf{Z} \cdot \mathbf{\Lambda}^{\frac{1}{2}}. \tag{15}$$

Note that, with this definition of $\mathbf{K}$, the covariance of the wavefront writes:

$$\mathbf{C}_w = \mathbf{K} \cdot \mathbf{K}^{\mathrm{T}}. \tag{16}$$

The expected value of $\boldsymbol{u}$ is $\langle \boldsymbol{u} \rangle = \mathbf{K}^{-1} \langle \boldsymbol{w} \rangle = \mathbf{0}$ and the covariance of $\boldsymbol{u}$ is the identity matrix:

$$\begin{aligned} \mathbf{C}_u &= \langle \boldsymbol{u} \cdot \boldsymbol{u}^{\mathrm{T}} \rangle \\ &= \mathbf{K}^{-\mathrm{T}} \cdot \langle \boldsymbol{w} \cdot \boldsymbol{w}^{\mathrm{T}} \rangle \cdot \mathbf{K}^{-1} \\ &= \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{\mathrm{T}} \cdot \mathbf{K} \cdot \mathbf{K}^{-1} \\ &= \mathbf{I}. \end{aligned}$$

This gives rise to a method for generating a wavefront since it is sufficient to generate a set $\boldsymbol{u}$ of N independent random values following a normal law: $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and to take $\boldsymbol{w} = \mathbf{K} \cdot \boldsymbol{u}$ to obtain a random wavefront with the proper covariance. In other words, the rows of $\mathbf{K}$ form a Karhunen-Loève basis of the random value $\boldsymbol{w}$. This method for generating a turbulent wavefront has been introduced by Roddier (1990) for the Zernike polynomials expansion of the wavefront. The drawback of using the Karhunen-Loève basis, is that it requires to inverse a potentially huge $N \times N$ matrix $\mathbf{C}_w$. However, observing that a turbulent wavefront has a fractal nature, Lane et al. (1992) have shown that $\mathbf{K}$ can be approximated in $O(N)$ operations by a mid-point algorithm. By using a similar algorithm we expect to be able to iteratively solve for the optimal wavefront in only $O(N)$ operations per iteration.

5 Fractal operators

5.1 Principle and structure function

The mid-point algorithm (Lane et al., 1992) starts at the largest scale of the wavefront (corners of the support) and step-by-step builds smaller scales by interpolating the wavefront values at half scales and by adding a random value at the mid-point with a standard deviation computed so that the new wavefront values and their neighbors conform to the proper structure function. This original mid-point algorithm cannot be used directly here, mainly because it is not invertible. We have modified this algorithm so that all the operations can be easily applied transpose, inverse, inverse-transpose.

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

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

It 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. For a turbulent wavefront obeying Kolmogorov's law:

$$f(r) = 6.88 \left(\frac{r}{r_0}\right)^{5/3} \tag{18}$$

where r_0 is the so-called Fried's parameter (Roddier, 1981). From the structure function, we can deduce the covariance of the wavefront between two positions in the pupil:

$$C_{i,j} = \langle w_i \, w_j \rangle \tag{19}$$

$$= \frac{1}{2}\left(\sigma_i^2 + \sigma_j^2 - f_{i,j}\right) \tag{20}$$

where $w_i = w(\boldsymbol{r}_i)$ is the wavefront phase at position $\boldsymbol{r}_i$ with variance $\sigma_i^2 = \mathrm{Var}(w_i)$ and where $f_{i,j} = f(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$ is the value of the structure function for a distance $|\boldsymbol{r}_i - \boldsymbol{r}_j|$. However, the wavefront variances (thus the covariance) are only defined for a von Kármán model of the turbulence. For pure Kolmogorov statistics, only the structure function can be defined. We will see that that σ_i^2's will appear as free parameters in this latter case. In any case, any structure function f can be used: in case the variance is undefined, we will propose a solution to choose a suitable value.

5.2 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. (1992) used 6 random values to generate the 4 initial corners. It is however possible to use only 4 random values as we will show in what follows. In fact, it is necessary 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, a property which is mandatory in our wavefront reconstruction algorithm.

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

$$\mathbf{C}_{\text{out}} = \begin{pmatrix} c_0 & c_1 & c_2 & c_1 \\ c_1 & c_0 & c_1 & c_2 \\ c_2 & c_1 & c_0 & c_1 \\ c_1 & c_2 & c_1 & c_0 \end{pmatrix} \quad \text{with} \quad \begin{cases} c_0 = \sigma^2 \\ c_1 = \sigma^2 - \frac{1}{2} f(D) \\ c_2 = \sigma^2 - \frac{1}{2} f(\sqrt{2}\, D) \end{cases} \tag{21}$$

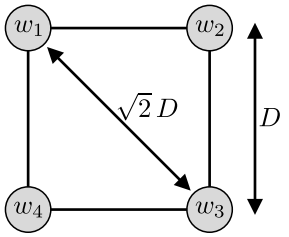

Figure 3: The initial values for wavefront generation are at the corner of the support of size D.

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 (*cf.* Fig. 3). Having the same variances σ^2 for the four outermost wavefront phase seems natural since none of these points play a particular role. For the four outer wavefront samples, the matrix of eigenvectors is:

$$\mathbf{Z}_{\text{out}} = \begin{pmatrix} 1/2 & -1/2 & 0 & 1/\sqrt{2} \\ 1/2 & 1/2 & -1/\sqrt{2} & 0 \\ 1/2 & -1/2 & 0 & -1/\sqrt{2} \\ 1/2 & 1/2 & 1/\sqrt{2} & 0 \end{pmatrix} \tag{22}$$

Note that the first eigenvector (first column) is a simple *piston*, and that the last two eigenvectors are the *tip-tilt* modes. The eigenvalues are:

$$\boldsymbol{\lambda}_{\text{out}} = \begin{pmatrix} c_0 + 2\,c_1 + c_2 \\ c_0 - 2\,c_1 + c_2 \\ c_0 - c_2 \\ c_0 - c_2 \end{pmatrix} = \begin{pmatrix} 4\,\sigma^2 - f(D) - \frac{1}{2}\,f(\sqrt{2}\,D) \\ f(D) - \frac{1}{2}\,f(\sqrt{2}\,D) \\ \frac{1}{2}\,f(\sqrt{2}\,D) \\ \frac{1}{2}\,f(\sqrt{2}\,D) \end{pmatrix}. \tag{23}$$

One way to choose σ^2 is to cancel the piston mode at this scale. This can be easily done from the eigenvalues expression in Eq. (23), by choosing:

$$\sigma^2 = \frac{1}{4}\,f(D) + \frac{1}{8}\,f(\sqrt{2}\,D)\,.$$

But in this case, the operator $\mathbf{K}$ is however no longer invertible. We have chosen σ^2 so that the smallest covariance, which is $c(\sqrt{2}\,D)$, between the most remote points, is exactly zero. Hence:

$$\sigma^2 = \frac{1}{2}\,f(\sqrt{2}\,D)\,. \tag{24}$$

Of course, when a von Kármán model of turbulence is chosen, both σ^2 and f are fixed by the model; Eq. (24) is to be used only for Kolmogorov case, where the value of σ^2 corresponds to the saturation level of the structure function. For a von Kármán model of the structure function, this value of the saturation level, corresponds to an outer scale $L_0 \approx 13D$.

The operators $\mathbf{K}_{\text{out}}$ and $\mathbf{K}_{\text{out}}^{-1}$ write:

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

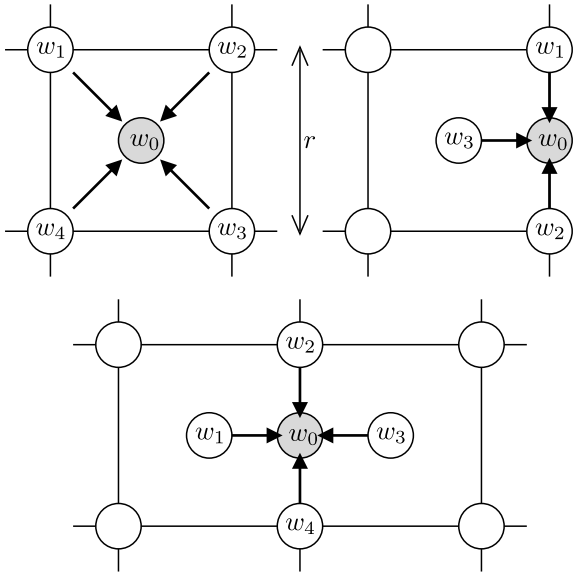

Figure 4: Wavefront refinement. To generate a grid with cell size $r/2$, new values (in gray) are generated from wavefront values (in white) of a grid with cell size equal to r. Top left: new value from 4 values $r/\sqrt{2}$ apart. Top right: new edge value from 3 values $r/2$ apart. Bottom: new value from 4 values $r/2$ apart.

$$\text{with} \quad \begin{cases} a &= \sqrt{4\,\sigma^2 - f(D) - \frac{1}{2}\,f(\sqrt{2}\,D)} \\ b &= \sqrt{f(D) - \frac{1}{2}\,f(\sqrt{2}\,D)} \\ c &= \sqrt{f(\sqrt{2}\,D)} \end{cases} \tag{25}$$

5.3 Generation of wavefront samples at smaller scales

Given the wavefront with a sampled step r, the mid-point algorithm generates a refined wavefront with a sampled step size of $r/2$ by a perturbated interpolation:

$$w_0 = \alpha_0\,u_0 + \sum_{j=1}^{N_{\text{interp}}} \alpha_j\,w_j \tag{26}$$

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_{interp} is the number of wavefront samples from the previous scale which are used to generate the new sample (*cf.* Fig. 4). This equation comes from a generalization of the principle of the original mid-point algorithm. Note that from Eq. (26), all these operations can be done *in-place*, since we proceed from the largest scale to smaller one.

The $N_{\text{interp}} + 1$ scalars α_j have to be adjusted so that the structure function between w_0 and

any of the $w_{i=1,\ldots,N_{\text{interp}}}$ matches the chosen one:

$$\begin{aligned} f_{i,0} &= \langle (w_0 - w_i)^2 \rangle \\ &= \left\langle \left(\alpha_0\, u_0 + \sum_{j=1}^{N_{\text{interp}}} \alpha_j\, w_j - w_i \right)^2 \right\rangle \\ &= \alpha_0^2 + \sum_{j=1}^{N_{\text{interp}}} \alpha_j\, f_{i,j} - \sum_{1 \le j < k \le N_{\text{interp}}} \alpha_j\, \alpha_k\, f_{j,k} \\ &\quad + \left(1 - \sum_{k=1}^{N_{\text{interp}}} \alpha_k \right) \left(\sigma_i^2 - \sum_{j=1}^{N_{\text{interp}}} \alpha_j\, \sigma_j^2 \right) \end{aligned} \tag{27}$$

where $f_{i,j} = f(|\boldsymbol{r}_i - \boldsymbol{r}_j|)$ and $C_{i,j} = \langle w_i\, w_j \rangle$ are the structure function and covariance between wavefront samples i and j, and where $\sigma_i^2 = C_{i,i}$ is the variance of sample i. Note that to obtain Eq. (27) we have accounted for the fact that $u_0 \sim \mathcal{N}(0,1)$ and hence $\langle u_0^2 \rangle = 1$ and, since u_0 and $w_{j=1,\ldots,N_{\text{interp}}}$ are uncorrelated, $\langle u_0\, w_{j=1,\ldots,N_{\text{interp}}} \rangle = 0$. This make N_{interp} equations, whereas there are $N_{\text{interp}} + 1$ unknown parameters $\{\alpha_0, \ldots, \alpha_{N_{\text{interp}}}\}$; an aditionnal constraint is therefore needed.

In original mid-point algorithm, Lane et al. (1992) choose to normalize the sum of the interpolation coefficients and use the constraint that $\sum_{j=1}^{N_{\text{interp}}} \alpha_j = 1$. In that case, Eq. (27) simplifies and the coefficients are obtained by solving:

$$\begin{aligned} f_{i,0} = \alpha_0^2 + \sum_{j=1}^{N_{\text{interp}}} \alpha_j\, f_{i,j} - \sum_{1 \le j < k \le N_{\text{interp}}} \alpha_j\, \alpha_k\, f_{j,k} \\ \text{s.t.} \quad \sum_{j=1}^{N_{\text{interp}}} \alpha_j = 1\,. \end{aligned} \tag{28}$$

Note that all the variances σ_j^2 are irrelevant with these constraints.

We consider here another constraint which is to have the same variance, says σ^2, for all the wavefront samples. In other words, we consider wavefront with stationary (shift-invariant) statistical properties. This is justified by the ultimate aim to reconstruct phase corrugations in several layers accross the atmosphere for the other schemes of AO systems.

With this choice, the additional equation is provided by $\langle w_0^2 \rangle = \sigma^2$ and the interpolation coefficients $\{\alpha_0, \ldots, \alpha_{N_{\text{interp}}}\}$ are obtained by solving the system of $N_{\text{interp}} + 1$ equations:

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

Accounting for the last equation, the system can be further simplified to:

$$\begin{aligned} &\sum_{j=1}^{N_{\text{interp}}} (2\,\sigma^2 - f_{i,j})\,\alpha_j = 2\,\sigma^2 - f_{i,0} \quad \text{for } i = 1, \ldots, N_{\text{interp}} \\ &\alpha_0^2 = \left[1 - \left(\sum_{j=1}^{N_{\text{interp}}} \alpha_j\right)^2\right] \sigma^2 + \sum_{1 \le j < k \le N_{\text{interp}}} \alpha_j\,\alpha_k\,f_{j,k}\,, \end{aligned} \tag{29}$$

where the first N_{interp} equations is a linear system which must be solved to obtain the $\alpha_{j=1,\ldots,N_{\text{interp}}}$ 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. (29) is equivalent to:

$$\begin{aligned} &\sum_{j=1}^{N_{\text{interp}}} C_{i,j}\,\alpha_j = C_{0,i} \quad \text{for } i = 1, \ldots, N_{\text{interp}} \\ &\alpha_0^2 = \sigma^2 - \sum_{j=1}^{N_{\text{interp}}} C_{0,j}\,\alpha_j\,. \end{aligned} \tag{30}$$

5.4 Derivation of the interpolation coefficients

The expressions of the interpolation coefficients for the different cases illustrated by Fig. 4 can now be derived. Denoting 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, $r/\sqrt{2}$ or $r/2$ (Fig. 4). Hence the only covariances required in our computations are:

$$\begin{array}{rcccl} c_0 & = & c(0) & = & \sigma^2 \\ c_1 & = & c(r/2) & = & \sigma^2 - \frac{1}{2}\,f(r/2) \\ c_2 & = & c(r/\sqrt{2}) & = & \sigma^2 - \frac{1}{2}\,f(r/\sqrt{2}) \\ c_3 & = & c(r) & = & \sigma^2 - \frac{1}{2}\,f(r) \\ c_4 & = & c(\sqrt{2}\,r) & = & \sigma^2 - \frac{1}{2}\,f(\sqrt{2}\,r) \end{array} \tag{31}$$

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

5.4.1 Square configuration

For the interpolation stage illustrated by the top-left part of Fig. 4 and according to Eq. (30), 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. (30) leads to:

$$\begin{aligned} \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 &= \frac{c_2}{c_0 + 2\,c_3 + c_4}\,, \\ \alpha_0 &= \pm\sqrt{c_0 - \frac{4\,c_2^2}{c_0 + 2\,c_3 + c_4}}\,. \end{aligned} \tag{32}$$

Note that the sign of α_0 is irrelevant.

5.4.2 Triangle configuration

In original mid-point algorithm (Lane et al., 1992), the values at the edges of the support (top-right part of Fig. 4) are generated from only the two neighbors on the edge, ignoring the third interior neighbor (denoted w_3 in the figure). According to Eq. (30), the interpolation coefficients $\{\alpha_1, \alpha_2, \alpha_3\}$ for this stage are obtained by solving:

$$\begin{pmatrix} c_0 & c_3 & c_2 \\ c_3 & c_0 & c_2 \\ c_2 & c_2 & c_0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_1 \\ c_1 \end{pmatrix} . \tag{33}$$

Solving this linear system and plugging the solution into Eq. (30) leads to:

$$\begin{aligned} \alpha_1 = \alpha_2 &= \frac{c_1\,(c_0 - c_2)}{c_0\,(c_0 + c_3) - 2\,c_2^2}\,, \\ \alpha_3 &= \frac{c_1\,(c_0 - 2\,c_2 + c_3)}{c_0\,(c_0 + c_3) - 2\,c_2^2}\,, \\ \alpha_0 &= \pm\sqrt{c_0 - \frac{c_1^2\,(3\,c_0 - 4\,c_2 + c_3)}{c_0\,(c_0 + c_3) - 2\,c_2^2}}\,. \end{aligned} \tag{34}$$

Note that the sign of α_0 is irrelevant.

5.4.3 Diamond configuration

The interpolation coefficients for the stage in the bottom part of Fig. 4 can be deduced from Eq. (32) by replacing r by $r/\sqrt{2}$, then:

$$\begin{aligned} \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 &= \frac{c_1}{c_0 + 2\,c_2 + c_3}\,, \\ \alpha_0 &= \pm\sqrt{c_0 - \frac{4\,c_1^2}{c_0 + 2\,c_2 + c_3}}\,. \end{aligned} \tag{35}$$

Note that the sign of α_0 is irrelevant.

5.5 The inverse operator

The inverse operator $\mathbf{K}^{-1}$ is easy to code and requires only $O(N)$ operations. For all the wavefront values w_0 given by Eq. (26), the value u_0 of the corresponding wavefront generator is trivially obtained by:

$$u_0 = \frac{1}{\alpha_0}\left(w_0 - \sum_{j=1}^{N_{\text{interp}}} \alpha_j\, w_j\right), \tag{36}$$

where $\{w_1, \ldots, w_{N_{\text{interp}}}\}$ are the neighbors of w_0 (*cf.* Fig. 4).

For the 4 outer points, the inversion is also trivial since $\mathbf{Z}_{\text{out}}$ is orthogonal, thus its inverse is: $\mathbf{Z}_{\text{out}}^{-1} = \mathbf{Z}_{\text{out}}^{\text{T}}$. The expressions are given by Eq. (25).

Note that, if the inversion is done starting at the smallest scales toward the largest ones, the operator $\mathbf{K}^{-1}$ can be performed *in-place*. This property may be important to avoids memory page faults and speed-up the computation.

5.6 The transpose operator

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

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

5.7 The inverse transpose operator

This operator combine the two previous ones, from the largest scale to the smallest one. The following algorithm computes $z = \mathbf{K}^{-\mathrm{T}} \cdot \boldsymbol{v}$ for any input vector $\boldsymbol{v}$:

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

Again, the operation can be done *in-place*, and the number of operations is $O(N)$.

6 Preconditioning for FrIM

Preconditioning is a convenient way to speed up the convergence of iterative optimization methods (Barrett et al., 1994). It is implemented with the preconditioned conjugate gradient (PCG) algorithm described in Fig. 1.

Preconditioning is generally introduced as finding a matrix $\mathbf{M}$ so that the transformed system

$$\mathbf{M}^{-1} \cdot \mathbf{A} \cdot \boldsymbol{w} = \mathbf{M}^{-1} \cdot \boldsymbol{b} \tag{37}$$

has the same solution as the original system $\mathbf{A}\boldsymbol{w} = \boldsymbol{b}$ but the spectral properties of its coefficient matrix $\mathbf{M}^{-1}\mathbf{A}$ may be more favorable. This is preconditioning on the left. Although it is almost

no change for the PCG algorithm (Fig. 1), we use here preconditioning on the right. The idea is to operate a change of variables such that the problem to solve with respect to the new variables is better conditioned.

Let $\boldsymbol{w} = \mathbf{Q} \cdot \boldsymbol{v}$ where $\mathbf{Q}$ is some non-singular matrix and $\boldsymbol{v}$ are the new set of variables. Since $\mathbf{Q}$ is non-singular, solving $(\mathbf{A} \cdot \mathbf{Q}) \cdot \boldsymbol{v} = \boldsymbol{b}$ for $\boldsymbol{v}$ and taking $\boldsymbol{w} = \mathbf{Q} \cdot \boldsymbol{v}$ yields the same solution as solving $\mathbf{A} \cdot \boldsymbol{w} = \boldsymbol{b}$ for $\boldsymbol{w}$ directly. The trick is to find a linear operator $\mathbf{Q}$ very inexpensive to apply whereas the condition number of $\mathbf{A} \cdot \mathbf{Q}$ is much lower than that of $\mathbf{A}$.

Using a preconditioner in an iterative method gives rise to some extra cost both initially for the setup and per iteration for applying it. For FrIM, we can allow some cost for constructing the preconditioner, but, considering the size of the system, we don't want to spend a lot of time applying it. This is why we have tested two *diagonal* preconditioners to speed-up FrIM, the classical Jacobi preconditioner (Barrett et al., 1994), and an original one so-called "optimal diagonal preconditioner", both presented in this section. Introducing preconditioning as a change of variables points to the use of operator $\mathbf{K}$ as a preconditioner, since our fractal operator as been defined as a change of variable (Eq. (15)). Last section will explain how this can be done.

6.1 Jacobi preconditioner

The preconditioner is effective if operator $\mathbf{Q}$ approximates the inverse of $\mathbf{A}$ or if $\mathbf{A} \cdot \mathbf{Q} \sim \mathbf{I}$. The Jacobi preconditioner achieves this aim but only for the diagonal of $\mathbf{A}$ (Barrett et al., 1994):

$$Q_{i,j} = \begin{cases} 1/A_{i,i} & \text{if} \quad i = j \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

It is the simplest preconditioner, both to construct and to apply.

6.2 Optimal diagonal preconditioner

Since we use a right-preconditioner, we know that it is applied to a vector of normally distributed random values $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The optimal diagonal preconditioner is computed so that its product with $\mathbf{A}$ is statistically as close as possible to the identity matrix. We want then to minimize $\left\langle \|(\mathbf{A} \cdot \mathbf{Q} - \mathbf{I}) \cdot \boldsymbol{u}\|^2 \right\rangle$:

$$\begin{aligned} \frac{\partial \left\langle \|(\mathbf{A} \cdot \mathbf{Q} - \mathbf{I}) \cdot \boldsymbol{u}\|^2 \right\rangle}{\partial \mathbf{Q}} &= \left\langle \frac{\partial \|(\mathbf{A} \cdot \mathbf{Q} - \mathbf{I}) \cdot \boldsymbol{u}\|^2}{\partial \mathbf{Q}} \right\rangle \\ &= 2\,\mathbf{A}^{\mathrm{T}} \cdot (\mathbf{A} \cdot \mathbf{Q} - \mathbf{I}) \cdot \left\langle \boldsymbol{u} \cdot \boldsymbol{u}^{\mathrm{T}} \right\rangle \\ &= 2\,\mathbf{A}^{\mathrm{T}} \cdot (\mathbf{A} \cdot \mathbf{Q} - \mathbf{I}) = 0. \end{aligned}$$

So the optimal *diagonal* preconditioner writes:

$$Q_{i,i} = \frac{A_{i,i}}{\sum_j A_{i,j}^2}, \qquad \text{and} \qquad Q_{i,j \neq i} = 0. \tag{39}$$

In contrast to Jacobi preconditioner, this preconditioner is costly to compute since every element of matrix $\mathbf{A}$ must be evaluated once to compute the denominator. But this is done only once, and it is fast to apply. With the preconditioner, we replace the system to solve by $\mathbf{A} \cdot \mathbf{Q} \cdot \boldsymbol{u} = \boldsymbol{b}$.

6.3 Fractal operator K as a preconditioner

If we consider preconditioning as a change of variable, $\boldsymbol{w} = \mathbf{Q} \cdot \boldsymbol{v}$, as explained before, we could also use the already seen change of variable $\boldsymbol{w} = \mathbf{K} \cdot \boldsymbol{u}$ through our fractal operator, $\mathbf{K}$ (Eq. (15)). Introducing this change in Eq. (8) and using Eq. (16), the system we have to solve becomes:

$$\left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} \cdot \mathbf{K} + \mathbf{I}\right) \cdot \boldsymbol{u} = \left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d}\right). \tag{40}$$

After $\boldsymbol{u}$ is found, we can recover $\boldsymbol{w}$ by using $\boldsymbol{w} = \mathbf{K} \cdot \boldsymbol{u}$. Owing to the properties of $\boldsymbol{u}$, the *a priori* covariance matrix now appear as the identity matrix. This is expected since the rows of $\mathbf{K}$ form a Karhunen-Loève basis of the random value $\boldsymbol{w}$.

7 Simulations and Results

7.1 Summary of the various methods

Let us summarize here the various possibilities we found to solve the system given by Eq. (8).

By using Eq. (16), we can first directly solve for $\boldsymbol{w}$:

$$\left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} + \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1}\right) \cdot \boldsymbol{w} = \left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d}\right) \tag{41}$$

This is now possible to do it quickly thanks to the fractal operators $\mathbf{K}^{-1}$ and $\mathbf{K}^{-\mathrm{T}}$ described in sections 5.5 and 5.7.

We can expect to speed up the convergence by using one of the 2 preconditioners $\mathbf{Q}$ presented in section 6. The system to solve is then:

$$\begin{aligned} \left(\left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} + \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1}\right) \cdot \mathbf{Q}\right) \cdot \boldsymbol{v} &= \left(\mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d}\right) \\ \text{and} \qquad \boldsymbol{w} &= \mathbf{Q} \cdot \boldsymbol{v}. \end{aligned} \tag{42}$$

The final step is easy to do since we have chosen diagonal preconditioners $\mathbf{Q}$.

We can also solve Eq. (9) in the space of random variables $\boldsymbol{u}$ as seen in section 6.3. In this case, the system to solve is:

$$\begin{aligned} \left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} \cdot \mathbf{K} + \mathbf{I}\right) \cdot \boldsymbol{u} &= \left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d}\right), \\ \text{and} \qquad \boldsymbol{w} &= \mathbf{K} \cdot \boldsymbol{u}. \end{aligned} \tag{43}$$

Finally, we can also apply one of the preconditioners $\mathbf{Q}$ in the previous system:

$$\begin{aligned} \left(\left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \mathbf{S} \cdot \mathbf{K} + \mathbf{I}\right)\mathbf{Q}\right) \cdot \boldsymbol{v} &= \left(\mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{C}_n^{-1} \cdot \boldsymbol{d}\right), \\ \text{and} \qquad \boldsymbol{w} &= \mathbf{K} \cdot \mathbf{Q} \cdot \boldsymbol{v}. \end{aligned} \tag{44}$$

We will compare all these methods.

7.2 Number of operations

In order to compare the wavefront reconstruction algorithms, we need to estimate the number of floating point operations for each of them. We detail here the steps of the conjugate gradient method and count the number of operations involved in the multiplication by the different linear

operators **S**, **K**, *etc*. 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 **K**'s dependence on r_0 can be factorized and included in operator **W** with no extra computational cost. This kind of optimization is not considered here.

Figure 1 summarizes the steps of the preconditioned conjugate gradient (PCG) algorithm (Barrett et al., 1994) to solve:

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

where $\boldsymbol{x}$ are the unknowns, the left hand side matrix **A** is positive definite and $\boldsymbol{b}$ is the right hand side vector.

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

$$\begin{aligned} \mathbf{A} &= \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot \mathbf{S} + \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1} \\ \boldsymbol{b} &= \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot \boldsymbol{d}\,, \end{aligned}$$

where $\mathbf{W} = \mathbf{C}_{\boldsymbol{n}}^{-1}$. Starting the algorithm with $\boldsymbol{w}_0$, the initial residuals write:

$$\begin{aligned} \boldsymbol{r}_0 &= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{w}_0 \\ &= \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot (\boldsymbol{d} - \mathbf{S} \cdot \boldsymbol{w}_0) - \mathbf{K}^{-\mathrm{T}} \cdot \mathbf{K}^{-1} \cdot \boldsymbol{w}_0\,. \end{aligned} \tag{45}$$

If the unknowns are the wavefront generators, *i.e.* $\boldsymbol{x} = \boldsymbol{u}$, then:

$$\begin{aligned} \mathbf{A} &= \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot \mathbf{S} \cdot \mathbf{K} + \mathbf{I} \\ \boldsymbol{b} &= \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot \boldsymbol{d}\,, \end{aligned}$$

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

$$\begin{aligned} \boldsymbol{r}_0 &= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{u}_0 \\ &= \mathbf{K}^{\mathrm{T}} \cdot \mathbf{S}^{\mathrm{T}} \cdot \mathbf{W} \cdot (\boldsymbol{d} - \mathbf{S} \cdot \mathbf{K} \cdot \boldsymbol{u}_0) - \boldsymbol{u}_0\,. \end{aligned} \tag{46}$$

7.2.1 Operator K

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

$$\begin{aligned} \mathcal{N}(\mathbf{K}) = \mathcal{N}(\mathbf{K}^{\mathrm{T}}) = \mathcal{N}(\mathbf{K}^{-1}) &= \mathcal{N}(\mathbf{K}^{-\mathrm{T}}) \\ &= 6\,N_{\boldsymbol{u}} - 14 \\ &\sim 6\,N\,, \end{aligned}$$

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, $\mathcal{N}(\mathbf{L})$ is the number of floating point operations required to apply a linear operator **L** to a vector.

7.2.2 Operator W

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

$$\mathcal{N}\,(\mathbf{W}) = M \sim 2\,N\,;$$

however note that these $\sim 2\,N$ floating point operations are generally saved since stationary noise is generally assumed.

7.2.3 Operator S

For our wavefront sensor model and after proper factorization:

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

This assumes, for instance, that the data were pre-multiplied by 2 (see section 4.1).

7.2.4 Operator A

Finally, whatever the unknown are ($\boldsymbol{w}$ 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:

$$\begin{aligned}\mathcal{N}(\mathbf{A}) &\sim 2\,\mathcal{N}(\mathbf{K}) + 2\,\mathcal{N}(\mathbf{S}) + \mathcal{N}(\mathbf{W}) + N \\ &\sim 23\,N\,,\end{aligned}$$

The last N taking into account the addition of likelihood and regularization terms. This is interesting to note that this number of operations do not depend of the chosen unknown ($\boldsymbol{w}$ or $\boldsymbol{u}$).

7.2.5 Number of operations in PCG

From equations (45) and (46), using either $\boldsymbol{w}$ or $\boldsymbol{u}$ as the unknowns, the computation of the initial residuals $\boldsymbol{r}_0$ while initializing the CG, involves

$$\begin{aligned}\mathcal{N}(\boldsymbol{r}_0) &\sim 2\,\mathcal{N}(\mathbf{K}) + 2\,\mathcal{N}(\mathbf{S}) + \mathcal{N}(\mathbf{W}) + M + N \\ &\sim 25\,N\end{aligned}$$

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 $\sim 6\,N$ and $\sim 12\,N$ when respectively $\boldsymbol{w}$ and $\boldsymbol{u}$ are used as unknowns.

Since in any cases, the number of unknowns is $\sim N$, any dot product in the CG algorithm involves $2\,N - 1 \sim 2\,N$ floating point operations.

The first CG iteration (Fig. 1) requires two dot products to compute ρ_k and α_k, applying $\mathbf{A}$ once and two vector updates (involving $\sim 2\,N$ operations each); hence a total of $\sim 31\,N$ operations. Any subsequent iteration requires an additional vector update to compute the conjugate gradient direction; hence $\sim 33\,N$ operations. Finally, preconditioning by a diagonal preconditioner simply adds $\sim N$ operations per iteration.

The number of floating operations required by the different versions of the reconstruction algorithm are summarized in table 1. Note that the number of operations depends on which variables $\boldsymbol{w}$ or $\boldsymbol{u}$ are used only when the algorithms start from a null guess.

If T is the number of (P)CG iterations required to achieve convergence, then:

$$\begin{aligned}\mathcal{N}_{\mathrm{CG}} &\sim (23 + 33\,T_{\mathrm{CG}})\,N \\ \mathcal{N}_{\mathrm{PCG}} &\sim (23 + 34\,T_{\mathrm{PCG}})\,N.\end{aligned} \tag{47}$$

We will adopt these formula for estimating the number of floating point operations in the following.

algorithm step	floating point operations
initialization	$\sim 25\,N$
1st CG iteration	$\sim 31\,N$
any subsequent CG iteration	$\sim 33\,N$
total after $T \geq 1$ iterations	$\sim (23 + 33\,T)\,N$
1st PCG iteration	$\sim 32\,N$
any subsequent PCG iteration	$\sim 34\,N$
total after $T \geq 1$ iterations	$\sim (23 + 34\,T)\,N$

Table 1: Number of operations involved in conjugate gradient (CG) and preconditioned conjugate gradient (PCG) applied to the wavefront restoration problem solved by our algorithm. The integers N and T 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 in this step is reduced down to $\sim 6\,N$ and $\sim 12\,N$ when respectively $\boldsymbol{w}$ and $\boldsymbol{u}$ are used as unknowns.

7.3 Configuration for the simulations

We have tested the different wavefront reconstruction algorithms on a number of different conditions. The only aim of the simulations is here to assess the speed of the reconstruction.

For every simulation, the wavefront sensor sampling is such that the size of the Shack-Hartmann subaperture is equal to Fried's parameter r_0. A wavefront is first generated, 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). Noise level is given by its standard deviation σ_{noise} in radians per r_0 (which is also here radians per subaperture). At each iteration of the algorithm, the residual wavefront is computed as the difference between the current solution and the unnoisy initial wavefront. The root mean squared error of the residual wavefront is computed over the pupil, piston removed. The piston mode is the only mode removed, and the so-called ill-conditioned *waffle* mode is *not* removed. Indeed, unlike least squares method, the priors allow this "hidden mode" to be estimated even if it is not measured. The diameter of the circular pupil is always the same as the size of the square support. A central obscuration is always introduced, with a diameter sets to $1/3$ of pupil diameter.

7.4 Comparison of rates of convergence

The following graphs sample two AO systems, 65×65 and 257×257 in size, and several level of noise from 1 rad$/r_0$ down to 0.05 rad$/r_0$ are examined. On each curve, the 6 algorithms are compared. All the curves are the median value obtained for 100 simulations under the same conditions. The different algorithms were applied to the same simulations.

The graphs have been plotted assuming a number of floating point operations given by Eq. (47), where here the number of unknowns is $N = 4225$ and $N = 66049$ for AO systems 65×65 and 257×257 respectively.

Various observations can be drawn from theses curves (Fig. 5 to Fig. 10):

- Solving by using $\boldsymbol{w}$ as unknowns is much slower than with $\boldsymbol{u}$, by more than one order of magnitude for a 65×65 system, and 2 orders of magnitude for 257×257.

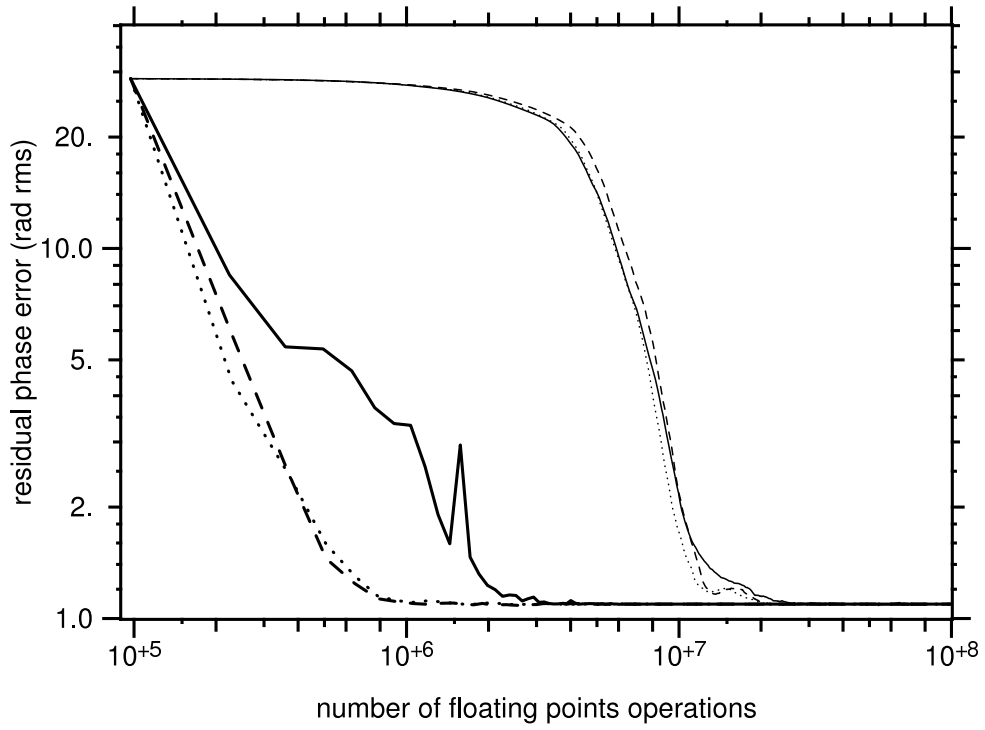

Figure 5: Phase error as a function of the number of operations. Curves are the median value of 100 simulations with $D/r_0 = 65$, $\sigma_{\text{noise}} = 1\,\text{rad}/r_0$ and one subaperture per r_0. Solid curves are for CG, dashed curves are for PCG with Jacobi preconditioner, dotted curves are for PCG with optimal diagonal preconditioner. Thin curves are for (P)CG onto the wavefont samples $\boldsymbol{w}$, whereas thick curves are for (P)CG onto the wavefront generator $\boldsymbol{u}$.

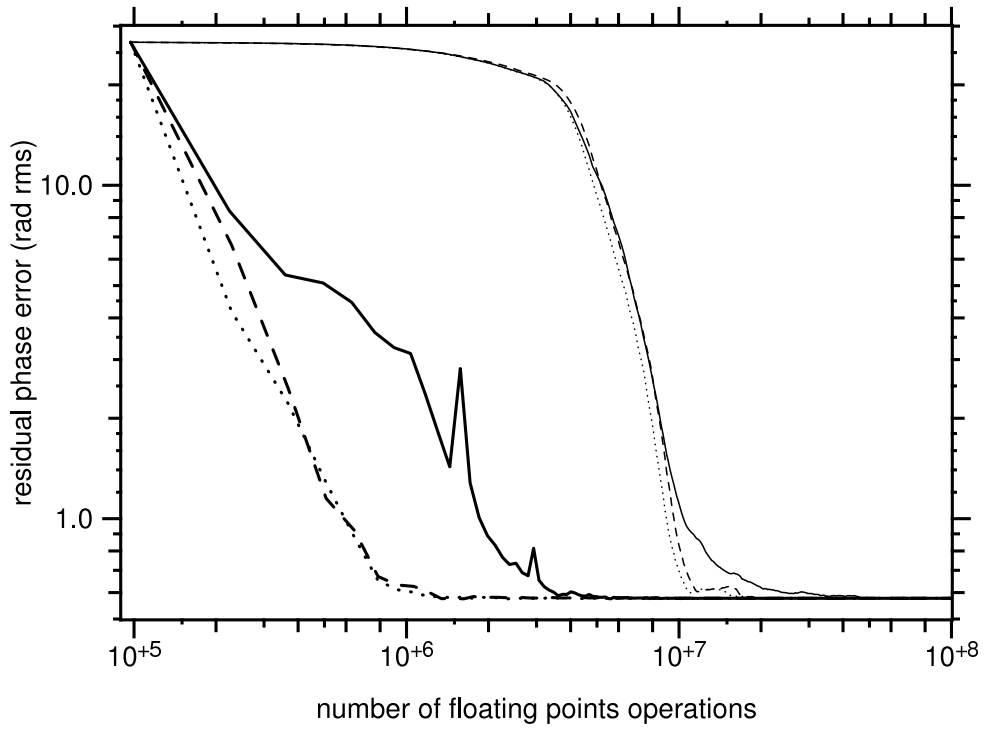

Figure 6: Same as Fig. 5 but for $\sigma_{\text{noise}} = 0.5\,\text{rad}/r_0$.

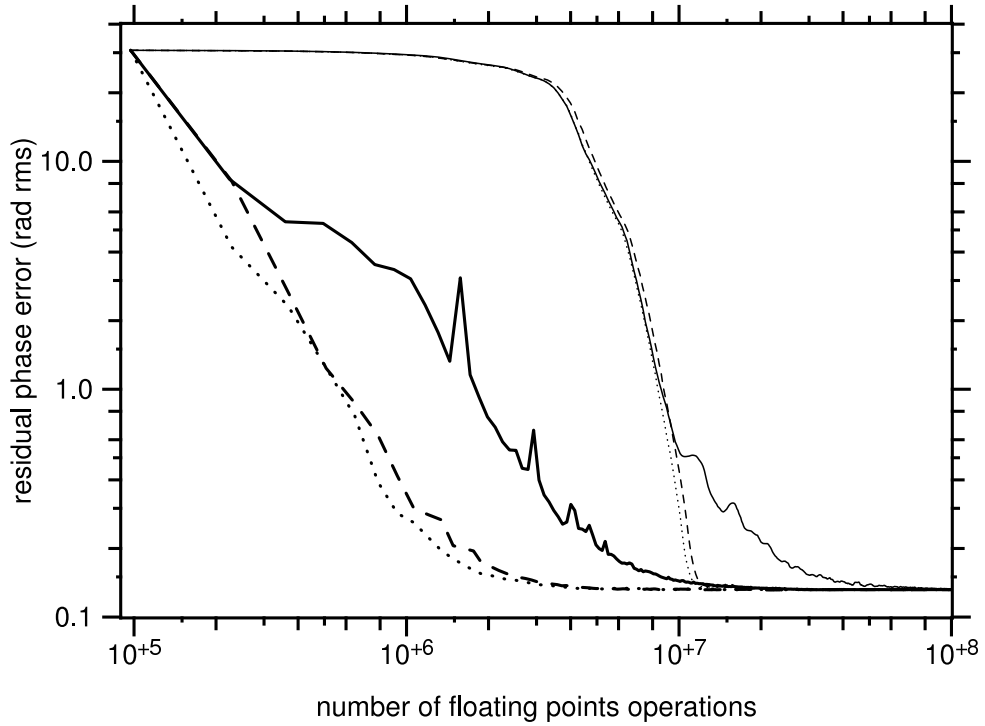

Figure 7: Same as Fig. 5 but for $\sigma_{\text{noise}} = 0.1\,\text{rad}/r_0$.

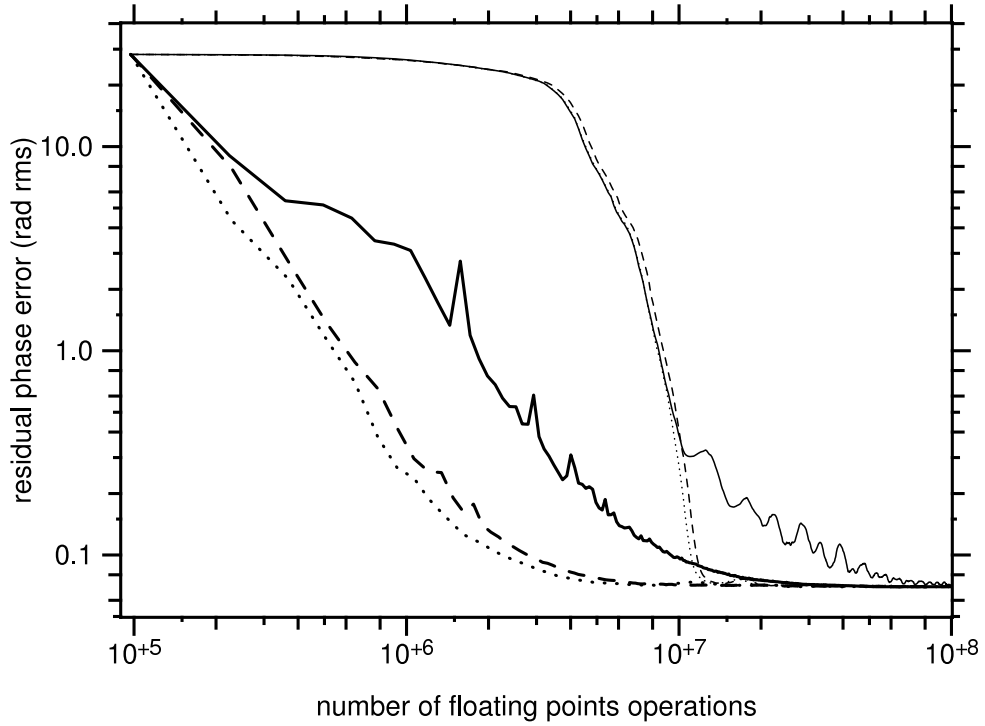

Figure 8: Same as Fig. 5 but for $\sigma_{\text{noise}} = 0.05\,\text{rad}/r_0$.

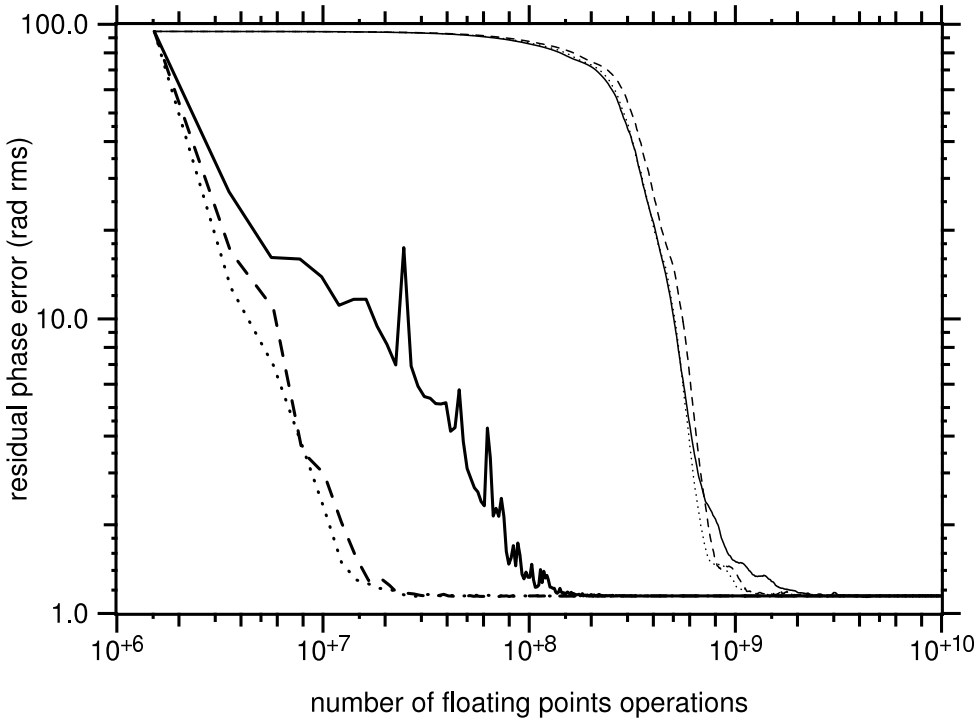

Figure 9: Same as Fig. 5 but for $D/r_0 = 257$ and $\sigma_{\text{noise}} = 1\,\text{rad}/r_0$.

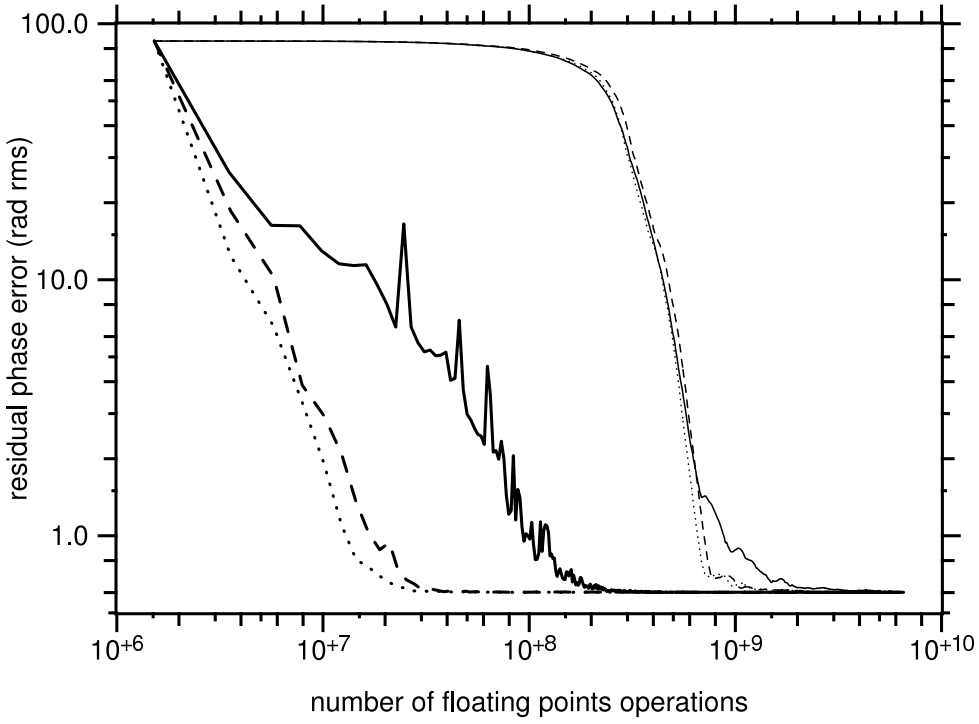

Figure 10: Same as Fig. 5 but for $D/r_0 = 257$ and $\sigma_{\text{noise}} = 0.5\,\text{rad}/r_0$.

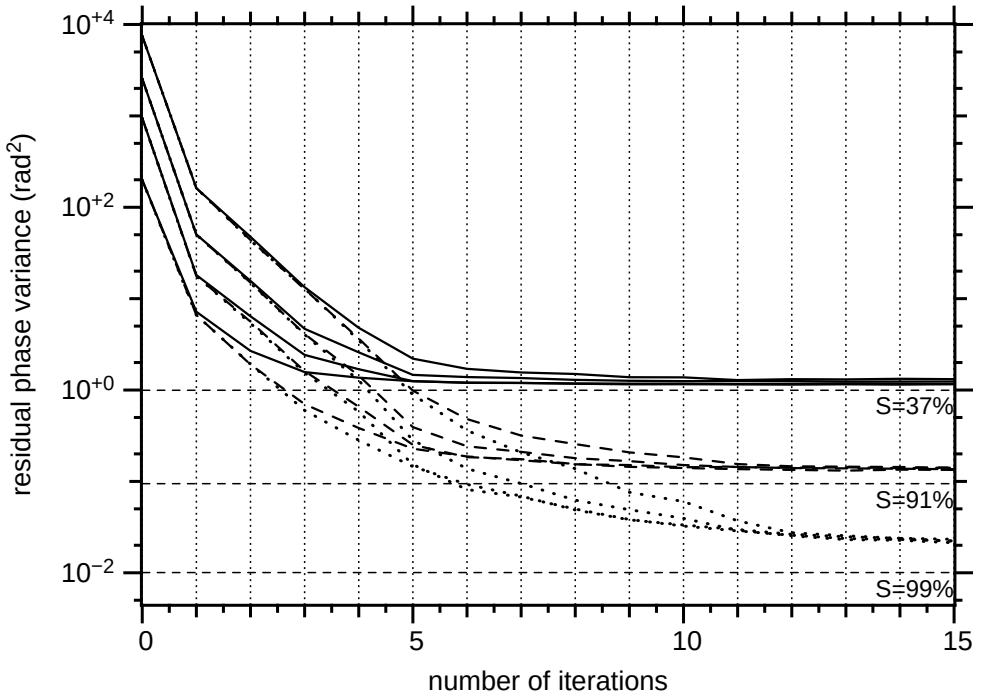

Figure 11: Decrease of the residual phase variance as a function of the number of iterations when using $\boldsymbol{u}$ as unknowns and optimal diagonal preconditioner. Each curve is the median value of 100 reconstructions. Three sets of curves are plotted for different values of σ^2_{noise}: 1 (solid), 0.09 (dashed), and 0.01 (dotted). 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 5 to 10 iterations are enough in most cases for a full reconstruction.

- With $\boldsymbol{w}$, the algorithm does not show any improvement of the residual error for a long time before to find its way through the solution. In contrast, the very first steps with $\boldsymbol{u}$ already show a tremendous reduction of the residual error.
- When solving for $\boldsymbol{w}$, diagonal preconditionners have almost no effect. They are only useful at the very end of the convergence, mainly in case of high signal to noise ratio. In contrast, the effect of the diagonal preconditioners is very effective from the beginning when working with $\boldsymbol{u}$. In this case, we can see that for the 257×257 system, the preconditioners make the convergence 10 times faster.
- When working with $\boldsymbol{u}$, the difference between the two diagonal preconditioners is significant but not critical. The optimal diagonal preconditioner gives the best results.
- When σ_{noise} decreases, the convergence of the two fastest methods takes longer to reach a lower residual errors, as expected, but the rate of convergence keeps steady.

7.5 Number of iterations

From the previous section, we now consider only the fastest method, using both the optimal diagonal preconditioner and $\boldsymbol{u}$ as unknowns. The aim here is to assess the number of iterations. As in the previous section, we consider one subaperture per r_0, so the variance of the incoming

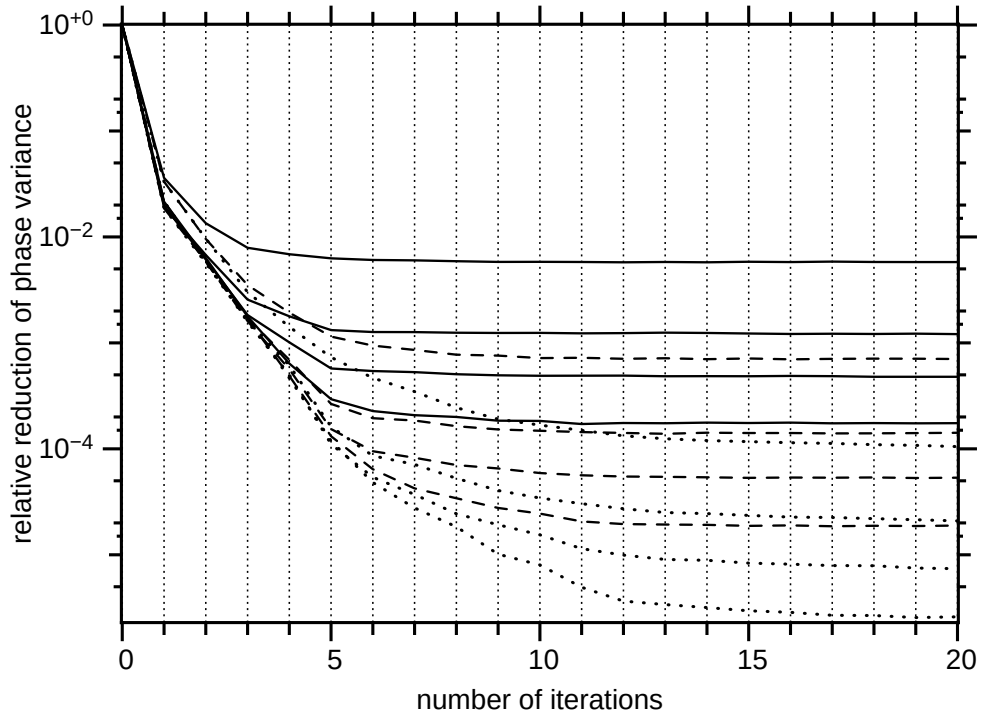

Figure 12: The same curves as those in Fig. 11 are plotted here, normalized by the initial variance of the phase. This shows that the high relative attenuation ($\sim 1/40$) after the first iteration, in any configuration. In each set of curves (σ^2_{noise}: 1 (solid), 0.09 (dashed), and 0.01 (dotted)), the size of the system increases from top to bottom: 32, 64, 128 and 256 subapertures along the diameter of the pupil.

wavefronts increases with the size of the system. Figure 11 shows how the residual phase variance decreases at each iteration for various configurations of the system, in size (33×33, 65×65, 129×129, 257×257), and in noise level (σ^2_{noise}= 1, 0.09 and 0.01). In the first iterations, we can see that the behavior of the algorithm does not depend on the signal to noise ratio. Further, the final value obtained does not depend on the size of the system. Strehl levels corresponding to the noise variance are indicated. The curves show that 5 to 10 iterations only are enough in most cases for a full reconstruction.

In order to remove the effect of starting from different initial phase variance, the curves have been plotted on Fig. 12, normalized by the initial variance. The curves show that the behavior of the algorithm is the same in any situation, the curves being only separated by the different values of the final variance, which depends on the noise level. They also show that the variance is already reduced by a factor $\sim 1/40$ at the first iteration only, $\sim 1/100$ at the second iteration and $\sim 10^{-4}$ after 6 iterations. This steep descent will be an asset in closed-loop.

8 Conclusion

We have introduced *FrIM*, a new minimum variance iterative algorigthm for fast wavefront reconstruction and fast control of an adaptive optics system. As other iterative methods, it takes advantage of the sparsity of the model matrix of wavefront sensors (or of interaction matrices), **S**, but it is able to preserve this advantage when computing the priors, by using a new "fractal

operator".

Based on a generalization of the mid-point algorithm (Lane et al., 1992), this fractal operator $\mathbf{K}$ is not sparse but can be implemented so that it requires only $O(N) \simeq 6\,N$ operations. The modifications introduced in the original algorithm allows the operator to be inverted and the generated wavefront to be stationary. Any structure function can be implemented. We have found algorithms for computing $\mathbf{K}^{-1}$, $\mathbf{K}^{\mathrm{T}}$ and $\mathbf{K}^{-\mathrm{T}}$ in the same number of floating point operations. The property of stationary is expected to be helpful for turbulence tomography.

Another breakthrough comes from the efficiency of this operator when used as a preconditioner. With the help of a so-called "optimal diagonal preconditioner", we have further reduced the number of iterations in the range of 5 – 10 for any size of the AO system. The number of iterations only depends on the signal to noise ratio, *i.e.* on the ratio $(D/r_0)^{5/3}/\sigma^2_{\mathrm{noise}}$.

It is beyond this report to compare with the other methods currently studied in response to the huge increase of the number of degrees of freedom for the AO system on ELTs. Nevertheless, we can easily compare to the classical vector-matrix multiplication. The simulations show that with *FrIM*, assuming uncorrelated data noise (*i.e.* $\mathbf{C}_{\boldsymbol{n}}$ is diagonal), the number of operations is $\mathcal{N}_{\mathrm{PCG}} \sim (23 + 34\,T_{\mathrm{PCG}})\,N$, where $T_{\mathrm{PCG}} \lesssim 10$ for any number of degrees of freedom N.

For up to $N_{\mathrm{DF}} = 13 \times 10^3$ degrees of freedom[1] (*i.e.* $D/r_0 \leq 128$), one wavefront estimation (from scratch) involves $\lesssim 6 \times 10^6$ operations, that is a bandwidth of $\sim 500\,\mathrm{Hz}$ for a machine capable of 3 Gflops (3×10^9 floating point operations per second) which is typically that of current workstations. Note that conventional (non-sparse) matrix multiplication would require $\sim 4\,N_{\mathrm{DF}}^2 \sim 7 \times 10^8$ operations to compute the wavefront: our method is $\gtrsim 100$ faster. Furthermore, since the operations can be done *in-place*, it is expected that the computation with FrIM could all be done in cache memory.

The next step of this work is to extend the theory to 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 will be lower.

After analyzing FrIM in closed-loop, the main points to address in the near future are the test of the method with a multi-conjugate configuration of the AO system, and the experimentation of FrIM implemented in a AO numerical simulator at ESO.

References

Barchers, J. D. 2004, "Multigrid approach to predictive wave-front reconstruction in adaptive optical systems", Appl. Opt., 43, 3708

Barrett, R., Berry, M., Chan, T. F., et al. 1994, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (Philadelphia, PA: SIAM)

Fried, D. L. 1977, "Least-squares fitting a wave-front distortion estimate to an array of phase-difference measurements", J. Opt. Soc. Am., 67, 370

Gendron, E. & Léna, P. 1994, "Astronomical adaptive optics. I. Modal control optimization", Astron. Astrophys., 291, 337

[1] In the fractal algorithm, the number of unknowns is $N_{\boldsymbol{u}} = (2^n + 1)^2$, which is suitable for $D/r_0 \leq 2^n$ and the number of degrees of freedom is $N_{\mathrm{DF}} \sim \frac{\pi}{4}\,(D/r_0)^2 \lesssim \frac{\pi}{4}\,N_{\boldsymbol{u}}$.

Herrmann, J. 1992, "Phase variance and Strehl ratio in adaptive optics", J. Opt. Soc. Am. A, 9, 2258

Lane, R. G., Glindemann, A., & Dainty, J. C. 1992, "Simulation of a Kolmogorov phase screen", Wave in random media, 2, 209

Le Louarn, M., Hubin, N., Sarazin, M., & Tokovinin, A. 2000, "New challenges for adaptive optics: extremely large telescopes", Mon. Not. R. Astr. Soc., 317, 535

MacMartin, D. G. 2003, "Local, hierarchic, and iterative reconstructors for adaptive optics", J. Opt. Soc. Am. A, 20, 1084

Roddier, F. 1981, The effects of atmospheric turbulence in optical astronomy, ed. E. Wolf, Vol. 19 (Amsterdam: North-Holland Publishing Company), 281–376

—. 1999, Adaptive Optics in Astronomy (Cambridge University Press)

Roddier, N. A. 1990, "Atmospheric wavefront simulation using Zernike polynomials", Opt. Eng., 29, 1174

Rousset, G. 1993, in Proc. NATO ASI Series C, Vol. 423, Adaptive optics for astronomy, ed. D. M. Alloin & J.-M. Mariotti (Kluwer, Dordrecht, The Netherlands), 115–137

Southwell, W. H. 1980, "Wave-front estimation from wave-front slope measurements", J. Opt. Soc. Am., 70, 998

Tarentola, A. & Valette, B. 1982, "Inverse Problems = Quest for Information", Journal of Geophysics, 50, 159

Thiébaut, E. 2005, in Optics in Astrophysics, ed. R. Foy & F.-C. Foy, NATO ASI (Kluwer Academic)

Vogel, C. R. & Yang, Q. 2006, "Fast optimal wavefront reconstruction for multi-conjugate adaptive optics using the Fourier domain preconditioned conjugate gradient algorithm", Opt. Express, 14, 7487

Acknowledgments

The algorithms and the simulations presented in this report have been implemented in Yorick, a free data processing language written by D. Munro (`ftp://ftp-icf.llnl.gov/pub/Yorick/`).