# Fast iterative optimal estimation of turbulence wavefronts with recursive block Toeplitz covariance matrix

R. Conan<sup>a</sup>

<sup>a</sup>RSAA, The Australian National University, Weston Creek, ACT 2611, Australia

# ABSTRACT

The estimation of a corrugated wavefront after propagation through the atmosphere is usually solved optimally with a Minimum-Mean-Square-Error algorithm. The derivation of the optimal wavefront can be a very computing intensive task especially for large Adaptive Optics (AO) systems that operates in real-time. For the largest AO systems, efficient optimal wavefront reconstructor have been proposed either using sparse matrix techniques or relying on the fractal properties of the atmospheric wavefront. We propose a new method that exploits the Toeplitz structure in the covariance matrix of the wavefront gradient. The algorithm is particularly well–suited to Shack–Hartmann wavefront sensor based AO systems. Thanks to the Toeplitz structure of the covariance, the matrices are compressed up to a thousand-fold and the matrix-to-vector product is reduced to a simple one–dimension convolution product. The optimal wavefront is estimated iteratively with the MINRES algorithm which exhibits better convergence properties for ill–conditioned matrices than the commonly used Conjugate Gradient algorithm. The paper describes, in a first part, the Toeplitz structure of the covariance matrices and shows how to compute the matrix-to-vector product using only the compressed version of the matrices. In a second part, we introduced the MINRES iterative solver and shows how it performs compared to the Conjugate Gradient algorithm for different AO systems.

Keywords: wavefront sensing, adaptive optics, toeplitz matrix, optimal wavefront estimation

# 1. INTRODUCTION

In an adaptive optics system, the estimation of the wavefront from the measurements of the wavefront sensor is the main real-time computing burden. Recently several efficient methods have been proposed to address this problem. The wavefront can be efficiently estimated in the Fourier domain<sup>1</sup> and more recently the CURE<sup>2</sup> algorithm has shown remarkable performance for real-time applications.

While both methods are able to accurately reconstruct the wavefront, they are not optimal. By optimal, we mean that these methods don't minimize the mean square difference between the original wavefront and its estimate, instead the euclidian norm between the wavefront measurements and the estimation measurements is minimized. Efficient optimal wavefront reconstructor have been already proposed. One is based on sparse<sup>3</sup> matrix techniques using an approximation to the inverse of the wavefront covariance matrix. Another is using the fractal properties of the atmospheric wavefront to reconstruct the wavefront iteratively by successive application of a fractal operator.<sup>4</sup>

The wavefront estimation method presented here is also optimal and it used the exact covariance matrices. Its effectiveness is due to the exploitation of the Toeplitz structure of the covariance matrices for a system defined in a cartesian geometry like the Shack–Hartmann wavefront sensor (SH–WFS) or the pyramid wavefront sensor.

Adaptive Optics Systems IV, edited by Enrico Marchetti, Laird M. Close, Jean-Pierre Véran, Proc. of SPIE Vol. 9148, 91480R · © 2014 SPIE CCC code: 0277-786X/14/\$18 · doi: 10.1117/12.2054472

Further author information: (Send correspondence to R. Conan)

R. Conan: E-mail: rod.conan@anu.edu.au

#### 2. OPTIMAL WAVEFRONT ESTIMATION

Assuming a vector parameter  $\theta$  to be estimated from the data x,  $\hat{\theta}$  is the estimator of  $\theta$  defined by

$$\hat{\boldsymbol{\theta}} = M\boldsymbol{x}$$
 (1)

where M is chosen such as it minimizes the Bayesian mean-square error<sup>5</sup> (MSE):

$$b_{MSE} = \left\langle (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^2 \right\rangle \tag{2}$$

where  $\langle x \rangle$  is the mean of the stochastic variable x.  $\hat{\theta}$  is the Linear–Minimum–Mean–Square–Error (LMMSE) estimator of  $\theta$  and M is given by

$$M = C_{\theta x} C_{xx}^{-1}. \tag{3}$$

Eq. (1) assumes a kind of linear relationship between  $\boldsymbol{x}$  and  $\boldsymbol{\theta}$ .

For an adaptive optics system, the measurement equation between the wavefront sensor data s and the wavefront  $\varphi$  is

$$\boldsymbol{s} = \boldsymbol{G}\boldsymbol{\varphi} + \boldsymbol{n} \tag{4}$$

where G is the WFS operator and  $\boldsymbol{n}$  is a vector of white noise. Since this is a linear equation, the wavefront estimate  $\hat{\boldsymbol{\varphi}}$  can be derived from Eq. (1) and Eq. (3) setting  $\boldsymbol{\theta} \equiv \boldsymbol{\varphi}$  and  $\boldsymbol{x} \equiv \boldsymbol{s}$  with

$$C_{\boldsymbol{x}\boldsymbol{x}} = G C_{\boldsymbol{\varphi}\boldsymbol{\varphi}} G^T + C_{\boldsymbol{n}\boldsymbol{n}} \tag{5}$$

and

$$C_{\boldsymbol{\theta}\boldsymbol{x}} = C_{\boldsymbol{\varphi}\boldsymbol{\varphi}} G^T. \tag{6}$$

 $C_{\varphi\varphi}$  is the wavefront covariance and  $C_{nn}$  is the noise covariance. With the former equations, the wavefront estimate is written

$$\hat{\boldsymbol{\varphi}} = C_{\boldsymbol{\varphi}\boldsymbol{\varphi}}G^T \left(GC_{\boldsymbol{\varphi}\boldsymbol{\varphi}}G^T + C_{\boldsymbol{n}\boldsymbol{n}}\right)^{-1}s \tag{7}$$

or

$$\hat{\boldsymbol{\varphi}} = \left( C_{\boldsymbol{\varphi}\boldsymbol{\varphi}}^{-1} + G^T C_{\boldsymbol{n}\boldsymbol{n}}^{-1} G \right)^{-1} G^T C_{\boldsymbol{n}\boldsymbol{n}}^{-1} s.$$
(8)

Eq. (8) is used in Ref. 3 and in Ref. 4 to derive  $\hat{\varphi}$ . Our approach is to derive  $\hat{\varphi}$  with Eq. (7). First lets note that

$$GC_{\varphi\varphi}G^T = C_{ss} \tag{9}$$

is the covariance of the measurements and that

$$C_{\varphi\varphi}G^T = C_{\varphi s} \equiv \Xi \tag{10}$$

is the cross-correlation between the measurements and the wavefront. Moreover  $C_{nn}$  is equal to  $I\sigma_n^2$  with  $\sigma_n^2$  the wavefront sensor noise variance and I the identity matrix. We also have diag  $(C_{ss}) = I\sigma_s^2$  with  $\sigma_s^2$  the variance of the measurements. The above considerations are enough to prove that the matrix

$$\Theta = C_{ss} + C_{nn} \tag{11}$$

will have the same structure than  $C_{ss}$ . This is important for the method we are presenting. Finally Eq. (7) is rewritten

$$\hat{\boldsymbol{\varphi}} = \Xi \boldsymbol{\zeta} \tag{12}$$

with  $\boldsymbol{\zeta}$  solution of

$$\Theta \boldsymbol{\zeta} = \boldsymbol{s}. \tag{13}$$

This is similar to the tomographic wavefront estimation method proposed for Multi–Object Adaptive Optics systems in Ref. 6. Eq. (13) will be solved with an iterative method requiring a single matrix–vector product (MVP) per iteration. Another MVP, Eq. (12), is needed to obtain  $\hat{\varphi}$ .

In the following, we show that the MVP can be computed very efficiently and we use a fast converging iterative method without pre–conditionning.

#### Proc. of SPIE Vol. 9148 91480R-2

#### **3. THE COVARIANCES**

In the following the WFS is assumed to be of the Shack–Hartmann (SH) type. The SH–WFS measurement s is made of two vectors, the measurement along the x–axis  $s_x$  and along the y–axis  $s_y$ , i.e.  $s = [s_x \ s_y]^T$ . With this definition for s, the matrices  $\Theta$  and  $\Xi$  are decomposed into x and y components:

$$\Theta = \left\langle ss^{T} \right\rangle = \begin{bmatrix} \left\langle s_{x}s_{x}^{T} \right\rangle & \left\langle s_{x}s_{y}^{T} \right\rangle \\ \left\langle s_{y}s_{x}^{T} \right\rangle & \left\langle s_{y}s_{y}^{T} \right\rangle \end{bmatrix} = \begin{bmatrix} \Theta_{xx} & \Theta_{xy} \\ \Theta_{yx} & \Theta_{yy} \end{bmatrix}$$
(14)

and

$$\Xi = \left\langle \boldsymbol{\varphi} s^T \right\rangle = \left[ \begin{array}{cc} \left\langle \boldsymbol{\varphi} s^T_x \right\rangle & \left\langle \boldsymbol{\varphi} s^T_y \right\rangle \end{array} \right] = \left[ \begin{array}{cc} \Xi_x & \Xi_y \end{array} \right]. \tag{15}$$

The covariance matrix are computed for the full square lenslet array. So for a lenslet array of linear size N, the matrix  $\Theta$  is made of 4 blocks of equal size  $N^2 \times N^2$ .

Fig. 1 shows a  $4 \times 4$  lenslet array with the location of the x and y measurements inside each lenslet. In Fig. 2, the matrix of the vectors between pairs of lenslet is drawn. This matrix is a two-level recursive block Toeplitz matrix (2RBT). It is made of  $4 \times 4$  blocks in a Toeplitz arrangement and each block is a  $4 \times 4$  Toeplitz matrix. A  $N \times N$  lenslet array will exhibit a matrix of vectors with  $N \times N$  blocks in a Toeplitz arrangement with each block a  $N \times N$  Toeplitz matrix.

The covariance matrix of the measurements  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  depend solely on the amplitude and direction of the vectors linking pairs of lenslet. So each matrix has the structure shown in Fig. 2. A  $N \times N$  Toeplitx matrix is fully defined with its first row and column i.e. 2N - 1 elements.  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  are each a two-level recursive block Toeplitz matrix so each is defined with only  $(2N - 1)^2$  elements instead of  $N^4$ .

In Fig. 2, the unique elements of  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  are those inscribed into the dash closed regions. Table 1 compares the number of elements in the matrices  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  between their full and compressed versions. The compression factor defined as the ratio between the total number of elements and the unique elements is also reported.

The MVP of a recursive block Toeplitz matrix can be perform very efficiently, manipulating only the  $(2N-1)^2$  unique elements and never expanded the matrix to its full size.<sup>7</sup> The computing cost is reduced down to  $O(N^2 \log(N))$  instead of  $O(N^4)$  traditionally.

Table 1. Number of elements in  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  for both the full and the Toeplitz-compressed matrices.

Ν	10	20	40	64	84	150
$\Theta_{xx}$ Full	10,000	160,000	2,560,000	16,777,216	49,787,136	506,250,000
$\Theta_{xx}$ 2RBT	361	1,521	$6,\!241$	$16,\!129$	$27,\!889$	89,401
Compression	28	105	410	1,040	1,785	$5,\!663$

$\stackrel{4}{\longleftrightarrow}^4$	$\stackrel{\$}{\longleftrightarrow}$	$\stackrel{\uparrow 12}{\longleftrightarrow}$	$\stackrel{16}{\longleftrightarrow}$	0	0	0	0
$\stackrel{3}{\longleftrightarrow}$	$\stackrel{7}{\longleftrightarrow}$	$\stackrel{\uparrow 11}{\longleftrightarrow}$	$\stackrel{15}{\longleftrightarrow}$	0	0	0	0
$\stackrel{2}{\longleftrightarrow}^{2}$	$\stackrel{6}{\longleftrightarrow}$	$\stackrel{10}{\longleftrightarrow}$	$\stackrel{14}{\longleftrightarrow}$	0	0	0	0
$\stackrel{1}{\longleftrightarrow}$	$\stackrel{5}{\longleftrightarrow}$	$\stackrel{9}{\longleftrightarrow}$	$\stackrel{\uparrow 13}{\longleftrightarrow}$	0	0	0	0

Figure 1. Measurement sampling with a  $4 \times 4$  lenslet array. In the left hand side figure, the red circles show the locations of the measurements with respect to the location of the wavefront samples (black dots).

The MVP in Eq. (12) can also be performed very efficiently with the algorithm in Ref. 7 if the  $\Xi_x$  and  $\Xi_y$  are both 2RBT matrices. Fig. 2 shows the sampling for the wavefront and the measurements for the usual Fried

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	245	26	27	28	<del>3</del> 2	<b>3</b> 3	34	35	<del>≺3</del> 9	<b>*</b> #0	<b>*</b> }1	42	<del>&lt; 4</del> 6	₹47	* 48	49
2	24	*	*	<b>↑</b>	<b>≩</b> 1	*	×	۲	<b>₽</b> 38	,   ←	▼	٢	<b>4</b> 5	, 	*	~
3	23	¥	٨	٨	30 7	¥	*	K	37 ×	K	←	ĸ	<b>x</b> 44	r	←	*
4	22	¥	۲	٨	29	¥	¥	*	36	K	K	←	43		~	←
5	1 <del>8</del>	19	20	21	*	٨	↑	Ť	*	٢	۲	1	*	ĸ	K	1
6	17	*	*	1	۷	٨	٨	ţ	¥	٠	٢	۲	K	*	K	٢
7	16	×	۶	я	¥	۲	٨	٨	¥	¥	*	۲	K	K	←	₹
8	15	¥	×	*	Ļ	¥	۴	٨	1	¥	¥	*	1	¥	K	*
9	1 <del>1&gt;</del>	12 <b>7</b>	13	14	>	*	1	1	*	٨	ţ	Ť	4	٣	۲	Ń
10	10 <u>.</u>	→	*	1	¥	*	*	1	¥	٨	٨	↑	¥	*	K	۲
11	٩	×	→	ѫ	¥	×	۶	1	¥	۴	٨	٨	¥	¥	*	٢
12	8	$\mathbf{M}$	1	$\rightarrow$	7	Å	×	۶	↓ ↓	¥	۴	*	1	¥	¥	*
13	4→	5-7	6	$\checkmark$	<b>→</b>	7	1	1	*	*	1	1		٨	↑	1
14	3-,		7	/	*	<b>→</b>	*	1	×	*	1	1	*	*	٨	Ŷ
15	2,	~	$\rightarrow$	~	<b>`</b> 4	×	→	7	1	×	٠	А	↓	۴	*	٠
16	1	$\searrow$	~	$\rightarrow$		7	×	<b>→</b>		À	×	*		¥	۷	*

Figure 2. WFS vectors of measurement pairs for  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  matrices.

geometry.<sup>8</sup> The measurement samples are represented with the red circles at the lenslet centers and the wavefront samples located on the black dots. The covariance matrix resulting from this geometry is a rectangular 2RBT matrix as shown in Fig. 3. Fig. 3 depicts the matrix of the vectors between all pairs of slopes and wavefront. This matrix is a rectangular 2RBT matrix. It is made of  $5 \times 4$  blocks in a Toeplitz arrangement and each block is a  $5 \times 4$  Toeplitz matrix. A  $N \times N$  lenslet array will exhibit a matrix of vectors with  $(N + 1) \times N$  blocks in a Toeplitz arrangement with each block a  $(N + 1) \times N$  Toeplitz matrix.

The covariance matrices  $\Xi_x$  and  $\Xi_y$  depend solely on the amplitude and direction of the vectors linking pairs of slopes and wavefronts. So each matrix has the structure shown in Fig. 3. A  $(N + 1) \times N$  Toeplitx matrix is fully defined with its first row and column i.e. 2N elements.  $\Xi_x$  and  $\Xi_y$  are each a two-level recursive block Toeplitz matrix so each is defined with only  $4N^2$  elements instead of  $(N + 1)^2N^2$ . The size of the matrices  $\Xi_x$ and  $\Xi_y$  for the different geometry are summarized in Table 2. The compression factor is the ratio between the full matrix in the Fried geometry and the compressed matrix.

Table 2. Number of elements in  $\Xi_x$  and  $\Xi_y$  for the Fried geometry, the 2RTB geometry and the Toeplitz-compressed matrices.

Ν	10	20	40	64	84	150
$N_{\boldsymbol{\varphi}}$	11	21	41	65	85	151
$\Xi_x$ Full	12,100	176,400	2,689,600	17,305,600	50,979,600	513,022,500
$\Xi_x 2 \text{RBT}$	400	$1,\!600$	6,400	$16,\!384$	28,224	90,000
Compression	30	110	420	1056	1806	5700

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	347	<b>3</b> 8	39 39	40	<b>\$</b> 5	<b>4</b> 6	\$7	$\lambda_8$	<b>≺</b> 53	\$54	55	56	← <del>6</del> 1	<b>*</b> 62	63	64
2	<b>3</b> 6	۲	٨	1	<b>4</b> 4	*	ĸ	1	<b>4</b> 52	*	K	*	<del>-6</del> 0	<u>+</u>	*	
3	3 <sub>4</sub> 5	۶	۲	۴	43 ¥	K	*	ĸ	× <sup>51</sup>	Ł	*	~	-59	4	←	*
4	34 •	¥	r	۲	42 4	¥	K	*	50		*	*	58		←	←
5	33	¥	¥	*	41	4	K	*	49	K	K	4	57			-
6	29	30	31	32	+	¥	1	Ţ	×	K	٢	Λ,	Ł	K	K	
7	248	4	1	1	٨	۲	۲	ţ	×	*	K	٢	*	*	K	r
8	27	4	4	1	¥	*	۲	۲	¥	¥	*	K	×	Ł	*	~
9	26	¥	4	4	Ļ	¥	٨	۲	1	¥	¥	*	~	K	¥	*
10	25	ł	4	4	l	¥	¥	+	1	4	¥	*		$\checkmark$	×	*
11	21	22	23	24	7	1	1	Ĩ	*	¥	1	Ţ	×	٢	٢	/
12	20	×	1	1	4	4	1	1	*	۲	۲	1	¥	ĸ	٢	٢
13	18	¥	7	*	¥	4	4	1	¥	٨	۲	۲	¥	K	₭	٢
14	18	×	*	¥	ł	¥	*	4	Ļ	¥	٢	۲	4	¥	¥	*
15	17	4	×	*	Ţ	ţ	¥	4	1	¥	¥	*	1	4	¥	*
16	137	14	15	16	7	*	1	/	7	1	1	Ţ	*	٨	ţ	Ţ
17	1 <del>8</del> →	->	7	1	×	7	1	1	4	4	1	1	Ł	۲	۲	1
18	14	->	->	7	×	*	¥	*	*	4	4	1	¥	۲	۲	۲
19	10	~	*	->	7	×	¥	¥	t	۲	4	7	¥	¥	٢	۲
20	9	$\mathbf{Y}$	~	*	. \.	7	×	*	ľ	ţ	۲	4		¥	¥	+
21	5-1	6	~	8	->	↗	7	/	7	1	1	/	4	1	1	ľ
22	4			. /	*	->	↗	1	*	Ŧ	1	1	•	4	1	t
23	3	$\rightarrow$		>	~	*	->	7	×	×	*	×	4	4	4	1
24	2	<u>``</u>	$\rightarrow$	. —		$\searrow$	*	->	1	×	¥	*	1	۷	4	4
25	1	<u> </u>	~	. –		$\searrow$	~	*		7	X	*		ţ	۲	4

Figure 3. vectors of WFS and measurement pairs for  $\Xi_x$  and  $\Xi_y$  matrices.

## 3.1 Matrix vector product

It has been shown that the matrices within  $\Theta$  and  $\Xi$  are 2RBT matrices. Each of  $\Theta_{xx}$ ,  $\Theta_{yy}$ ,  $\Theta_{xy}$  and  $\Theta_{yx}$  is now replaced by a single vector,  $t_{\Theta}^{xx}$ ,  $t_{\Theta}^{yy}$ ,  $t_{\Theta}^{yy}$ ,  $t_{\Theta}^{yy}$ , respectively; each vector with  $(2N-1)^2$  elements. In the same way,  $\Xi_x$  and  $\Xi_y$  are replaced with the vectors  $t_{\Xi}^x$  and  $t_{\Xi}^y$ , of length  $4N^2$ . The vector elements are ordered according to the numbered elements in the covariance matrices drawn in Fig. 2 and in Fig. 3. In the following, data will be filtered by the spectral contents of these vectors. So we replace them by they Fourier transform:  $\tilde{t} = \mathcal{F}t$  where  $\mathcal{F}$  is the Fourier operator. As all the vector t are real, only half of the elements of  $\tilde{t}$  need to be computed, as the other half is the complex conjugate of the first half.

The matrix  $\Theta$  and  $\Xi$  are now redefined by

$$\Theta \equiv \left\{ \tilde{T}_{\Theta} \right\} \tag{16}$$

with the matrix  $\tilde{T}_{\Theta}$  given by

and

$$\tilde{T}_{\Theta} = \begin{bmatrix} \tilde{t}_{\Theta}^{xx} \ \tilde{t}_{\Theta}^{xy} \ \tilde{t}_{\Theta}^{yx} \ \tilde{t}_{\Theta}^{yy} \end{bmatrix}$$
(17)

$$\Xi \equiv \left\{ \tilde{T}_{\Xi} \right\} \tag{18}$$

with the matrix  $\tilde{T}_{\Xi}$  given by

$$\tilde{T}_{\Xi} = \begin{bmatrix} \tilde{\boldsymbol{t}}_{\Xi}^x \ \tilde{\boldsymbol{t}}_{\Xi}^y \end{bmatrix}.$$
(19)

An algorithm for fast multiplication of recursive block Toeplitz matrix is derived in Ref. 7. A very brief summary of the method is given in following. Assuming a  $m^2 \times n^2$  2RBT matrix A i.e.  $A \equiv {\tilde{t}}$ , the matrixvector product Ax = y is obtained through the following steps (vectors x and y have  $n^2$  and  $m^2$  elements, respectively):

1. the vector  $\boldsymbol{x}$  of length  $n^2$  is assigned into a zero-valued vector  $\boldsymbol{\beta}$  of length  $(2n-1)^2$  according to the rules:

$$\boldsymbol{\beta}_{\mu(i,j)} = \boldsymbol{x}_{k(i,j)}, \ \forall \ 0 \le i, j < n,$$
(20a)

$$k(i,j) = in+j, \tag{20b}$$

$$\mu(i,j) = (m+n)(n-1) - i(m+n-1) - j, \qquad (20c)$$

2. the Fourier transform of  $\boldsymbol{\beta}$  is computed

$$\tilde{\boldsymbol{b}} = \mathcal{F}\boldsymbol{\beta},$$
 (21)

3. the vectors  $\tilde{t}$  and  $\tilde{b}$  are multiplied element wise

$$\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{t}}\boldsymbol{b},\tag{22}$$

4. the inverse Fourier transform of  $\tilde{\boldsymbol{b}}$  is computed

$$\boldsymbol{c} = \mathcal{F}^{-1} \tilde{\boldsymbol{c}} \tag{23}$$

5. elements of c are assigned into y according to the rules:

$$\boldsymbol{y}_{k(i,j)} = \omega_{k(i,j)} \boldsymbol{c}_{\boldsymbol{\xi}(i,j)}, \, \forall \, 0 \le i, j < m,$$
(24a)

$$k(i,j) = im + j, \tag{24b}$$

$$\xi(i,j) = (m+n)(m+n-1) - (i+1)(m+n-1) - (j+1), \qquad (24c)$$

#### Proc. of SPIE Vol. 9148 91480R-6

 $\omega_{k(i,j)}$  is zeros-ones valued vector indicating where in the square lenslet array slopes exist.

With the algorithm described above, Eq. (12) and Eq. (13) are re-written

$$\hat{\varphi} = \mathcal{F}^{-1} \left( \tilde{t}_{\Xi}^x \tilde{\zeta}_x + \tilde{t}_{\Xi}^y \tilde{\zeta}_y \right)$$
(25)

and

$$\Theta \boldsymbol{\zeta} = \begin{bmatrix} \mathcal{F}^{-1} \left( \tilde{\boldsymbol{t}}_{\Theta}^{xx} \tilde{\boldsymbol{\zeta}}_{x} + \tilde{\boldsymbol{t}}_{\Theta}^{xy} \tilde{\boldsymbol{\zeta}}_{y} \right) \\ \mathcal{F}^{-1} \left( \tilde{\boldsymbol{t}}_{\Theta}^{xx} \tilde{\boldsymbol{\zeta}}_{x} y + \tilde{\boldsymbol{t}}_{\Theta}^{yy} \tilde{\boldsymbol{\zeta}}_{y} \right) \end{bmatrix} = \boldsymbol{s}$$
(26)

with  $\tilde{\zeta}_x = \mathcal{F}\zeta_x$  and  $\tilde{\zeta}_y = \mathcal{F}\zeta_y$ , assuming that the vector  $\hat{\varphi}$  and s have been sized and ordered appropriately.

# 3.2 Derivation of the covariance matrices

#### 3.2.1 Wavefront sensor slope covariance

Shack–Hartmann WFS centroids are a measure of the spatial derivatives of the wavefront averaged on each subaperture. These derivatives are often referred as angle of arrivals:

$$s_x = \frac{\lambda f}{2\pi d^2} \frac{\partial \varphi}{\partial x} * \Pi\left(\frac{x}{d}\right) \Pi\left(\frac{y}{d}\right), \qquad (27)$$

$$s_y = \frac{\lambda f}{2\pi d^2} \frac{\partial \varphi}{\partial y} * \Pi\left(\frac{x}{d}\right) \Pi\left(\frac{y}{d}\right), \qquad (28)$$

where  $\lambda$  is the sensing wavelength, f the lenslet focal length and d the linear size of one subaperture.

The covariance matrix in Eq. (14) are derived from

$$B_{\boldsymbol{\alpha}\cdot\boldsymbol{\beta}}(\boldsymbol{r}) = \mathcal{F}\left[W_{\boldsymbol{\alpha}\cdot\boldsymbol{\beta}}(\boldsymbol{\zeta})H^2(\boldsymbol{\zeta})\right](\boldsymbol{r})$$
(29)

The power spectrum density of the slopes is given by:<sup>9</sup>

$$W_{\boldsymbol{\alpha}\cdot\boldsymbol{\beta}}(\boldsymbol{\zeta}) = \lambda^2 \left(\boldsymbol{\zeta}\cdot\boldsymbol{\alpha}\right) \left(\boldsymbol{\zeta}\cdot\boldsymbol{\beta}\right) W_{\varphi}(\boldsymbol{\zeta}).$$
(30)

 $\alpha$  and  $\beta$  are unit vector of coordinates [1,0] and [0,1], respectively.  $W_{\varphi}(f)$  is the wavefront power spectrum density<sup>10</sup> given by

$$W_{\varphi}(f) = 0.0229 r_0^{-5/3} \left( f^2 + \frac{1}{\mathcal{L}_0^2} \right)^{-11/6}$$
(31)

and

$$H(\boldsymbol{\zeta}) = \operatorname{sinc}(\pi d\boldsymbol{\zeta}_y) \operatorname{sinc}(\pi d\boldsymbol{\zeta}_y).$$
(32)

## 3.2.2 Wavefront and slopes cross-correlation

The cross-correlation in Eq. (15) are obtained by setting

$$B_{\varphi\alpha}(\mathbf{r}) = \mathcal{F}\left[W_{\varphi\alpha}(\boldsymbol{\zeta})H(\boldsymbol{\zeta})\right](\mathbf{r})$$
(33)

The cross–power spectrum density is given by:

$$W_{\varphi\alpha}(\vec{f}) = -\iota\lambda\left(\boldsymbol{\zeta}\cdot\boldsymbol{\alpha}\right)W_{\varphi}(\boldsymbol{\zeta}) \tag{34}$$

#### Proc. of SPIE Vol. 9148 91480R-7



Figure 4. The vector of the T matrix for  $\Theta$  (left). The vector of the T matrix for  $\Xi$  (right).

#### 3.2.3 Numerical estimation

In the former section, we gave the expression for the covariance of the slopes and the cross-correlation between the wavefront and the slopes. The covariances need to be computed only for the unique elements in the corresponding matrices as shown in Fig. 2 and Fig. 3. In Fig. 4, all the unique vectors in Fig. 2 and Fig. 3, respectively in the left and right hand side panels, are put together, all coming out from the same point of origin. From Fig. 4, it is clear that these vectors sample a 2D covariance map with sample points at (id, jd),  $\forall i, j \in [1 - N, ..., N - 1]$  for  $\Theta$  and at ((i - 1/2)d, (j - 1/2)d),  $\forall i, j \in [1 - N, ..., N]$  for  $\Xi$ .

The covariances are derived from the Fourier transform of the power spectrum density (PSD) (Eq. (29) and Eq. (33)). The power spectrum must be sampled such as its Fourier transform gives accurate and unbiased values of the covariance. Based on sampling requirements of the covariance, the largest spatial frequency is set to  $f_{max} = \kappa/2d$  with  $\kappa$  an integer,  $\kappa \geq 1$ , and the sampling resolution is  $N_F \geq N$ , leading to the sampling frequency

$$(k,l)\frac{\kappa}{d(N_F-1)}, \ \forall (k,l) \in \left[\frac{1-N_F}{2}, \dots, \frac{N_F-1}{2}\right].$$

The accuracy of the covariance depends on the choice of both  $\kappa$  and  $N_F$ . From the 2D Fourier transform of the PSD, the covariance is extracted at index given by  $(i\kappa, j\kappa)$ ,  $\forall i, j \in [1 - N, \dots, N - 1]$  for  $\Theta$  and by  $(i\kappa, j\kappa)$ ,  $\forall i, j \in [1 - N, \dots, N]$  for  $\Xi$ . Fig. 5 and Fig. 6 shows the compressed version of the matrix  $\Theta$  and  $\Xi$ , respectively.



Figure 5.  $\Theta$  2RBT matrix for N = 40.



Figure 6.  $\equiv$  2RBT matrix for N = 40.

In Fig. 7, full and 2RBT MVP are compared for different lenslet array sizes. The runtime of MVP is plotted in the left hand side graph and the memory requirements is plotted in the right hand side graph. The results for the full matrix and for the 2RBT compressed matrix are drawn in blue and green, respectively. The circle and square markers correspond to the matrix  $\Theta$  and  $\Xi$ , respectively.

The runtime has been measured on a NVIDIA Tesla M2090 GPU card. The MVP for the full matrices has been implemented with CUBLAS 5.5 and the MVP for the compressed 2RBT matrices is using CUFFT 5.5.

Looking first at the memory requirement, the 2RBT matrices never exceed 10MB whereas the full matrices rapidly required several GB. Due to the large memory requirement for the full matrices and the limited amount of memory available, 5GB on the GPU, the MVP for the full matrix was limited to N = 84 for both  $\Theta$  and  $\Xi$ .

The MVP for both full matrices takes several milliseconds when N reach 64 whereas it takes only a few 100 microseconds up to N = 150 for the 2RBT compressed matrix.

The runtime for the MVP with the 2RBT matrices is dominated by the time spent to compute the Discrete Fourier Transform (DFT). For example, the runtime for the MVP for the matrix  $\Xi_{64}$  and  $\Xi_{84}$  are nearly the same because their DFT sizes are identical. The MVP of  $\Theta$  requires the computation of 6 DFTs compared to 3 for  $\Xi$  (Eq. (26) and Eq. (25)). However the DFT execution are batched whenever possible.



Figure 7. MVP runtime and memory requirement.

# 4. WAVEFRONT ESTIMATION

The estimation of the wavefront  $\hat{\varphi}$  from the wavefront sensor measurements *s* is given by Eq. (25) and Eq. (26). Eq. (25) is a MVP that can be done very efficiently as shown in the former section. In Eq. (25),  $\zeta$  is solution of the linear system of equation given in Eq. (26). In the following, Eq. (26) is solved using iterative methods.<sup>11</sup> The matrix  $\Theta$  is symmetric and positive definite. An iterative method of choice for this type of matrices is the conjugate gradient.<sup>12</sup> The conjugate gradient algorithm is written:

- initialization:
  - 1.  $r = Ax_0$
  - 2.  $r_0 = b r_0$
  - 3.  $p_0 = r_0$
- loop from k = 0:
  - 1.  $q = Ap_k$ 2.  $\gamma = r_k^T r_k$ 3.  $||r||_2 = \sqrt{\gamma}$ 4.  $\alpha_k = \frac{\gamma}{p_K^T q}$ 5.  $x_{k+1} = x_k + \alpha_k p_k$ 6.  $r_{k+1} = r_k - \alpha_k q$ 7.  $\beta_k = \frac{r_{k+1}^T r_{k+1}}{\gamma}$ 8.  $p_{k+1} = r_{k+1} + \beta_k p_k$

The convergence of the conjugate gradient method is proportional to the condition number of  $\Theta$ .<sup>13</sup> Fig. 8 gives the condition number for matrix  $\Theta$  as a function of the lenslet pitch. This matrix is obviously ill–conditioned. To remedy to this situation, pre–conditioner can be employed to improve the conditioning of the system and



Figure 8. Condition number of the matrix  $\Theta$  as a function of the lenslet array pitch for 3 lenslet array size 20, 40 and 64.

hence increasing the converging rate of the conjugate gradient. Alternatively, another iterative method could be used. The MINRES algorithm<sup>14–16</sup> has been shown to have better convergence properties for ill–conditionned positive definite matrices. The MINRES algorithm is written:

Altitude[m]	25	275	425	1250	4000	8000	13000
$\xi_0$	0.1257	0.0874	0.0666	0.3498	0.2273	0.0681	0.0751
$\ ec{v}\ [\mathrm{m/s}]$	5.6540	5.7964	5.8942	6.6370	13.2925	34.8250	29.4187
$\operatorname{arg}\left(\vec{v}\right)\left[\operatorname{rd}\right]$	0.0136	0.1441	0.2177	0.5672	1.2584	1.6266	1.7462
Table 3. 7 layers atmosphere profile.							

• initialization:

- 1.  $r = Ax_0$ 2.  $\nu_1 = b - \nu_1$ 3.  $\beta_1 = \|\nu_1\|_2$ 4.  $\eta = \beta_1, \ \gamma_1 = \gamma_0 = 1, \ \sigma_1 = \sigma_0 = 0, \ \nu_0 = 0$ 5.  $\nu_0 = 0, \ w_0 = w_{-1} = 0$
- loop from k = 1:
  - 1.  $\nu_i = \nu_i / \beta_i$ 2.  $\nu_{i+1} = A\nu_i$ 3.  $\alpha_i = \nu_i^T \nu_{i+1}$ 4.  $\nu_{i+1} = \nu_{i+1} - \alpha_i \nu_i - \beta_i \nu_{i-1}$ 5.  $\delta = \gamma_i \alpha_i - \gamma_{i-1} \sigma_i \beta_i$ 6.  $\rho_2 = \sigma_i \alpha_i + \gamma_{i-1} \gamma_i \beta_i$ 7.  $\rho_3 = \sigma_{i-1}\beta_i$ 8.  $\beta_{i+1} = \|\nu_{i+1}\|_2$ 9.  $\rho_1 = \sqrt{\delta^2 + \beta_{i+1}^2}$ 10.  $\gamma_{i+1} = \delta/\rho_1$ 11.  $\sigma_{i+1} = \beta_{i+1}/\rho_1$ 12.  $w_i = (\nu_i - \rho_3 w_{i-2} - \rho_2 w_{i-1})/\rho_1$ 13.  $x_i = x_{i-1} + \gamma_{i+1} \eta w_i$ 14.  $||r_i||_2 = |\sigma_{i+1}| ||r_{i-1}||_2$ 15.  $\eta = \sigma_{i+1}\eta$ 16.

# 5. SIMULATIONS

The simulations were done for single conjugated adaptive optics system (SCAO) running in open-loop. All the test runs are noiseless. The same 7 layers atmospheric profile (Table 3) is used for all the tests with the same outer scale set to  $\mathcal{L}_0 = 30$ m. The parameters of the 5 test cases are given in Table 4.

	Case	Ι	II	III	IV	V
	Ν	20	40	64	84	150
	D[m]	3.6	8	5	42	30
	d[cm]	18	20	7.5	50	20
Tab	ole 4. The	e simu	lation	test c	ase p	arameters.
Ν	runtim	e [ms]	#	solve	r it.	WFE[nm]
20	3.(	)1		18		36
40	4.4		26		39	
64	11.	18		58		18
84	3.7	3.75				82
150	18.		47		41	

Table 5. Open-loop performance for a sampling rate of 500Hz and no frame delay.

## 5.1 Iterative solver convergence

For this test, one sets  $d/r_0 = 1$ . For each test, Eq. (26) was solved numerically with the GG and MINRES algorithms for an increasing number of iterative step from 1 to 200. The wavefront is then estimated with Eq. (25) and the wavefront error as a function of the number of iterative steps is shown in Fig. 9. The MINRES algorithm exhibits a smaller residue but also the residue decrease quasi-monotonically with the number of iterations while the residue from the CG algorithm decreases more slowly and in a more chaotical manner. The MINRES algorithm always reach the limit corresponding to the fitting error whereas the CG algorithm is able to reach the fitting error limit only for one case (N = 20). It is worth noting that, for the MINRES algorithm, the WFE is reduced by a factor 5 per decade of iterations for all the cases.

These results demonstrate that MINRES always converges faster that CG, but this convergence is still rather slow with several 10 of iterations required to reach the minimal WFE.

## 5.2 Open-loop

For the open-loop tests,  $r_0$  is set to 15cm. The wavefront and wavefront gradient are sampled at 500Hz and the open-loop simulations are run for 200 steps. There is no servo-lag error meaning there is no time delay between the wavefront estimate and the on-axis NGS wavefront. The iterative solvers are set such as the first guess of the iterative solvers is the previous estimate. Table 5 and Table 6 give, for each case, the computing time for the wavefront reconstruction, the number of iterations required by the solver and the wavefront error (WFE).

For Table 5, the MINRES algorithm is set such as it stop itself when it reaches the minimum WFE. The highly sampled pupil N = 64 and N = 150 are the ones converging the slowest as expected. For a 500Hz SCAO system, the wavefront reconstruction computing time should be less than 2ms. In Table 6, for the cases N = 20, 40 and 82, the number of iterations in MINRES is set such as the computing time is less than 2ms. This results in a slight increase of the WFE. Fig. 10 compares, for N = 150, the true" wavefront to the reconstructed wavefront from left to right in the top graph and gives the residual wavefront error map in the bottom graph.

# 6. CONCLUSION AND FUTURE WORK

A new method to estimate a wavefront with optical turbulence aberrations has been presented. This method belongs to the LMMSE class of methods. The method achieves small computer memory requirements and fast wavefront reconstruction by exploiting the recursive Toeplitz structure of the stochastic covariance matrices. It

Ν	runtime [ms]	# solver it.	WFE[nm]
20	1.70	10	37
40	1.82	10	48
64	-	-	-
84	1.93	8	92
150	_	-	-

Table 6. Open-loop performance for a sampling rate of 500Hz and no frame delay with limited number of iterations.



Figure 9. Residual wavefront error as a function of the number of iterations of the CG and MINRES solvers.



Figure 10. Top left: on–axis wavefront; Top right: estimated on–axis wavefront; Bottom: Residual wavefront error for N = 150.

is shown that on a single NVIDIA GPU, the required storage of the matrices never exceed 10MB for SCAO system up to  $150 \times 150$  and the MVP of the covariances matrices can be performed in a few  $100\mu$ s. For all the SCAO systems, using the MINRES iterative solver, they converge to the fitting error in a few milli–seconds making the methods real-time ready for a few of the simulated SCAO systems.

The CUDA code for the wavefront estimator is freely available at http://github.com/rconan/CEO.

As for future work, the use of preconditioners to accelerate the convergence of the iterative solver is going to be investigated as well as the tayloring of the algorithm for real-time application using multiple GPUs.

#### REFERENCES

- Poyneer, L. A., Gavel, D. T., and Brase, J. M., "Fast wave-front reconstruction in large adaptive optics systems with use of the fourier transform," J. Opt. Soc. Am. A 19(10), 2100–2111 (2002).
- [2] Rosensteiner, M., "Cumulative reconstructor: fast wavefront reconstruction algorithm for extremely large telescopes," J. Opt. Soc. Am. A 28, 2132–2138 (Oct 2011).
- [3] Ellerbroek, B. L., "Efficient computation of minimum-variance wave-front reconstructors with sparse matrix techniques," J. Opt. Soc. Am. A 19(9), 1803–1816 (2002).
- [4] Thiébaut, E. and Tallon, M., "Fast minimum variance wavefront reconstruction for extremely large telescopes," J. Opt. Soc. Am. A 27, 1046–1059 (May 2010).
- [5] Kay, S., [Fundamentals of Statistical Signal Processing: Estimation Theory], Prentice Hall, New-Jersey (2010).
- [6] Vidal, F., Gendron, E., and Rousset, G., "Tomography approach for multi-object adaptive optics," J. Opt. Soc. Am. A 27, A253–A264 (Nov 2010).

- [7] Lee, D., "Fast multiplication of a recursive block toeplitz matrix by a vector and its application," *Journal of Complexity* 2, 295–305 (1968).
- [8] Fried, D. L., "Least-square fitting a wave-front distortion estimate to an array of phase-difference measurements," J. Opt. Soc. Am. 67(3), 370–375 (1977).
- [9] Conan, R., J.Borgnino, Ziad, A., and Martin, F., "Analytical solution for the covariance and for the coherence time of the angle of arrival of a wave front corrugated by atmospheric turbulence," J. Opt. Soc. Am. A 17, 1807–1818 (Octobre 2000).
- [10] Conan, R., [Importance de l'échelle externe pour la Haute Résolution en Astronomie], EUE, Sarrebruck, Germany (2011).
- [11] Golub, G. and Van Loan, C., [Matrix Computations], The Johns Hopkins University Press (1989).
- [12] Saad, Y., [Iterative Methods for Sparse Linear Systems], SIAM, Philadelphia, PA (2003).
- [13] Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J. M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and der Vorst", H. V., [*Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*], SIAM, Philadelphia, PA (1994).
- [14] Paige, C. C. and Saunders, M. A., "Solution of sparse indefinite systems of linear equations," SIAM J. Numerical Analysis 12, 617–629 (1975).
- [15] van der Vorst, H. A., [Iterative Krylov Methods for Large Linear Systems], Cambridge University Press, Cambridge (2003).
- [16] Choi, S.-C., Iterative Methods for Singular Linear Equations and Least-Squares Problems, PhD thesis, Stanford University (2006).