Submitted to **Journal of the European Optical Society – Rapid Publications, JEOS-RP**
Template provided by EDP Sciences

RESEARCH ARTICLE

Zernike by One *Pascal* Triangle

A high performance, low memory cost and flexible computation scheme for *Zernike* polynomials

Wei-Jun CHEN[1]*

[1] Carl Zeiss Meditec AG, Göschwitzer Str. 51–52, 07745 Jena, Germany

Abstract – This work uncovers two hidden cases of block–wise recurrence in *Zernike* computations. Based on these findings, a new computation scheme for *Zernike* polynomials is proposed. It uses one *Pascal*'s triangle for all internal factors, thus avoiding the computation of factorials, *cos*/*sin* functions, and matrix inversions. It offers both a direct transformation method and a block–wise recursive method for calculating *Zernike* basis functions, thereby fulfilling the requirements for high accuracy, high speed, low memory footprint, and great application flexibility.

Keywords. *Zernike* polynomials, *Pascal* triangle, homogeneous bivariate polynomials, block–wise recurrence, recursive method, surface/wavefront reconstruction

1. Introduction

Zernike polynomials play a key role in many optical applications [1][2][3] due to their two main features: 1. the orthogonality between individual polynomial components and 2. the direct correspondence with optical *Seidel* aberrations. *Zernike* polynomials are usually defined as

$$z_n^{\pm m}(\rho,\varphi) \;=\; R_n^m(\rho)\begin{cases}\cos(m\varphi) & \text{for } +m\\ \sin(m\varphi) & \text{for } -m\end{cases}, \tag{1}$$

$$R_n^m \;=\; \sum_{k=0}^{(n-m)/2}\left(\frac{(-1)^k(n-k)!}{k!\left(\frac{n+m}{2}-k\right)!\left(\frac{n-m}{2}-k\right)!}\rho^{n-2k}\right), \tag{2}$$

where R_n^m is called the *Zernike* radial polynomial. The above equations appear in the literature [4][5] with different notations for the m–term. This work adopts the $\pm m$ notation since this notation provides an explicit and convenient way to divide *Zernike* components of order n into their even part ($\pm m = \pm 2l$), and their odd part $\pm m = \pm(2l+1)$, and to subdivide each part into distinct *cos* and *sin* subparts. Incidentally, all odd parts of the components for the even order ($n = 2j$) and all even parts of the components for the odd order ($n = 2j+1$) are discarded to avoid non–integer factorials

* Corresponding author: wei-jun.chen@zeiss.com

$n=0$							1					
$n=1$						1		1				
$n=2$					1		2		1			
$n=3$				1		3		3		1		
$n=4$			1		4		6		4		1	
$n=5$		1		5		10		10		5		1
$n=6$	1		6		15		20		15		6		1

Table 1. A standard 6–row *Pascal* triangle and its right half vectors

in *Zernike* radial polynomials (Eq. 2), and all index terms n, m, l, j, k in this work are non–negative integers, and $l \leq j$ and $k \leq (j-l)$.

If the polynomial order is not high, the calculation of *Zernike* polynomials is usually satisfied by the definition formulas. For *Zernike* radial polynomials a **three–term–recurrence–relation** ($TTRR$) exists and has been applied component–wise to higher–order recursive *Zernike* calculations [6][7]:

$$R_n^m = \rho\Big(R_{n-1}^{|m-1|}(\rho) + R_{n-1}^{m+1}(\rho)\Big) - R_{n-2}^m(\rho). \tag{3}$$

On the other hand, *Zernike* polynomials are essentially one of the orthogonalized versions of the homogeneous bivariate polynomials in a *Cartesian* coordinate system with a unit circle and a coordinate center. In practice, the computation of *Zernike* polynomials is often based on a look–up–table (LUT) of the corresponding *Cartesian* forms of the individual *Zernike* components [3][8][9], although the complexity of such *Cartesian* forms increases sharply with polynomial order.

To achieve greater flexibility compared to component–wise recursive computations, and to lower the limits of polynomial order and complexity in *Cartesian* form based computations, this work uncovers two cases of block–wise recurrence in *Zernike* computations[1] and offers both a direct block–wise transformation method and a block–wise recursive method for matrix–form *Zernike* computations. The application of such block–wise methods promises higher computational speed and lower memory footprint without compromising accuracy.

The rest of this paper is organized as follows: The next section presents two hidden block–wise recurrences along with their close relationship to an n–row *Pascal*'s triangle; subsequently Sec. 3 explores a new computation scheme for *Zernike* polynomials, which covers basis function computation, coefficient determination, polar/*Cartesian* coordinate conversion, and application specific derivative analysis, among others; further technical discussions can be found in Sec. 4, where a preliminary performance evaluation is conducted in term of computational complexity, accuracy stability, memory requirements, and application flexibility; finally, Sec. 5 concludes this paper with an outlook on future works.

2. *Zernike* polynomials supported by one *Pascal*'s triangle

2.1. *Pascal* triangle supported block–wise recurrence

The *Zernike* radial polynomials of the first $n \leq 6$ orders can be expressed in matrix form, where all radial polynomials of an order are obtained by multiplying a coordinate value irrelevant factor

[1] To our knowledge, block–wise recurrence in *Zernike* calculations has hardly been addressed in the literature, although there is a long history of intensive studies on *Zernike* polynomials.

$n=0$	1						
$n=1$	1	1					
$n=2$	1	2	1				
$n=3$	1	3	3	1			
$n=4$	1	4	6	4	1		
$n=5$	1	5	10	10	5	1	
$n=6$	1	6	15	20	15	6	1

Table 2. A left–aligned 6–row *Pascal* triangle and its anti–diagonal vectors

matrix by a power vector of the radial coordinate values, as:

$$\begin{cases} R_0^0(\rho)=1 \\ R_1^1(\rho)=\rho \\ R_2^0(\rho)=2\rho^2-1 \\ R_2^2(\rho)=\rho^2 \\ R_3^1(\rho)=3\rho^3-2\rho \\ R_3^3(\rho)=\rho^3 \\ R_4^0(\rho)=6\rho^4-6\rho^2+1 \\ R_4^2(\rho)=4\rho^4-3\rho^2 \\ R_4^4(\rho)=\rho^4 \\ R_5^1(\rho)=10\rho^5-12\rho^3+3\rho \\ R_5^3(\rho)=5\rho^5-4\rho^3 \\ R_5^5(\rho)=\rho^5 \\ R_6^0(\rho)=20\rho^6-30\rho^4+12\rho^2-1 \\ R_6^2(\rho)=15\rho^6-20\rho^4+6\rho^2 \\ R_6^4(\rho)=6\rho^6-5\rho^4 \\ R_6^6(\rho)=\rho^6 \end{cases} \Longrightarrow \begin{cases} R_0^{\{m\}}=1\times\rho^0 \\ R_1^{\{m\}}=1\times\rho^1 \\ R_2^{\{m\}}=\begin{bmatrix}2&-1\\1&0\end{bmatrix}\times\begin{bmatrix}\rho^2\\\rho^0\end{bmatrix} \\ R_3^{\{m\}}=\begin{bmatrix}3&-2\\1&0\end{bmatrix}\times\begin{bmatrix}\rho^3\\\rho^1\end{bmatrix} \\ R_4^{\{m\}}=\begin{bmatrix}6&-6&1\\4&-3&0\\1&0&0\end{bmatrix}\times\begin{bmatrix}\rho^4\\\rho^2\\\rho^0\end{bmatrix} \\ R_5^{\{m\}}=\begin{bmatrix}10&-12&3\\5&-4&0\\1&0&0\end{bmatrix}\times\begin{bmatrix}\rho^5\\\rho^3\\\rho^1\end{bmatrix} \\ R_6^{\{m\}}=\begin{bmatrix}20&-30&12&-1\\15&-20&6&0\\6&-5&0&0\\1&0&0&0\end{bmatrix}\times\begin{bmatrix}\rho^6\\\rho^4\\\rho^2\\\rho^0\end{bmatrix} \end{cases} \tag{4}$$

where $R_n^{\{m\}}$ denotes a column vector $\left[R_{2j}^0\ R_{2j}^2\ \cdots\ R_{2j}^{2j}\right]^T$ for even orders ($n=2j$) or $\left[R_{2j+1}^1\ R_{2j+1}^3\ \cdots\ R_{2j+1}^{2j+1}\right]^T$ for odd orders ($n=2j+1$).

From the above equation seven weight vectors, $\{w_n^{\{k\}}|n=0,1,...,6\}$ with $0\le k\le j$, can be extracted from the corresponding order–specific factor matrices (Eq. 5). A block–wise recurrence among the de–weighted factor matrices ($\{T_n|n=0,1,...,6\}$) can be observed: the 0^{th} order de–weighted factor matrix appears in the 2^{nd} order de–weighted factor matrix, while the later subsequentially appears in the 4^{th} order de–weighted factor matrix, which in turn appears in the 6^{th} order de–weighted factor

matrix. The same applies to odd orders from the 1^{st} order to the 5^{th} order, as follows:

$$\begin{cases} R_0^{\{m\}} = 1 \times \mathrm{diag}(1) \times \rho^0 \\ R_1^{\{m\}} = 1 \times \mathrm{diag}(1) \times \rho^1 \\ R_2^{\{m\}} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \times \mathrm{diag}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}^T \right) \times \begin{bmatrix} \rho^2 \\ \rho^0 \end{bmatrix} \\ R_3^{\{m\}} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \times \mathrm{diag}\left(\begin{bmatrix} 1 \\ -2 \end{bmatrix}^T \right) \times \begin{bmatrix} \rho^3 \\ \rho^1 \end{bmatrix} \\ R_4^{\{m\}} = \begin{bmatrix} 6 & 2 & 1 \\ 4 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \times \mathrm{diag}\left(\begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}^T \right) \times \begin{bmatrix} \rho^4 \\ \rho^2 \\ \rho^0 \end{bmatrix} \\ R_5^{\{m\}} = \begin{bmatrix} 10 & 3 & 1 \\ 5 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \times \mathrm{diag}\left(\begin{bmatrix} 1 \\ -4 \\ 3 \end{bmatrix}^T \right) \times \begin{bmatrix} \rho^5 \\ \rho^3 \\ \rho^1 \end{bmatrix} \\ R_6^{\{m\}} = \begin{bmatrix} 20 & 6 & 2 & 1 \\ 15 & 4 & 1 & 0 \\ 6 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \mathrm{diag}\left(\begin{bmatrix} 1 \\ -5 \\ 6 \\ -1 \end{bmatrix}^T \right) \times \begin{bmatrix} \rho^6 \\ \rho^4 \\ \rho^2 \\ \rho^0 \end{bmatrix} \end{cases} \Longrightarrow R_{n\leq 6}^{\{m\}} = T_n \times \mathrm{diag}(w_n^{\{k\}}) \times \begin{bmatrix} \rho^n \\ \rho^{n-2} \\ \vdots \\ \rho^{(n \mod 2)} \end{bmatrix}, \quad (5)$$

where $\mathrm{diag}(w_n^{\{k\}})$ denotes the diagonalization of a vector to a diagonal matrix, as

$$\mathrm{diag}(w_n^{\{k\}}) = \begin{bmatrix} w_n^0 & 0 & \cdots & 0 \\ 0 & w_n^1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n^j \end{bmatrix}. \quad (6)$$

It appears that for order $n \geq 2$, the de–weighted factor matrix T_n can be interpreted as an extended T_{n-2}, with zero–padding at the bottom and a factor vector $t_n^{\{m\}}$ attached to its left side, as described in the following equation:

$$\begin{cases} T_0 = 1 \\ T_1 = 1 \\ T_{n\leq 6} = \begin{bmatrix} t_n^{\{m\}} & \begin{bmatrix} T_{n-2} \\ 0 \end{bmatrix} \end{bmatrix} \end{cases}. \quad (7)$$

Note that both the extracted weight vector $w_n^{\{k\}}$ and the de–weighted factor vector $t_n^{\{m\}}$ are closely related to an n–row *Pascal* triangle. In particular, $t_n^{\{m\}}$ corresponds to the right half of the n^{th} row vector in a standard *Pascal* triangle (Table 1), i.e., here we have $t_6^{\{m\}} = [20\ 15\ 6\ 1]^T$, $t_5^{\{m\}} = [10\ 5\ 1]^T$, $t_4^{\{m\}} = [6\ 4\ 1]^T$, $t_3^{\{m\}} = [3\ 1]^T$, $t_2^{\{m\}} = [2\ 1]^T$, $t_1^{\{m\}} = 1$ and $t_0^{\{m\}} = 1$ for radial polynomials of orders $n \leq 6$; On the other hand, all extracted weight vectors $w_n^{\{k\}}$ can be found as corresponding anti–diagonal vectors in the left–aligned *Pascal* triangle (Table 2), as $w_{0,1}^{\{k\}} = 1$, $w_2^{\{k\}} = [1\ -1]$, $w_3^{\{k\}} = [1\ -2]$, $w_4^{\{k\}} = [1\ -3\ 1]$, $w_5^{\{k\}} = [1\ -4\ 3]$, $w_6^{\{k\}} = [1\ -5\ 6\ -1]$, where all odd elements (numbered from 0 onwards) are additionally assigned a negative sign.

To our knowledge, the above–mentioned discoveries are rarely discussed in the literature. Therefore, it is necessary to prove that these discoveries, i.e., the separation of coordinate–relevant and –irrelevant computations (Eq. 4), the extraction of order–specific weight vectors (Eq. 5), the block–wise recurrence within de–weighted factor matrices (Eq. 7), and the close relationship to an n–row *Pascal* triangle (Tables 1 and 2), hold for *Zernike* radial polynomials without order limit (i.e., $\forall n > 6$); if possible and expected, this also holds for complete *Zernike* polynomials (i.e., radial $\times$ azimuth polynomials).

2.2. From canonical to block–wise *Zernike* calculation

The canonical definition of *Zernike* radial polynomials, Eq. 2, has its binomial form:

$$R_n^m = \sum_{k=0}^{(n-m)/2} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-m}{2}-k} \rho^{n-2k}. \tag{8}$$

If we swap the terms m and k in the second binomial term, we obtain

$$R_n^m = \sum_{k=0}^{(n-m)/2} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n-2k}{2}-\frac{m}{2}} \rho^{n-2k}. \tag{9}$$

In this equation, the term $\binom{n-k}{k}$ explicitly denotes the anti–diagonal vectors in a left–aligned *Pascal* triangle, and the term $\binom{n-2k}{\frac{n-2k}{2}-\frac{m}{2}}$ denotes the left and right halves of the $(n-2k)^{th}$–row in a standard *Pascal* triangle, since *Pascal*'s triangle has its mirror symmetry with respect to its center position $\frac{n-2k}{2}$.

By Combining *de Moivre*'s formula $(\cos m\varphi + i \sin m\varphi) = (\cos\varphi + i \sin\varphi)^m$ with the radial polynomials and converting the coordinate system from polar to *Cartesian* coordinates, we obtain a direct block–wise transformation method for calculating *Zernike* basis functions in the *Cartesian* coordinate system. If we define $X_j^{\{q\}} = \left[x^{2j}y^0 \;\; x^{2j-2}y^2 \;\; \cdots \;\; x^0y^{2j}\right]^T$, and set $w_n^k = (-1)^k\binom{n-k}{k}$ and $t_n^{\{l=m/2\}} = \binom{n-2k}{\frac{n-2k}{2}-\frac{\{m\}}{2}}$, where $\{l\}$ denotes the index vector as $\{l\} = [0 \;\; 1 \;\; \cdots \;\; n/2]^T$ for even orders or $\{l\} = [0 \;\; 1 \;\; \cdots \;\; (n-1)/2]^T$ for odd orders, then such a direct block–wise transformation is represented as a transformation of homogeneous bivariate (xy) polynomials into complete *Zernike* polynomials, as the following:

$$\begin{cases} z_{2j}^{-\{2l\}} = \sum_{k=0}^{j} w_{2j}^k \left(t_{2(j-k)}^{\{l\}} \odot B_{2(j-k)}^{-}\right) \times X_{(j-k)}^{\{q\}} x^{-1}y \\ z_{2j}^{+\{2l\}} = \sum_{k=0}^{j} w_{2j}^k \left(t_{2(j-k)}^{\{l\}} \odot B_{2(j-k)}^{+}\right) \times X_{(j-k)}^{\{q\}} \\ z_{2j+1}^{-\{2l+1\}} = \sum_{k=0}^{j} w_{2j+1}^k \left(t_{2(j-k)+1}^{\{l\}} \odot B_{2(j-k)+1}^{-}\right) \times X_{(j-k)}^{\{q\}} y \\ z_{2j+1}^{+\{2l+1\}} = \sum_{k=0}^{j} w_{2j+1}^k \left(t_{2(j-k)+1}^{\{l\}} \odot B_{2(j-k)+1}^{+}\right) \times X_{(j-k)}^{\{q\}} x \end{cases}, \tag{10}$$

where the terms of B_{2j}^{-}, B_{2j}^{+}, B_{2j+1}^{-} and B_{2j+1}^{+} denote four coordinate transformation matrices from the polar coordinate system to the *Cartesian* coordinate system for corresponding cos/sin components in

Zernike basis functions of even and odd order. These four transformation matrices form **the second block–wise recurrence** in *Zernike* computations, since the following recursion relation holds: [2]

$$\begin{cases} B_0^- = 0, B_0^+ = 1, B_1^- = 1, B_1^+ = 1 \\ B_{2j}^- = \begin{bmatrix} 0 & 0 \\ B_{2j-1}^+ + B_{2j-1}^- & 0 \end{bmatrix} \\ B_{2j}^+ = \begin{bmatrix} \begin{bmatrix} B_{2j-1}^+ & 0 \end{bmatrix} + \begin{bmatrix} 0 & B_{2j-1}^- \end{bmatrix} \\ \begin{bmatrix} B_{2j-1}^{+(j-1)} & 0 \end{bmatrix} - \begin{bmatrix} 0 & B_{2j-1}^{-(j-1)} \end{bmatrix} \end{bmatrix} \\ B_{2j+1}^- = B_{2j}^+ + B_{2j}^- \\ B_{2j+1}^+ = B_{2j}^+ - \begin{bmatrix} 0 & B_{2j}^{-\{0 \le k \le j-1\}} \end{bmatrix} \end{cases}, \tag{11}$$

which is fully supported by the recursive nature of *Pascal*'s triangle, as are $w_n^{\{k\}}$ and $t_n^{\{l\}}$. By expanding the expressions $w_n^{\{k\}}, t_n^{\{l\}}$ and $B_n^{\mp}$ in Eq. 10 with their corresponding recursive forms, we obtain a block–wise recursive computation method for *Zernike* basis functions:

$$\begin{cases} z_0^- = 0, z_0^+ = 1, z_1^- = y, z_1^+ = x \\ z_{2j}^- = x \begin{bmatrix} 0 \\ \mathcal{O}_+(z_{2j-1}^-) \end{bmatrix} - y \begin{bmatrix} 0 \\ \mathcal{O}_-(z_{2j-1}^+) \end{bmatrix} - \begin{bmatrix} z_{2j-2}^- \\ 0 \end{bmatrix} \\ z_{2j}^+ = x \begin{bmatrix} 2z_{2j-1}^{+0} \\ \mathcal{O}_+(z_{2j-1}^+) \end{bmatrix} + y \begin{bmatrix} 2z_{2j-1}^{-0} \\ \mathcal{O}_-(z_{2j-1}^-) \end{bmatrix} - \begin{bmatrix} z_{2j-2}^+ \\ 0 \end{bmatrix} \\ z_{2j+1}^- = x\mathcal{O}_+(z_{2j}^-) - y\mathcal{O}_-(z_{2j}^+) - \begin{bmatrix} z_{2j-1}^- \\ 0 \end{bmatrix} \\ z_{2j+1}^+ = x\mathcal{O}_+(z_{2j}^+) + y\mathcal{O}_-(z_{2j}^-) - \begin{bmatrix} z_{2j-1}^+ \\ 0 \end{bmatrix} \end{cases}, \tag{12}$$

where $z_{2j}^{\mp} \equiv z_{2j}^{\mp\{0 \le l \le j\}} \equiv z_{n=2j}^{\mp\{m=2l\}}$ denotes a column vector of the *Zernike* basis function of order $n = 2j$ with $j+1$ elements at position (x, y), and analogously $z_{2j+1}^{\mp}$ for an odd order $2j+1$. The two vector operators $\mathcal{O}_+$ and $\mathcal{O}_-$ are defined as follows:

$$\begin{cases} \mathcal{O}_+(z_n^{\mp}) = \begin{bmatrix} z_n^{\mp\{<\}} \\ 0 \end{bmatrix} + z_n^{\mp} \\ \mathcal{O}_-(z_n^{\mp}) = \begin{bmatrix} z_n^{\mp\{<\}} \\ 0 \end{bmatrix} - z_n^{\mp} \\ z_n^{\mp\{<\}} = \begin{bmatrix} \cancel{z_n^{\mp 0}} & z_n^{\mp 1} & z_n^{\mp 2} & \dots & z_n^{\mp j} \end{bmatrix}^T \\ \qquad\quad = \begin{bmatrix} z_n^{\mp 1} & z_n^{\mp 2} & \dots & z_n^{\mp j} \end{bmatrix}^T \end{cases}, \tag{13}$$

for performing offset addition and offset subtraction on a given vector.

[2] The mathematical derivation in this section briefly proves the two previously mentioned cases of block–wise recurrence in *Zernike* computations without order restriction. Further details can be found in Appendix A.

3. New computation scheme

Zernike calculations in optical applications include, among others, the following topics: basis function calculation, surface (or wavefront) reconstruction, coefficients evaluation/transformation, derivative analysis, etc.

3.1. *Zernike* basis functions

Eq. 10 can be rewritten as a

$$\left[z_n^{\mp m}\right]_{1\times\frac{(n+1)(n+2)}{2}} = \left[X_n^i\right]_{1\times\frac{(n+1)(n+2)}{2}}\mathcal{T} \quad \xrightarrow{\text{as}} \quad \begin{bmatrix} z_0^0 \\ z_1^{-1} \\ z_1^1 \\ z_2^{-2} \\ z_2^0 \\ z_2^2 \\ \vdots \\ z_n^{-n} \\ z_n^{-n+2} \\ \vdots \\ z_n^n \end{bmatrix}^T = \begin{bmatrix} x^0y^0 \\ x^1y^0 \\ x^0y^1 \\ x^2y^0 \\ x^1y^1 \\ x^0y^2 \\ \vdots \\ x^ny^0 \\ x^{n-1}y^1 \\ \vdots \\ x^0y^n \end{bmatrix}^T \mathcal{T}, \tag{14}$$

where $[X_n^i = x^{n-i}y^i]$ denotes the set of basis functions of homogeneous bivariate (x, y) polynomials with orders from 0 to n, $[z_n^{\mp m}]$ denotes the set of basis functions of *Zernike* polynomials with the same orders, and $\mathcal{T}$ denotes the transformation matrix between these two sets of basis function. It is convenient to arrange xy basis functions in a naturally increasing order (Eq. 14), and the same applies to *Zernike* basis functions, i.e., with the notation of $\mp m$ [3].

There is the inverse transformation matrix $\mathcal{T}^{-1}$, with which we have

$$\left[X_n^i\right] = \left[z_n^{\mp m}\right]\mathcal{T}^{-1}. \tag{15}$$

Although $\mathcal{T}^{-1}$ can be obtained by matrix inversion of $\mathcal{T}$, it is recommended to construct it recursively on the basis of the two block–wise recurrences supported by *Pascal*'s triangle.[4]

The direct transformation method for calculating *Zernike* basis functions can be performed by first calculating the basis functions of homogeneous xy polynomials and then transforming them by multiplying the sparse matrix $\mathcal{T}$. Theoretically, the calculation of the xy–basis functions can be carried out efficiently and accurately according to

$$\begin{cases} \left[x^ny^0 \;\; x^{n-1}y^1 \;\; \cdots \;\; x^1y^{n-1}\right] = x\left[x^{n-1}y^0 \;\; x^{n-2}y^1 \;\; \cdots \;\; x^0y^{n-1}\right] \\ \left[x^ny^0 \;\; x^{n-1}y^1 \;\; \cdots \;\; x^1y^{n-1} \;\; x^0y^n\right] = \left[\left[x^ny^0 \;\; x^{n-1}y^1 \;\; \cdots \;\; x^1y^{n-1}\right] \;\; y^n\right] \end{cases}. \tag{16}$$

In addition to the direct block–wise transformation method, this work also presents a block–wise recursive method for calculating *Zernike* basis functions, which is described in Eq. 12. These two methods are not only mathematically equivalent but also practically interchangeable. The block–wise recursion can start either from any order n once the previous basis functions of order $n-1$ and $n-2$ are known, or from the 0^{th} order.

[3] This arrangement corresponds to the *OSA/ANSI* standard indices for *Zernike* polynomials [3].

[4] Table B.1 in Appendix B gives $\mathcal{T}$, Table B.2 its inverse $\mathcal{T}^{-1}$, each up to order 6.

3.2. Surface/wavefront reconstruction

Given a set of samples of a surface or wavefront, which can be either a set of 3D coordinate values, $\{(x_i, y_i, z_i)|i = 0, 1, 2, \cdots\}$ or a set of normal vectors $\left\{\left(x_i, y_i, (z_x = \partial z/\partial x, z_y = \partial z/\partial y)_i\right)|i = 0, 1, 2, \cdots\right\}$, a polynomial description of such a surface or wavefront can be reconstructed using a *Least-Squares* optimization:

$$C_{\{k\}} = \arg\min_{C_{\{k\}}} \sum_k w_{\{i\}} \left(Z_{\{i\}} - \left(X_{\{i\}\times\{k\}}\right) \times C_{\{k\}}\right)^2 \longrightarrow C_{\{k\}} = \left(X^T \left(w_{\{i\}} I\right) X\right)^{-1} \left(X^T \left(w_{\{i\}} I\right) Z_{\{i\}}\right), \tag{17}$$

where $C_{\{k\}}$ denotes a set of coefficients, $X_{\{i\}\times\{k\}}$ basis function values at (x, y) with $\{i\}$ rows for individual samples and $\{k\}$ columns for individual basis functions, which can be either homogeneous xy basis functions, or *Zernike* basis functions. $Z_{\{i\}}$ denotes the set of measurement values of surface height $z_{\{i\}}$ or of surface normal vectors $(z_x, z_y)_{\{i\}}$, while $w_{\{i\}}$ assigns weights to the individual samples: if $w_{\{i\}} \equiv 1$, all samples are equally important for the reconstruction.

3.3. Polynomial coefficients

A surface/wavefront can be expressed based on *Zernike* polynomials with basis functions $[z_n^{\mp m}]$ or based on homogenous xy polynomials with basis functions $[x^{n-i}y^i]$, as

$$z(x, y/\rho, \varphi) = \sum_{n=0}^{N} \left(\sum_{\mp m=-n,\text{step}2}^{+n} d_n^{\mp m} z_n^{\mp m}(\rho, \varphi/x, y) \right) \equiv \sum_{n=0}^{N} \left(\sum_{i=0}^{n} a_{ni} x^{n-i} y^i \right). \tag{18}$$

According to Eq. 17, $C_{\{k\}} = [d_n^{\mp m}]$ holds for the coefficient vector of *Zernike* polynomials, and $C_{\{k\}} = [a_{ni}]$ the coefficient vector for homogeneous xy polynomials. This leads to another group of transformations

$$\left[d_n^{\mp m}\right]_{\frac{(n+1)(n+2)}{2}\times 1} = \mathcal{T}^{-1} \left[a_{ni}\right]_{\frac{(n+1)(n+2)}{2}\times 1} \tag{19}$$

$$\left[a_{ni}\right]_{\frac{(n+1)(n+2)}{2}\times 1} = \mathcal{T} \left[d_n^{\mp m}\right]_{\frac{(n+1)(n+2)}{2}\times 1} \tag{20}$$

which allow us to freely transform not only the basis functions but also the corresponding coefficients between *Zernike* polynomials and homogeneous xy polynomials describing a surface or a wavefront.

3.4. Derivatives

Derivative analysis plays an important role in optical applications: first–order derivatives for vertex detection and ray tracing, second–order derivatives for surface curvature analysis, third–order derivatives for local apex detection (with local maximum/minimum curvature) and surface tilt evaluation, etc. Using the two sparse transformation matrices mentioned above, derivative calculations, as well as the evaluation of the optical properties of a surface/wavefront, can be flexibly performed based on its polynomial description, either *Zernike* or homogeneous xy polynomials. In particular, we have

the following equation for homogeneous xy polynomials:

$$\begin{cases} \dfrac{\partial z(x,y/\rho,\varphi)}{\partial x} = \displaystyle\sum_{n=0}^{N}\sum_{i=0}^{n} a_{ni}(n-i)x^{n-i-1}y^{i} \\ \dfrac{\partial z(x,y/\rho,\varphi)}{\partial y} = \displaystyle\sum_{n=0}^{N}\sum_{i=0}^{n} a_{ni}ix^{n-i}y^{i-1} \end{cases} \xrightarrow{n\geq i+1;i-1\geq 0} \begin{cases} \dfrac{\partial z}{\partial x} = \displaystyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} a_{(n+1)i}(n-i+1)x^{n-i}y^{i} \\ \dfrac{\partial z}{\partial y} = \displaystyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} a_{(n+1)(i+1)}(i+1)x^{n-i}y^{i} \end{cases}, \tag{21}$$

from which two first–order derivatives of a surface/wavefront described by homogeneous xy polynomials with order $0 \leq n \leq N$ correspond to two surfaces described by two homogeneous xy polynomials with order $0 \leq n \leq N-1$, according to the following equation

$$\begin{cases} \dfrac{\partial z(x,y/\rho,\varphi)}{\partial x} = \displaystyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} a_{ni}^{(x)}x^{n-i}y^{i} \\ \dfrac{\partial z(x,y/\rho,\varphi)}{\partial y} = \displaystyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} a_{ni}^{(y)}x^{n-i}y^{i} \end{cases}, \tag{22}$$

in which we newly introduce two sets of polynomial coefficients have: $a_{ni}^{(x)} = a_{(n+1)i}(n-i+1)$ for $\partial z/\partial x$ and $a_{ni}^{(y)} = a_{(n+1)(i+1)}(i+1)$ for $\partial z/\partial y$, which can be calculated directly from the coefficients of the original surface. Furthermore, second–order derivatives can be calculated from first–order derivatives, third–order derivatives from second–order derivatives, etc. Using Eq. 19, these also apply to *Zernike* polynomials in polar and *Cartesian* coordinate systems.

4. Discussion

The two block–wise recurrences discovered in Sec. 2 can be interpreted as a two–dimensional extension of the chronically observed **three–term–recurrence–relation** (*TTRR*) in orthogonal polynomials. The *Gram–Schmidt* orthogonalization process, which naturally introduces the *TTRR* (i.e., $P_n(x) = (A_n x + B_n)P_{n-1}x + C_n P_{n-2}(x)$), is widely used to construct orthogonal bases from homogeneous xy–polynomials. In particular, for *Zernike* polynomials, two coordinate–value–independent processes: the orthogonalization process and the coordinate system transformation, respectively lead to these two cases of block–wise recurrence.

Building upon the block–wise recursions revealed above and the recursive nature of *Pascal*'s triangle, this work presents both a block–wise direct transformation method (Eq. 10) and a block–wise recursive computation method (Eq. 12) for *Zernike* basis functions. Mathematically, these two methods are equivalent. In contrast to the available *Zernike* computation techniques: the component–wise computation of *Zernike* polynomials using a list of functions in the *Cartesian* coordinate system [8], the brute–force conversion of *Zernike* polynomials to *Cartesian* polynomials (homogeneous xy-polynomials) [9], and the component–wise recursive computation method [6][7], this work contributes to a block–wise understanding of *Zernike* polynomials.

Our special contributions include: 1. In addition to avoiding *cos*/*sin* calculations (as in [8]), we also avoid all calculations of repetitive factorials, divisions of large integers, and matrix inversions; 2. In contrast to the brute–force implementation of bidirectional conversion up to order 5 (with partial extension up to order 8) between *Zernike* basis functions and homogeneous xy basis functions [9],

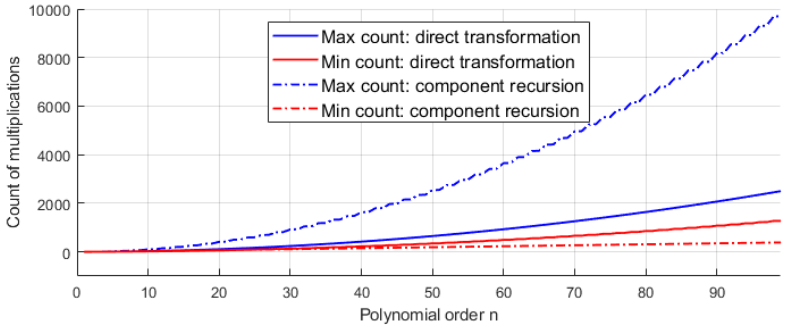
we separate the calculations irrelevant to the coordinate values from those relevant to the coordinate values, and offer either the recursive computation of the coordinate–irrelevant factors, or the recursive computation of the complete *Zernike* basis functions. Such recursions are based on block–wise recurrences for coordinate–irrelevant calculations and their close relationship to a *Pascal* triangle. We exploit these new insights for *Zernike*–related computations, thereby eliminating the tediousness/complexity of high–order *Zernike* components and thus achieving greater flexibility for optical applications without restriction of order. Furthermore, this work discovers a direct, but previously unnoticed, block–wise connection between *Zernike* polynomials in *Cartesian* form and their recursive computations (Eqs. 10, 11 and 12).

Computations	Count of Multiplication (×)	Count of addition (+)
$\left[x^{2j-2l-1}y^{2l+1}\right]_{j\geq 1,0\leq l\leq j}$	$\frac{j(j+1)}{2}$	0
$\left[x^{2j-2l}y^{2l}\right]_{j\geq 0,0\leq l\leq j}$	$\frac{(j+1)(j+2)}{2}-2$	0
$\left[x^{2j-2l}y^{2l+1}\right]_{j\geq 0,0\leq l\leq j}$	$\frac{(j+1)(j+2)}{2}-1$	0
$\left[x^{2j-2l+1}y^{2l}\right]_{j\geq 0,0\leq l\leq j}$	$\frac{(j+1)(j+2)}{2}-1$	0
z_{2j}^{-2l}	$\frac{(j+l)(j-l+1)}{2}$	$\frac{(j+l)(j-l+1)}{2}-1$
z_{2j}^{+2l}	$\frac{(j+l+1)(j-l)+2(j+1)}{2}$	$\frac{(j+l+1)(j-l)+2(j+1)}{2}-1$
$z_{2j+1}^{-(2l+1)}$	$\frac{(j+l+1)(j-l)+2(j+1)}{2}$	$\frac{(j+l+1)(j-l)+2(j+1)}{2}-1$
$z_{2j+1}^{+(2l+1)}$	$\frac{(j+l+1)(j-l)+2(j+1)}{2}$	$\frac{(j+l+1)(j-l)+2(j+1)}{2}-1$

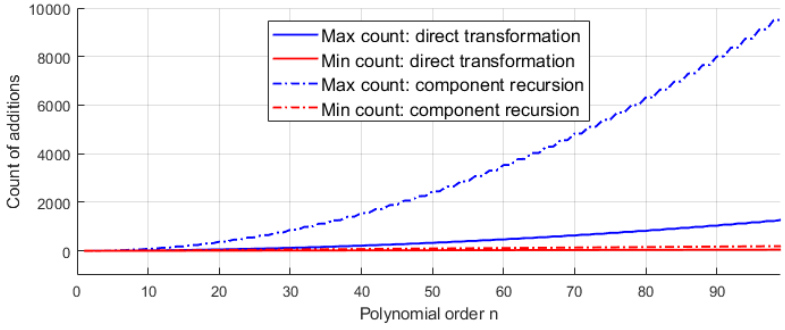
Table 3. Counts of arithmetic instructions for a *Zernike* component $z_n^{\pm m}$, where *sin* and *cos* components of even and odd orders are considered separately.

4.1. Computational performance: a preliminary evaluation

The direct block–wise transformation method described in Eq. 10 and Eq. 14 for calculating *Zernike* basis functions can be interpreted as an upgraded version of the xy–form–LUT method mentioned in section 1. Instead of explicitly defining a LUT of independent functions or equations for individual *Zernike* components in their xy–polynomial form, one or four recursively extendable transformation matrixes and a recursively extendable list of basis functions of *Cartesian* xy–polynomials (e.g., based on Eq. 16) are combined by matrix multiplications, resulting in the required *Zernike* basis functions. Theoretically, this upgrade does not change the essential computational complexity for a single *Zernike* component, which can be measured as the number of two arithmetic CPU instructions: multiplication and addition. For a required *Zernike* component with polynomial order $n = 2j$ or $n = 2j + 1$ and azimuth order $m = \pm 2l$ or $m = \pm(2l + 1)$, the numbers of essential arithmetic instructions are listed in Table 3, where the construction of $\mathcal{T}$ is not counted because it is coordinate–independent and can therefore be calculated in advance as a factor LUT for the *Zernike* calculation. In short, this work improves the traditional *Zernike* computation scheme based on the xy–polynomial form with two significant advantages: 1) It is no longer necessary to repeatedly calculate the xy–basis functions in each calculation function/equation of individual *Zernike* components in *Cartesian* form; 2) There is no longer order restriction caused by the limited list of *Zernike* components with their *Cartesian* form given in the literature.



(a)

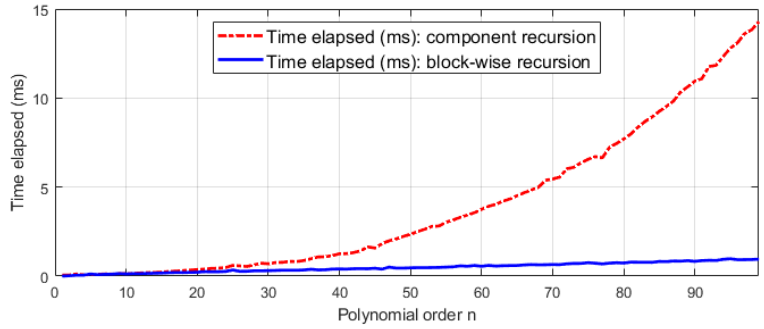


(b)

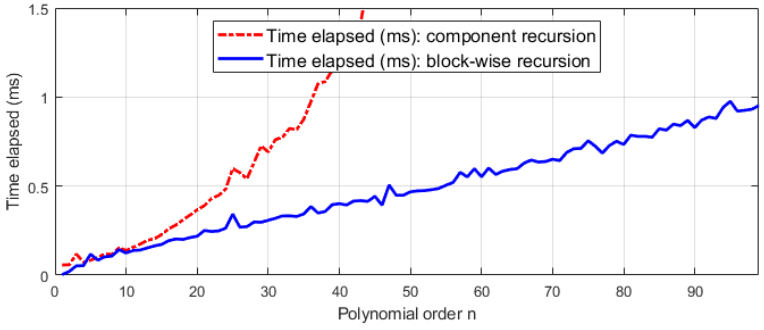
Figure 1. Comparison of computational complexity (I): (a) number of multiplications; (b) number of additions. Four curves in (a) show the maximum and minimum multiplication requirements for the component–wise recursion method and the direct transformation method for a required *Zernike* component at one (x, y) position; the same applies to four curves in (b) for addition operations.

It is noticeable that the computational complexity of this direct transformation method is not the same for individual *Zernike* components with the same polynomial order n, but depends on their azimuth order $\pm m$, where z_n^{-n} requires minimum and z_n^0 (or z_n^1) maximum calculations. In Fig. 1(a), two solid curves represent the minimum and maximum number of all multiplications, including the computation of the xy basis functions and the matrix multiplication $[X] \times \mathcal{T}$ (Eq. 14), for polynomial orders from 0 to 99. These two curves are compared with two additional dashed–dotted curves representing the corresponding performance measurements of the *Zernike* computation method based on component–wise recurrence, where the application scenario is set to compute a single *Zernike* component with given n and m for a coordinate position (x, y). The computational performance of component–wise recursion is achieved by manually programming the formulas 32–38 in [7] (*Andersen*, 2018), extending the three–term–recurrence to a five–term–recurrence for full *Zernike* polynomials. Similarly, Fig. 1(b) shows the comparison of computational complexity based on the number of addition operations. These two comparisons demonstrate that the *Zernike* computation method based on direct block–wise transformation has high computational performance.

The above performance evaluation makes sense in theory, but calculating a single component is not so common in practice. In contrast, preparing all components from order 0 to order n is necessary for surface/wavefront reconstruction and analysis. The implementation of the corresponding methods significantly influences the final performance assessment. Two further aspects must be considered: 1) the reuse of lower–order results; 2) the use of modern computer hardware such as *SIMD*



(a)



(b)

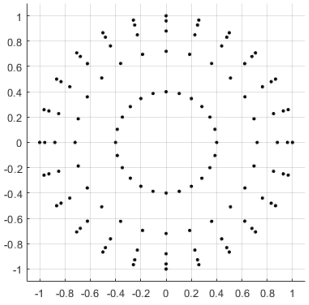
Figure 2. Comparison of computational complexity (II): Time elapsed for generating all *Zernike* components up to order 99 using component–wise recursion (red) compared to block–wise recursion (blue): (a) the comparison; (b) further details on the elapsed time of block–wise recursion (with y–axis $\leq 1.5ms$). The comparison was performed for one (x, y) position, and the elapsed time is measured in millisecond $(10^{-3}s)$.

(Single–Instruction–Multiple–Data) technique and *Cache*–hierarchy. Block–wise recursion offers a direct route to modern, hardware–compatible implementations. Another computational performance comparison was performed between the component–wise recursive calculation method mentioned above and the block–wise recursive calculation method. Fig. 2 shows the advantage of the latter method. Both methods were implemented and tested in the same MATLAB environment (*DELL* laptop, Intel(R) Core(TM) i5-1245U, 16GB RAM, 128MB Intel(R) UHD Graphics, Windows-11 OS, MATLAB-R2020b 64-bit) without any special coding optimization. Although this is only a preliminary assessment, the block–wise recursive method shows better computational performance, scaling linearly with the polynomial order while the component–wise recursive method appears to be exponentially relevant. This advantage stems from the inherent compatibility between the block–wise recursive method and modern hardware technology in civilian commercial computers.

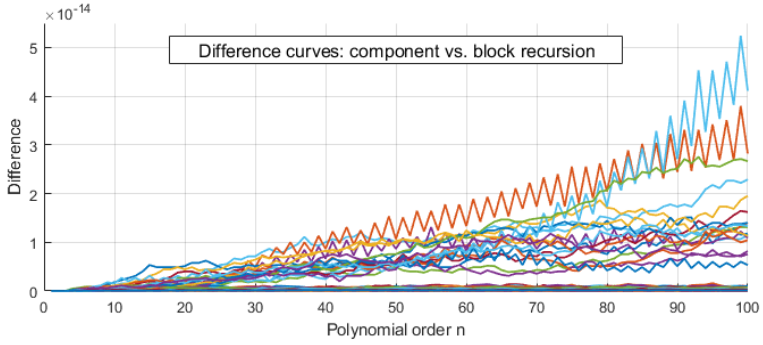
4.2. Computational accuracy: a preliminary evaluation

In addition to the computational performance test described above, the computational accuracy was also compared using 120 spatial positions taken from 5 radius positions times 24 azimuth positions (see Fig. 3(a)). For each spatial position, all *Zernike* components from order 0 to order 99 were calculated using both the component–wise recursive method mentioned above ([7]) and the block–

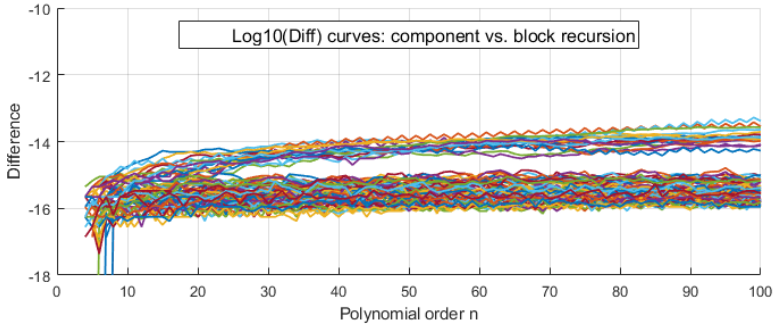
wise recursive method based on *Pascal*'s triangle (Eq. 12). For each order, the maximum absolute difference between the results of both methods was recorded; thus, a difference curve along polynomial orders was created for each position. All 120 curves were recorded during the test. They are shown in Fig. 3 in two versions: the raw difference curves (b) and their logarithmic ($\log 10$) version (c).



(a)



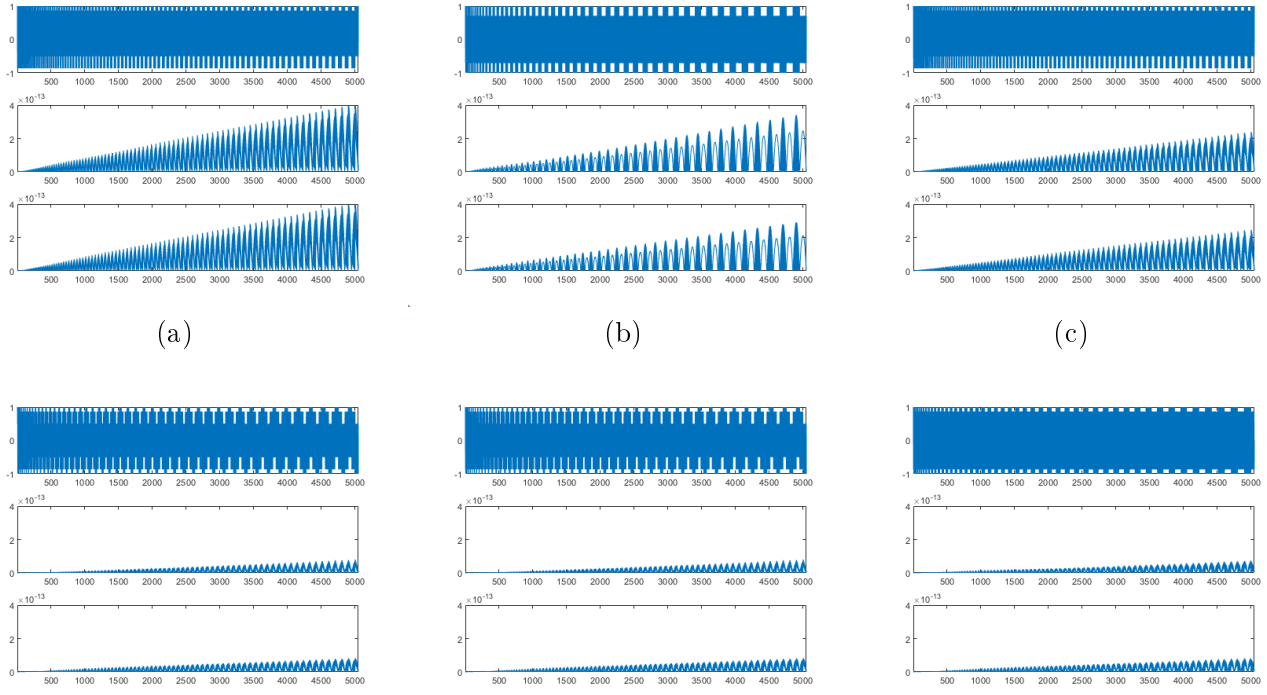
(b)



(c)

Figure 3. Accuracy assessment of the block–wise recursion compared to the component–wise recursion (*Andersen*'s method [7]) for calculating *Zernike* basis functions: (a) 120 test positions from a circular region with five radii $(1.00, 0.96, 0.88, 0.72, 0.40)$ and 24 uniformly distributed azimuth angles; (b) 120 difference curves (color–coded) with respect to the polynomial orders, each curve representing the maximum absolute differences between two sets of *Zernike* components along the orders from 0 to 99 for a position in (a); (c) the logarithmic representation ($\log 10$) of the corresponding curves in (b), of which 17 curves are distinguished from the others by their difference value of over 3×10^{-15} and whose corresponding positions all lie on the unit circle.

The comparison results show an absolute difference of less than 6×10^{-14}, with the maximum absolute difference exceeding 3×10^{-15} at 17 positions on the unit circle from order 50 onwards. This preliminary result confirmed our expectation that the block–wise recursive method can be interpreted as equivalent to the component–wise recursive method, although the former is proposed on the basis of one *Pascal* triangle and the latter was developed from the *TTRR* relationship in orthogonal polynomials.



(a) (b) (c)

(d) (e) (f)

Figure 4. Accuracy comparisons (a)–(f) at six positions on the unit circle: $(x, y) \approx$ (-0.5000000,-0.8660254), (-0.7071068,-0.7071068), (-0.5000000, 0.8660254), (0.5000000, 0.8660254). (0.5000000,-0.8660254), and (-0.8660254,-0.5000000). Each comparison includes three curves: the one above for 5050 *Zernike* components of orders 0 to 99, calculated using the direct transformation method, as the comparison reference; the middle one for the absolute difference of the component–wise recursive method to the reference; and the curve below for the block–wise recursive method. All difference curves are shown with the same value scale: $\leq 4 \times 10^{-13}$.

Naturally, the question arises as to the above 17 positions on the unit circle: which method is closer to the truth? Recursive methods (component–wise and block–wise recursion) were further compared with the direct transformation method (Eq. 10). Unlike that the two recursive methods, which were fully implemented in MATLAB with double precision, the direct transformation method requires additional support from an external library for "arbitrary–precision–arithmetic" due to the rapidly increasing integer values (i.e., binomial factors) for polynomials of order higher than 50. The comparison results for the six positions with the largest deviations, all located on the unit circle, are shown in Fig. 4. It is evident that for both recursive methods, all deviations are smaller than approximately 4×10^{-13} compared to the direct transformation method.

It should be noted that this work does not constitute a definitive assessment of the accuracy of one method compared to others, but rather a preliminary evaluation to support the conceptual investigation of the relationship between *Zernike* polynomials and *Pascal*'s triangle. Further evaluations with more detailed application scenarios are to be expected, as the difference curves reveal some order–dependent structural information which, although very small, is not random noise.

4.3. Memory requirement

Memory requirements are often a critical factor in development due to detailed application scenarios. This work offers high flexibility for practical system/algorithm development: Block–wise recursive computation for generating basis functions requires no additional memory; similarly, virtually no additional memory is needed for surface/wavefront analysis, which uses transformation matrices for coefficient operations and derivative computations, since all transformation matrices can be rapidly computed by block–wise recursive algorithms, e.g., Eq. 11 for the forward coordinate system transformation in *Zernike* computations.

5. Conclusion

In this work, the computation of *Zernike* basis functions was divided into two parts: the part irrelevant to the coordinate values and the part relevant to the coordinate values. The first part comprises the computation of the coordinate transformation along with an orthogonalization process (based on homogeneous xy polynomials). This leads to two cases of block–wise recurrence in the coordinate–independent *Zernike* computation. The use of these two block–wise recurrences, supported by *Pascal* triangle, enables a block–wise direct transformation method and a block–wise recursive computation method for *Zernike* polynomials. The latter is more suitable for computing basis functions, while the former is better suited for operations and analyses of surfaces using their polynomial coefficients. Theoretically, this significantly improves computational speed, reduces memory requirements, and increases application flexibility without compromising accuracy. Optical applications can benefit from these new insights.

This work provides a solid starting point for further theoretical and application–specific studies, such as a comprehensive evaluation of computational complexity, a dedicated algorithmic accuracy/stability investigation, and application–oriented algorithm design, *Zernike* analysis with non–circular pupil, fast higher–order derivatives of a surface described by *Zernike* polynomials, etc. It is also valuable for discovering similar cases of block–wise recurrence in other orthogonal polynomials constructed using the *Gram-Schmidt* process.

Funding

This research received no external funding.

Conflict of Interest

The authors declare no conflicts of interest in regards to this article.

Data Availability Statement

The underlying data can be provided by the authors upon reasonable request.

Author Contribution Statement

Wei–Jun CHEN contributes fully to this article.

References

1. Lakshminarayanan V., Fleck A., Zernike polynomials: a guide, Journal of Modern Optics, 58(7), 545–561 (2011), https://doi.org/10.1080/09500340.2011.633763.
2. Schwiegerling J., Description of Zernike Polynomials (2016), https://wp.optics.arizona.edu/visualopticslab/wp-content/uploads/sites/52/2016/08/Zernike-Notes-15Jan2016.pdf.
3. Niu K., Tian C., Zernike polynomials and their applications, Journal of Optics, 24(12), 1–54 (2022), http://doi.org/10.1088/2040-8986/ac9e08.
4. Born M., Wolf E., Bhatia A. B., Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th ed.), Cambridge, UK: Cambridge University Press, ISBN 978-0-521-64222-4, 523 (1999), https://books.google.de/books?id=nUHGpfNsGyUC&q=Zernike&redir_esc=y#v=snippet&q=Zernike&f=false.
5. Zernike F., Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrast-methode, Physica, 1(7–12), 689–704 (1934), https://doi.org/10.1016/S0031-8914(34)80259-5
6. Shakibaei B. H., Paramesran R., Recursive formula to compute Zernike radial polynomials, Optical Letters, 38(14), 2487–2489 (2013), http://dx.doi.org/10.1364/OL.38.002487.
7. Andersen T. B., Efficient and robust recurrence relations for the Zernike circle polynomials and their derivatives in Cartesian coordinates, Optics Express, 26(15), 18878–18896 (2018), http://doi.org/10.1364/OE.26.018878.
8. van Brug H. H., Efficient Cartesian representation of Zernike polynomials in computer memory, Proceedings SPIE: Fifth International Topical Meeting on Education and Training in Optics, 382–392 (1997), https://doi.org/10.1117/12.294412.
9. Mathar R. J., Zernike basis to Cartesian Transformations, Serbian Astronomical Journal, 179, 107–120 (2009), https://doi.org/10.2298/SAJ0979107M.

Appendix

Appendix A: Mathematical proofs

A.1. *Pascal* triangle supported block–wise recurrence in *Zernike* polynomials

The *Zernike* radial polynomial has its binomial form (Eq. 8). Defining $n = 2j, m = 2l$ for even orders and $n = 2j+1, m = 2l+1$ for odd orders, and defining two intermediate variables $J_k = j - k$

and $J_{2k}=2J_k$, we obtain

$$\begin{cases} R_{2j}^{2l}=\sum\limits_{k=0}^{j-l}(-1)^k\begin{pmatrix}J_{2k}+k\\J_{2k}\end{pmatrix}\begin{pmatrix}J_{2k}\\J_k+l\end{pmatrix}(\rho^2)^{J_k} \\ R_{2j+1}^{2l+1}=\rho\sum\limits_{k=0}^{j-l}(-1)^k\left(\begin{pmatrix}J_{2k}+k\\J_{2k}\end{pmatrix}+\begin{pmatrix}J_{2k}+k\\J_{2k}+1\end{pmatrix}\right)\left(\begin{pmatrix}J_{2k}\\J_k+l\end{pmatrix}+\begin{pmatrix}J_{2k}\\J_k+l+1\end{pmatrix}\right)(\rho^2)^{J_k}\end{cases}. \quad (A.1)$$

Further defining $w_k^e=\binom{J_{2k}+k}{J_{2k}}$, $w_k^o=w_k^e+\binom{J_{2k}+k}{J_{2k}+1}$, $t_{lk}^e=\binom{J_{2k}}{J_k+l}$, and $t_{lk}^o=t_{lk}^e+\binom{J_{2k}}{J_k+l+1}$, we obtain

$$\begin{cases} R_{2j}^{2l}=\sum\limits_{k=0}^{j-l}(-1)^k w_k^e t_{lk}^e(\rho^2)^{J_k} \\ R_{2j+1}^{2l+1}=\rho\sum\limits_{k=0}^{j-l}(-1)^k w_k^o t_{lk}^o(\rho^2)^{J_k}\end{cases}. \quad (A.2)$$

In particular, we have $w_k^e=\binom{J_{2k}+k}{J_{2k}}=\binom{2j-k}{k}$, for which we obtain a vector by continuously increasing k by 1 from 0 to $k=j-l$, provided $w_{\{k\}}^e=\left[\binom{2j}{0}\ \binom{2j-1}{1}\ \cdots\ \binom{j+l}{j-l}\right]$ as a row vector, where $2j=n$ for even orders and $l\le j$; we also have $w_k^o=w_k^e+\binom{J_{2k}+k}{J_{2k}+1}=\binom{2j-k}{k}+\binom{2j-k}{k-1}=\binom{2j+1-k}{k}$, provided $w_{\{k\}}^o=\left[\binom{2j+1}{0}\ \binom{2j}{1}\ \cdots\ \binom{j+1+l}{j-l}\right]$, where $2j+1=n$ for odd orders and $l\le j$. Either $w_{\{k\}}^e$ or $w_{\{k\}}^o$ denotes a corresponding anti–diagonal vector in a left–aligned n–row *Pascal* triangle (e.g., Table 2 for $n\le 6$). For one even (or odd) order, $w_{\{k\}}^e$ (or $w_{\{k\}}^o$) depends on a single variable k and is independent of l or m.

Meanwhile, for an even order $n=2j$, we have $t_{lk}^e=\binom{J_{2k}}{J_k+l}=\binom{2(j-k)}{(j-k)+l}$, where $t_{lk}^e\big|_{k=0}=\binom{2j}{j+l}$ subject to $k=0$; continuously increasing k by 1 till $k=j-l$, we obtain a row vector $t_{l\{k\}}^e=\left[\binom{2j}{j+l}\ \binom{2j-2}{j-1+l}\ \binom{2j-4}{j-2+l}\ \cdots\ \binom{2l}{2l}\right]$ with respect to a particular l, and obtain an anti–upper–triangular matrix for all the $0\le l\le j$, as

$$T_{n=2j}\left(t_{lk}^e\right)=t_{\{l\}\{k\}}^e=\begin{bmatrix}\binom{2j}{j} & \binom{2j-2}{j-1} & \binom{2j-4}{j-2} & \cdots & \binom{2}{1} & \binom{0}{0}\\ \binom{2j}{j+1} & \binom{2j-2}{j} & \binom{2j-4}{j-1} & \cdots & \binom{2}{2} & \\ \binom{2j}{j+2} & \binom{2j-2}{j+1} & \binom{2j-4}{j} & \cdots & & \\ \cdots & \cdots & \ddots & & & \\ \binom{2j}{2j-2} & \binom{2j-2}{2j-2} & & & & \\ \binom{2j}{2j} & & & & & \end{bmatrix}_{(j+1)\times(j+1)}. \quad (A.3)$$

Also for an odd order $n=2j+1$, we have $t_{lk}^o=t_{lk}^e+\binom{J_{2k}}{J_k+l+1}=\binom{2(j-k)}{(j-k)+l}+\binom{2(j-k)}{(j-k)+l+1}=\binom{2(j-k)+1}{\binom{(j-k)+1}{}+l}$, and an anti–upper–triangular matrix as follows:

$$T_{n=2j+1}\left(t_{lk}^o\right)=t_{\{l\}\{k\}}^o\begin{bmatrix}\binom{2j+1}{j+1} & \binom{2j-1}{j} & \binom{2j-3}{j-1} & \cdots & \binom{3}{2} & \binom{1}{1}\\ \binom{2j+1}{j+2} & \binom{2j-1}{j+1} & \binom{2j-3}{j} & \cdots & \binom{3}{3} & \\ \binom{2j+1}{j+3} & \binom{2j-1}{j+2} & \binom{2j-3}{j+1} & \cdots & & \\ \cdots & \cdots & \ddots & & & \\ \binom{2j+1}{2j} & \binom{2j-1}{2j-1} & & & & \\ \binom{2j+1}{2j+1} & & & & & \end{bmatrix}_{(j+1)\times(j+1)}. \quad (A.4)$$

The results of mathematical derivation starting from binomial equivalence agree with the intuitive visual observation in the previous section 2.1:

$$\begin{cases} (-1)^k w_k^e \equiv w_{n=2j}^k \\ (-1)^k w_k^o \equiv w_{n=2j+1}^k \\ t_{lk}^e\Big|_{k=0} \equiv t_{n=2j}^{m=2l} \\ t_{lk}^o\Big|_{k=0} \equiv t_{n=2j+1}^{m=2l+1} \\ T_n\left(t_{lk}^{o/e}/0\right) \equiv T_n = \left[t_n^{\{m\}} \begin{bmatrix} T_{n-2} \\ 0 \end{bmatrix}\right] \end{cases}, \tag{A.5}$$

where $T_n\left(t_{lk}^{o/e}/0\right)$ denotes a variation of $T_n\left(t_{lk}^{o/e}\right)$ by filling the elements below the anti–diagonal elements with zeros. Furthermore, by adopting

$$\begin{cases} \mathrm{diag}\left(w_{2j}^{\{k\}}\right) = \begin{bmatrix} (-1)^0 w_{k=0}^e & & & \\ & (-1)^1 w_{k=1}^e & & \\ & & \ddots & \\ & & & (-1)^{j-l} w_{k=j}^e \end{bmatrix}_{(j+1)\times(j+1)} \\ \mathrm{diag}\left(w_{2j+1}^{\{k\}}\right) = \begin{bmatrix} (-1)^0 w_{k=0}^o & & & \\ & (-1)^1 w_{k=1}^o & & \\ & & \ddots & \\ & & & (-1)^{j-l} w_{k=j}^o \end{bmatrix}_{(j+1)\times(j+1)} \end{cases}, \tag{A.6}$$

Eq. A.2 can be re–written as

$$\begin{cases} R_{2j}^{\{2l\}} = T_{2j} \times \mathrm{diag}\left(w_{2j}^{\{k\}}\right) \times \left(\left[(\rho^2)^j \ (\rho^2)^{j-1} \ \cdots \ (\rho^2) \ 1\right]^T\right)_{(j+1)\times 1} \\ R_{2j+1}^{\{2l+1\}} = T_{2j+1} \times \mathrm{diag}\left(w_{2j+1}^{\{k\}}\right) \times \left(\left[(\rho^2)^j \ (\rho^2)^{j-1} \ \cdots \ (\rho^2) \ \ 1\right]^T\right)_{(j+1)\times 1} \times \rho \end{cases}, \tag{A.7}$$

where the radial coordinate–dependent vector $\left[(\rho^2)^j \ (\rho^2)^{j-1} \ \cdots \ (\rho^2) \ 1\right]^T$ contains all possible $(\rho^2)^{J_k}$ in Eq. A.2, $\forall k \in [0, j-l], \forall l \in [0, j]$. Essentially Eq. A.7 is equivalent to Eq. 5, and Eqs. A.3 and A.4 are equivalent to Eq. 7, without an order limit ($\forall n \geq 0$).

It is noticed that the close relationship between the radial polynomials and *Pascal* triangle can be straightforwardly obtained even without the intermediate variables J_k and J_{2k}: The ordered set of binomial factors $\left\{\binom{n-k}{k}\middle| k = 0, 1, ...(n-m)/2\right\}$ corresponds to an anti–diagonal vector of a left–aligned *Pascal* triangle, where k increases while $n-k$ decreases, each in steps of 1, as shown in Table 2 for $n \leq 6$; **the binomial factor $\binom{n-2k}{(n-m)/2-k}$ can be rewritten as $\binom{n-2k}{(n-2k)/2-m/2}$, which is equivalent to $\binom{n-2k}{(n-2k)/2+m/2}$.** If k varies as the column index and $l = m/2$ for even orders or $l = (m-1)/2$ for odd orders varies as the row index, the ordered factor set $\left\{\binom{n-2k}{(n-2k)/2+l}\middle| l = 0, 1, j; k = 0, 1, ...j-l\right\}$ forms a 2D matrix. For a certain k, the column vector $\binom{n-2k}{(n-2k)/2+\{l\}}$ corresponds to the right half of the $(n-2k)^{th}$ row in *Pascal* triangle (Table 1 for $n \leq 6$). Nevertheless, using the intermediate variables J_k and J_{2k}, additional block–wise relationships such as $w_k^o = w_k^e + \binom{J_{2k}+k}{J_{2k}+1}$ and $t_{lk}^o = t_{lk}^e + \binom{J_{2k}}{J_k+l+1}$ can be obtained, which, although not as intuitive, are useful for further algorithm development.

A.2. *Zernike* radial polynomials in *Cartesian*: the second block–wise recurrence

Considering that $\rho^2 = x^2 + y^2$, its power value $(\rho^2)^j$ can be expressed as the multiplication of two vectors:

$$\left(\rho^2\right)^j = \sum_{q=0}^{j} b_j^q x^{2(j-q)} y^{2q} = b_j^{\{q\}} \times X_j^{\{q\}} \tag{A.8}$$

where $b_j^{\{q\}}$ denotes a row vector containing all binomial factors of $\left\{b_j^q = \binom{j}{q} \middle| q = 0, 1, 2, ...j\right\}$, corresponding to the j^{th} row in an n–row *Pascal* triangle, and $X_j^{\{q\}}$ denotes a column vector of the sequentially listed j^{th} basis functions of homogeneous bivariate polynomials for (x^2, y^2), as $\left[x^{2j}y^0 \;\; x^{2(j-1)}y^2 \;\; \cdots x^0 y^{2j}\right]^T$. Further, the radial coordinate–dependent vector in Eq. A.7 can be represented in matrix form as follows:

$$\begin{bmatrix} (\rho^2)^j \\ (\rho^2)^{j-1} \\ \vdots \\ (\rho^2) \\ 1 \end{bmatrix}_{(j+1)\times 1} = \begin{bmatrix} b_j^{\{q\}} & 0 & \cdots & 0 & 0 \\ 0 & b_{j-1}^{\{q\}} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_1^{\{q\}} & 0 \\ 0 & 0 & \cdots & 0 & b_0^{\{q\}} \end{bmatrix}_{(j+1)\times \frac{(j+1)(j+2)}{2}} \times \begin{bmatrix} X_j^{\{q\}} \\ X_{j-1}^{\{q\}} \\ \vdots \\ X_1^{\{q\}} \\ X_0^{\{q\}} \end{bmatrix}_{\frac{(j+1)(j+2)}{2}\times 1}, \tag{A.9}$$

which can be iteratively represented in submatrix form as follows:

$$\begin{bmatrix} (\rho^2)^j \\ (\rho^2)^{j-1} \\ \vdots \\ (\rho^2) \\ 1 \end{bmatrix}_{(j+1)\times 1} = \mathcal{B}_j \times \mathcal{X}_j = \begin{bmatrix} b_j^{\{q\}} & 0 \\ 0 & \mathcal{B}_{j-1} \end{bmatrix}_{(j+1)\times \frac{(j+1)(j+2)}{2}} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix}_{\frac{(j+1)(j+2)}{2}\times 1} \tag{A.10}$$

where $\mathcal{B}_j$ denotes the complete matrix with $(j+1)$ rows and $\frac{(j+1)(j+2)}{2}$ columns in Eq. A.9, $\mathcal{B}_{j-1}$ is its right/below submatrix with j rows and $\frac{(j)(j+1)}{2}$ columns, and the same applies to $\mathcal{X}_j$ and its sub–vector $\mathcal{X}_{j-1}$ for the right column vector containing all $X_j^{\{q\}}$s. Subsequentially Eq. A.7 can be written as

$$\begin{cases} R_{2j}^{\{2l\}} = T_{2j} \times \mathrm{diag}\left(w_{2j}^{\{k\}}\right) \times \mathcal{B}_j \times \mathcal{X}_j \\ R_{2j+1}^{\{2l+1\}} = T_{2j+1} \times \mathrm{diag}\left(w_{2j+1}^{\{k\}}\right) \times \mathcal{B}_j \times \mathcal{X}_j \times \rho \end{cases} \tag{A.11}$$

inside which we have two cases of coordinate–irrelevant block–wise recurrence, as follows:

$$\begin{cases} R_{2j}^{\{2l\}} = \begin{bmatrix} t_{2j}^{\{2l\}} & \begin{bmatrix} T_{2j-2} \\ 0 \end{bmatrix} \end{bmatrix} \times \mathrm{diag}\left(w_{2j}^{\{k\}}\right) \times \begin{bmatrix} b_j^{\{q\}} & 0 \\ 0 & \mathcal{B}_{j-1} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} \\ R_{2j+1}^{\{2l+1\}} = \begin{bmatrix} t_{2j+1}^{\{2l+1\}} & \begin{bmatrix} T_{2j-1} \\ 0 \end{bmatrix} \end{bmatrix} \times \mathrm{diag}\left(w_{2j+1}^{\{k\}}\right) \times \begin{bmatrix} b_j^{\{q\}} & 0 \\ 0 & \mathcal{B}_{j-1} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} \times \rho \end{cases}. \tag{A.12}$$

For weight factors, $w_n^0 = 1$ always applies, and subsequentially $w_n^{\{k\}} = \left[1 \; w_n^{\{k>0\}}\right]$, where $w_n^{\{k>0\}}$ denotes the sub–vector of $w_n^{\{k\}}$ without its first element. Eq. A.12 can be written as follows:

$$\begin{cases} R_{2j}^{\{2l\}} = \begin{bmatrix} t_{2j}^{\{2l\}} & \begin{bmatrix} T_{2j-2} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}\left(w_{2j}^{\{k>0\}}\right) \end{bmatrix} \times \begin{bmatrix} b_j^{\{q\}} & 0 \\ 0 & \mathcal{B}_{j-1} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} \\ R_{2j+1}^{\{2l+1\}} = \begin{bmatrix} t_{2j+1}^{\{2l+1\}} & \begin{bmatrix} T_{2j-1} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}\left(w_{2j+1}^{\{k>0\}}\right) \end{bmatrix} \times \begin{bmatrix} b_j^{\{q\}} & 0 \\ 0 & \mathcal{B}_{j-1} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} \times \rho \end{cases} \tag{A.13}$$

where all coordinate–irrelevant calculations, i.e., calculations of and between t, T, w, b and $\mathcal{B}$, are fully supported by one n–row *Pascal* triangle.

A.3. Complete *Zernike* by one *Pascal* triangle

If we further define $J_{kl} = j - k - l$ and combine *de Moivre*'s formula with Eq. A.2, we obtain

$$\begin{cases} z_{2j}^{+2l} = \sum_{k=0}^{j-l}(-1)^k w_k^e t_{lk}^e \left(\left(\sum_{q=0}^{J_{kl}} b_{J_{kl}}^q (x^2)^q (y^2)^{J_{kl}-q} \right) \times \left(\sum_{p=0}^{l} (-1)^p b_{2l}^{2p} (x^2)^{l-p} (y^2)^p \right) \right) \\ z_{2j}^{-2l} = x^{-1} y \sum_{k=0}^{j-l}(-1)^k w_k^e t_{lk}^e \left(\left(\sum_{q=0}^{J_{kl}} b_{J_{kl}}^q (x^2)^q (y^2)^{J_{kl}-q} \right) \times \left(\sum_{p=0}^{l-1} (-1)^p b_{2l}^{2p+1} (x^2)^{l-p} (y^2)^p \right) \right) \\ z_{2j+1}^{+(2l+1)} = x \sum_{k=0}^{j-l}(-1)^k w_k^o t_{lk}^o \left(\left(\sum_{q=0}^{J_{kl}} b_{J_{kl}}^q (x^2)^q (y^2)^{J_{kl}-q} \right) \times \left(\sum_{p=0}^{l} (-1)^p b_{2l+1}^{2p} (x^2)^{l-p} (y^2)^p \right) \right) \\ z_{2j+1}^{-(2l+1)} = y \sum_{k=0}^{j-l}(-1)^k w_k^o t_{lk}^o \left(\left(\sum_{q=0}^{J_{kl}} b_{J_{kl}}^q (x^2)^q (y^2)^{J_{kl}-q} \right) \times \left(\sum_{p=0}^{l} (-1)^p b_{2l+1}^{2p+1} (x^2)^{l-p} (y^2)^p \right) \right) \end{cases}, \tag{A.14}$$

which is equivalent to the following equation in matrix form:

$$\begin{cases} z_{2j}^{-\{2l\}} = \begin{bmatrix} t_{2j}^{\{2l\}} & \begin{bmatrix} T_{2j-2} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}(w_{2j}^{\{k>0\}}) \end{bmatrix} \times \begin{bmatrix} [b_{2l}^{\{2p+1\}} \otimes b_{j-l}^{\{q\}}]^{\{l\}\{k\}} & 0 \\ 0 & \mathcal{B}_{j-1}^{e-} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} x^{-1} y \\ z_{2j}^{+\{2l\}} = \begin{bmatrix} t_{2j}^{\{2l\}} & \begin{bmatrix} T_{2j-2} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}(w_{2j}^{\{k>0\}}) \end{bmatrix} \times \begin{bmatrix} [b_{2l}^{\{2p\}} \otimes b_{j-l}^{\{q\}}]^{\{l\}\{k\}} & 0 \\ 0 & \mathcal{B}_{j-1}^{e+} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} \\ z_{2j+1}^{-\{2l+1\}} = \begin{bmatrix} t_{2j+1}^{\{2l+1\}} & \begin{bmatrix} T_{2j-1} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}(w_{2j+1}^{\{k>0\}}) \end{bmatrix} \times \begin{bmatrix} [b_{2l+1}^{\{2p+1\}} \otimes b_{j-l}^{\{q\}}]^{\{l\}\{k\}} & 0 \\ 0 & \mathcal{B}_{j-1}^{o-} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} y \\ z_{2j+1}^{+\{2l+1\}} = \begin{bmatrix} t_{2j+1}^{\{2l+1\}} & \begin{bmatrix} T_{2j-1} \\ 0 \end{bmatrix} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{diag}(w_{2j+1}^{\{k>0\}}) \end{bmatrix} \times \begin{bmatrix} [b_{2l+1}^{\{2p\}} \otimes b_{j-l}^{\{q\}}]^{\{l\}\{k\}} & 0 \\ 0 & \mathcal{B}_{j-1}^{o+} \end{bmatrix} \times \begin{bmatrix} X_j^{\{q\}} \\ \mathcal{X}_{j-1} \end{bmatrix} x \end{cases}. \tag{A.15}$$

This is an extension of Eq. A.13, where we have distinguished four submatrices $\mathcal{B}_{j-1}^{e-}, \mathcal{B}_{j-1}^{e+}, \mathcal{B}_{j-1}^{o-}$ and $\mathcal{B}_{j-1}^{o+}$ for the *sin* and *cos* components of even and odd orders, respectively, as well as the submatrix

with index (l,k) introduced by the intermediate variable $J_{kl}=j-k-l$, as follows:

$$B_{2j}^{+}=\left[b_{2l}^{\{2p\}}\otimes b_{j-l}^{\{q\}}\right]^{\{l\}\{k\}}=\begin{bmatrix} b_{0}^{\{2p\}}\otimes b_{j}^{\{q\}} \\ b_{2}^{\{2p\}}\otimes b_{j-1}^{\{q\}} \\ \vdots \\ b_{2j}^{\{2p\}}\otimes b_{0}^{\{q\}} \end{bmatrix}_{(j+1)\times(j+1)} . \tag{A.16}$$

Each submatrix in the matrix above corresponds to a convolution of two vectors as follows:

$$b_{2l}^{\{2p\}}\otimes b_{j-l}^{\{q\}}=\left(\begin{bmatrix} (-1)^{0}b_{2l}^{0} \\ (-1)^{1}b_{2l}^{2} \\ \vdots \\ (-1)^{l}b_{2l}^{2l} \end{bmatrix}^{T}\right)_{1\times(l+1)}\times\begin{bmatrix} b_{j-l}^{0} & b_{j-l}^{1} & \cdots & b_{j-l}^{j-l} & 0 & \cdots & 0 \\ 0 & b_{j-l}^{0} & \cdots & b_{j-l}^{j-l-1} & b_{j-l}^{j-l} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & b_{j-l}^{0} & b_{j-l}^{1} & \cdots & b_{j-l}^{j-l} \end{bmatrix}_{(l+1)\times(j+1)} , \tag{A.17}$$

which performs the multiplication between the azimuth polynomials and the radial polynomials for $n=2j$ of even order, $+m=+(2l)$ part, just as $b_{2l}^{\{2p+1\}}\otimes b_{j-l}^{\{q\}}$ for $-m=-(2l)$ part, while $b_{2l+1}^{\{2p\}}\otimes b_{j-l}^{\{q\}}$ and $b_{2l+1}^{\{2p+1\}}\otimes b_{j-l}^{\{q\}}$ stand for $n=2j+1$ of odd order, $+m=(2l+1)$ and $-m=-(2l+1)$ parts, respectively.

Overall, Eq. A.14 and its matrix form (Eq. A.15) show two cases of block–wise recurrence in complete *Zernike* calculations, which are irrelevant to the coordinates, since Eq. A.15 can be represented as follows:

$$\begin{cases} z_{2j}^{-\{2l\}} = T_{2j}\times \mathrm{diag}(w_{2j}^{\{k\}})\times \mathcal{B}_{j}^{e-}\times \mathcal{X}_{j}x^{-1}y \\ z_{2j}^{+\{2l\}} = T_{2j}\times \mathrm{diag}(w_{2j}^{\{k\}})\times \mathcal{B}_{j}^{e+}\times \mathcal{X}_{j} \\ z_{2j+1}^{-\{2l+1\}} = T_{2j+1}\times \mathrm{diag}(w_{2j+1}^{\{k\}})\times \mathcal{B}_{j}^{o-}\times \mathcal{X}_{j}y \\ z_{2j+1}^{+\{2l+1\}} = T_{2j+1}\times \mathrm{diag}(w_{2j+1}^{\{k\}})\times \mathcal{B}_{j}^{o+}\times \mathcal{X}_{j}x \end{cases} \tag{A.18}$$

where we have

$$\begin{cases} \mathcal{B}_{0}^{e-}=0, \mathcal{B}_{0}^{e+}=1 \\ \mathcal{B}_{0}^{o-}=1, \mathcal{B}_{0}^{o+}=1 \\ \mathcal{B}_{j}^{e-}=\begin{bmatrix} B_{2j}^{-} & 0 \\ 0 & \mathcal{B}_{j-1}^{e-} \end{bmatrix}, B_{2j}^{-}=\left[b_{2l}^{\{2p+1\}}\otimes b_{j-l}^{\{q\}}\right]^{\{l\}\{k\}} \\ \mathcal{B}_{j}^{e+}=\begin{bmatrix} B_{2j}^{+} & 0 \\ 0 & \mathcal{B}_{j-1}^{e+} \end{bmatrix}, B_{2j}^{+}=\left[b_{2l}^{\{2p\}}\otimes b_{j-l}^{\{q\}}\right]^{\{l\}\{k\}} \\ \mathcal{B}_{j}^{o-}=\begin{bmatrix} B_{2j+1}^{-} & 0 \\ 0 & \mathcal{B}_{j-1}^{o-} \end{bmatrix}, B_{2j+1}^{-}=\left[b_{2l+1}^{\{2p+1\}}\otimes b_{j-l}^{\{q\}}\right]^{\{l\}\{k\}} \\ \mathcal{B}_{j}^{o+}=\begin{bmatrix} B_{2j+1}^{+} & 0 \\ 0 & \mathcal{B}_{j-1}^{o+} \end{bmatrix}, B_{2j+1}^{+}=\left[b_{2l+1}^{\{2p\}}\otimes b_{j-l}^{\{q\}}\right]^{\{l\}\{k\}} \end{cases} , \tag{A.19}$$

such that both $T_{n=2j}$, $T_{n=2j+1}$, $\mathcal{B}_{j=n/2}^{e\pm}$, and $\mathcal{B}_{j=(n-1)/2}^{o\pm}$ have their recursive submatrix decomposition and are fully supported by an n–row *Pascal* triangle.

A.4. Block–wise *Zernike* calculation

Since $w_n^0 \equiv 1$ always holds, all formulas in Eq. A.18 have an alternative pseudo–iterative form. For example, we have

$$z_{2j}^{+\{2l\}} = \left(t_{2j}^{\{l\}} \odot B_{2j}^{+}\right) \times X_j^{\{q\}} + T_{2(j-1)} \times \mathrm{diag}\left(w_{2j}^{\{k>0\}}\right) \times \mathcal{B}_{j-1}^{e+} \times \mathcal{X}_{j-1} \tag{A.20}$$

where $\odot$ denotes the *Hadamard* product as element–wise multiplication between a vector and a matrix, e.g., we have

$$t^{\{l\}} \odot B^{\{l\}\{k\}} = \begin{bmatrix} t^0 \\ t^1 \\ \vdots \\ t^j \end{bmatrix}_{(j+1)\times 1} \odot \begin{bmatrix} B^{0,0} & B^{0,1} & \cdots & B^{0,j} \\ B^{1,0} & B^{1,1} & \cdots & B^{1,j} \\ \vdots & \vdots & \ddots & \vdots \\ B^{j,0} & B^{j,1} & \cdots & B^{j,j} \end{bmatrix}_{(j+1)\times(j+1)} = \begin{bmatrix} t^0 B^{0,0} & t^0 B^{0,1} & \cdots & t^0 B^{0,j} \\ t^1 B^{1,0} & t^1 B^{1,1} & \cdots & t^1 B^{1,j} \\ \vdots & \vdots & \ddots & \vdots \\ t^j B^{j,0} & t^j B^{j,1} & \cdots & t^j B^{j,j} \end{bmatrix}_{(j+1)\times(j+1)} . \tag{A.21}$$

Eq. A.20 is equivalent to

$$z_{2j}^{+\{2l\}} = \sum_{k=0}^{j} w_{2j}^{k} \left(t_{2(j-k)}^{\{l\}} \odot B_{2(j-k)}^{+}\right) \times X_{(j-k)}^{\{q\}}, \tag{A.22}$$

which offers us a direct block–wise transformation method for calculating the *cos*–relevant components of even–order ($n = 2j$) *Zernike* basis functions in the *Cartesian* coordinate system. Such a transformation applies to all *Zernike* basis functions, as shown in Eq. 10.

Appendix B: Forward and inverse transformation matrices, up to order 6

n		0	1		2			3				4					5						6						
	$i\backslash m$	0	-1	1	-2	2	0	-3	-1	1	3	-4	-2	0	2	4	-5	-3	-1	1	3	5	-6	-4	-2	0	2	4	6
0	0	1	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
1	0	0	0	1	0	0	0	0	0	-2	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	-2	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	2	0	0	0	0	0	0	-6	-3	0	0	0	0	0	0	0	0	0	0	12	6	0	0
	1	0	0	0	0	2	0	0	0	0	0	0	-6	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0
	2	0	0	0	0	0	2	0	0	0	0	0	0	-6	3	0	0	0	0	0	0	0	0	0	0	12	-6	0	0
3	0	0	0	0	0	0	0	3	0	0	1	0	0	0	0	0	0	0	0	-12	-4	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	-12	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	-12	0	12	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	-1	3	0	0	0	0	0	0	0	4	0	-12	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	4	1	0	0	0	0	0	0	0	0	0	0	-30	-20	-5	0
	1	0	0	0	0	0	0	0	0	0	0	4	8	0	0	0	0	0	0	0	0	0	0	-20	-40	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	12	0	-6	0	0	0	0	0	0	0	0	0	-60	0	30	0
	3	0	0	0	0	0	0	0	0	0	0	-4	8	0	-4	1	0	0	0	0	0	0	0	20	-40	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	6	-4	1	0	0	0	0	0	0	0	0	0	-30	20	-5	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	5	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	15	10	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-10	-10	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	20	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	-15	5	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-5	10	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	15	6	1
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	24	30	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	15	-30	-15	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	-15	-30	15	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-24	30	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	-15	6	-1

Table B.1. The forward transformation matrix $\mathcal{T}$, up to order 6.

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	1.00	0	0	0	0	0	0
1	-1	0	1.00	0	0	0	0	0
	1	0	1.00	0	0	0	0	0
2	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
3	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
5	-5	0	0	0	0	0	0	0
	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
6	-6	0	0	0	0	0	0	0
	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	1.00	0	0	0	0	0	0
1	-1	0	0	0.25	0	0	0	0
	1	0	1.00	0	0	0	0	0
2	-2	0	0	0	0	0	0	0
	0	0	0	0.25	0	0	0	0
	2	0	0	0.50	0	0	0	0
3	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
5	-5	0	0	0	0	0	0	0
	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
6	-6	0	0	0	0	0	0	0
	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0.13	0	0
1	-1	0	0	0	0.50	0	0	0
	1	0	0	0.17	0	0	0.31	0
2	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0.19	0	0
	2	0	0	0	0	0.38	0	0
3	-3	0	0	0	-0.25	0	0	0
	-1	0	0	0	0.25	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
5	-5	0	0	0	0	0	0	0
	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
6	-6	0	0	0	0	0	0	0
	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	0	0	0.04	0	0.13	0	0
1	-1	0	0	0	0	0	0.06	0
	1	0	0	0	0	0	0.31	0
2	-2	0	0	0	0	0	0	0
	0	0	0	0	0.19	0	0	0
	2	0	0	0.00	0	0.19	0	0
3	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0.05	0
	1	0	0	0	0	0	0.25	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0	-0.13	0	0	0
	-2	0	0	0	0.06	0	0	0
	0	0	0	0.02	0	0.06	0	0
	2	0	0	0.00	0	-0.13	0	0
	4	0	0	-0.13	0	0.13	0	0
5	-5	0	0	0	0	0	0	0
	-3	0	0	0	0	0	0.04	0
	-1	0	0	0	0	0	0.01	0
	1	0	0	0	0	0	0.06	0
	3	0	0	0	0	0	0.06	0
	5	0	0	0	0	0	0.06	0
6	-6	0	0	0	0	0	0	0
	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0	0.08
1	-1	0	0	0	0.06	0	0	0
	1	0	0.06	0	0	0	0	0
2	-2	0	0	0.06	0	0.06	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0.14
3	-3	0	0	0	0	0	0	0.28
	-1	0	0	0.05	0.05	0	-0.25	0
	1	0	0.05	0.05	0.05	0	0.25	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0.05	0	0.05	0	0
	-2	0	0	-0.05	0	-0.15	0	0
	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0
5	-5	0	0	-0.06	0	0.06	0	0
	-3	0	0	0.01	0	0	-0.06	0
	-1	0	0	0.01	0	0.01	0.06	0
	1	0	0	0	0.01	0	0	0
	3	0	0	-0.01	0	-0.04	0	0
	5	0	0	-0.06	0	0.06	0	0
6	-6	0	0	0	0	0	0	0
	-4	0	0	0	0	0	0	0
	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.02
	2	0	0	0	0	0	0	0.03
	4	0	0	0	0	0	0	0.03
	6	0	0	0	0	0	0	0.03

n	$m\backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0	0.08
1	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
2	-2	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0.09
	2	0	0.09	0	0.06	0	0	0
3	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
4	-4	0	0	0	0	0	0	0
	-2	0	0.10	0	0.00	0	-0.10	0
	0	0	0	0	0.03	0	0.05	0
	2	0	0.05	0	0	0	0	0
	4	0	0	0	0	0	0	0
5	-5	0	0	0	0	0	0	0
	-3	0	0	0	0	0	0	0
	-1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0
6	-6	0	0.03	0	-0.03	0	0.03	0
	-4	0	0.02	0	0.00	0	-0.02	0
	-2	0	0.01	0	0.01	0	0.01	0
	0	0	0	0	0	0	0	0
	2	0	0.00	0	0	0	0	0
	4	0	0.00	0	0	0	0	0
	6	0	0.03	0	0.03	0	0.03	0

Table B.2. The inverse transformation matrix $\mathcal{T}^{-1}$, up to order 6, with 2 decimal digits for visualization.