Recurrence relations for Zernike annular polynomials

Michael Peck <mlpeck54@gmail.com>

February 16, 2023

1 Introduction

It's well known that every sequence of real valued univariate orthogonal polynomials satisfy recurrence relations of the generic form [5, §18.2]

$$p_{k+1}(x) = (a_k x + b_k) p_k(x) - c_k p_{k-1}(x). \tag{1}$$

In the special case of *monic* orthogonal polynomials, that is ones for which the highest order term has coefficient 1 this simplifies to

$$p_{k+1}(x) = (x - a_k) p_k(x) - b_k p_{k-1}(x) \tag{2}$$

where $p_k(x)$ is an orthogonal polyomial sequence initialized with $p_{-1}(x) = 0, p_0(x) = 1$.

It's also well known that when numerical values are required it's generally preferable to use a recurrence relation rather than explicit formulas to calculate them, provided the former are numerically stable. Recurrences are especially valuable when a complete sequence of polynomial values is required since no calculations are wasted.

Several recurrence relations are known for Zernike radial polynomials that were derived based on their relation to various classical systems of orthogonal polynomials. In particular recurrences discovered by Prata and Rusch [9, their equation 6] and Shakibaei and Paramesran [11] are highly stable and efficient, with the latter being superior when a full set of polynomial values is required. Their results were extended by Andersen [1] to Zernikes and their gradients in cartesian coordinates. Andersen also carefully studied the stability of the algorithm and found that errors using double precision arithmetic don't exceed $\approx 10^{-13}$ for radial orders up to 50, which is far higher than needed for most applications.

Somewhat to my surprise I have been unable to find a published recurrence relation for radial Zernike *annular* polynomials in a fairly extensive literature search. A different form of recursive evaluation was proposed by Tatian [12] and repeated with some claimed corrections by Mahajan [7], but I have been unable to verify their correctness. Only a small number of explicit formulas have been published – Mahajan [8] gives a complete listing up to radial order 6 with a few additional higher order radial polynomials. This covers the annular Zernike polynomial analogs of the primary and secondary classical optical aberrations only, which is insufficient for many applications.

In this note I'm going to present an algorithm for *numerically* generating the recurrence coefficients (a_k and b_k in equation 2 above), which are then used to generate the radial Zernike values. The algorithm is an application of the method of modified moments extending the classical method of moments of Chebyshev [6, section 2.3]. I will present the algorithm, which closely follows Gautschi [6, section 2.4], in the next section. The main purpose is to explain the logic and implementation of the algorithm. A number of mathematical details are glossed over; the original references should be consulted if the reader is interested. Some examples and a brief discussion of numerical precision and stability are in section 4.

The algorithm is coded in C++ and makes use of the open source linear algebra library Armadillo. The code is part of a larger R library for manipulation of Zernike polynomials and interferogram fringe analysis, but is designed to be usable independently of the R system. An extended precision version of the code is also available that uses a very lightly modified copy of `fastgl` [3] to fetch the nodes and weights for Gauss-Legendre quadrature (about which see below) and quad-precision arithmetic for the most rounding error prone steps. The source code is freely available and the functions authored by me have a permissive license (MIT).

Following Mahajan [8], the orthonormal Zernike annular polynomials have the form

$$Z_n^m(\rho,\theta;\epsilon) = \begin{cases} \sqrt{2(n+1)}R_n^m(\rho;\epsilon)\cos(m\theta) & m \neq 0 \\ \sqrt{2(n+1)}R_n^m(\rho;\epsilon)\sin(m\theta) & m \neq 0 \\ \sqrt{n+1}R_n^m(\rho;\epsilon) & m = 0 \end{cases} \tag{3}$$

where $0 \leq \epsilon < 1$ is the obstruction fraction, $\epsilon \leq \rho \leq 1$ is the normalized radial coordinate and the indexes n, m satisfy $m \geq 0, n \geq m$ and $n - m$ is even. This is exactly the form of the *Zernike* polynomials, to which they reduce when $\epsilon = 0$.

When numerical values are required the computational challenge is to evaluate the radial polynomials $R_n^m(\rho;\epsilon)$. In order for the Zernike annular polynomials to be orthonormal in both indexes n, m the radial polynomials must satisfy the relation

$$\int_\epsilon^1 R_n^m(\rho;\epsilon)R_{n'}^m(\rho;\epsilon)\rho \mathrm{d}\rho = \frac{1-\epsilon^2}{2(n+1)}\delta_{nn'} \tag{4}$$

Because of the conditions the indexes must satisfy we can let $n = 2j + m$ with $j = 0, 1, \ldots$. Again following Mahajan, who in turn follows Born and Wolf [4, Appendix VII] for the case of Zernike polynomials, letting $u = \rho^2, \epsilon^2 \leq u \leq 1$ we can write the radial polynomials as

$$R_{2j+m}^m(\rho;\epsilon) = \left[\frac{1-\epsilon^2}{2(2j+m+1)h_j^m}\right]^{1/2} \rho^m Q_j^m(u) \tag{5}$$

where $[Q_j^m(u)]$ is a set of orthogonal polynomials that orthogonalize $1, u, \ldots, u^j$ over the interval $[\epsilon^2, 1]$ with weight function u^m, that is they satisfy

$$\frac{1}{2h_j^m}\int_{\epsilon^2}^1 Q_j^m(u)Q_{j'}^m(u)u^m \mathrm{d}u = \delta_{jj'} \tag{6}$$

Note that for each value of the azimuthal index m there is a *separate* sequence of orthogonal polynomials $[Q_j^m, j = 0, 1, \ldots]$. In the next section I show how to recursively generate the recurrence relations for these polynomials, which are then used to generate the full set of desired Zernike polynomial values.

2 Algorithm

The problem I'm trying to solve is to calculate a matrix of Zernike annular polynomials for a complete set of aberrations, that is all allowable combinations of radial indexes $n = 0, 1, \ldots, N$ and azimuthal indexes

$m = 0, 1, \ldots, M \le N$ at a possibly lengthy set of polar coordinates (ρ, θ). For the two single index schemes I support $M = N$ or $M = N/2$. This amounts to solving for the polynomials Q_j^m and normalization factors $c_j^m \equiv 2h_j^m$ in Equation 6 above, then substituting in Equation 5 to get the radial Zernike component, and finally Equation 3 to get the full Zernike expression. To solve for the polynomials Q_j^m, which I will treat as monic, we require the recurrence coefficients $a_k, b_k, k = 0, \ldots, j-1$ in equation 2.

The method I use to calculate the required recurrence coefficients is called the "modified Chebyshev algorithm" by Gautschi [6] because it extends Chebyshev's classical method of moments, which unfortunately is nuerically extremely ill-conditioned. I will adopt Gautschi's notation for the following discussion. There is also a brief description of the method in Press et al. [10, section 4.5].

The algorithm requires a sequence of polynomials orthogonal with respect to some weight function and with known recurrence coefficients. It turns out we have one: when $m = 0$ the polynomials are just Legendre polynomials shifted to the interval $[\epsilon^2, 1]$, which in monic form have recurrence coefficients

$$\begin{aligned} a_k &= (1+\epsilon^2)/2 \quad \forall k = 0, \ldots \\ b_k &= \tfrac{k^2(1-\epsilon^2)^2}{4(4k^2-1)} \qquad k = 1, \ldots \end{aligned} \tag{7}$$

Besides serving as the known function in the modified moments procedure the recurrence coefficients give us the nodes and weights for Gauss-Legendre quadrature. If we use the Golub-Welsch [10, section 4.5] procedure to determine these the nodes will automatically fall in the open interval $(\epsilon^2, 1)$.

The modified moments under weight function u^m are given by

$$\nu_k = \int_{\epsilon^2}^{1} p_k(u) u^m \mathrm{d}u, \;\; k = 0, \ldots \tag{8}$$

where p_k is calculated recursively from equation 2 and the recurrence coefficients in equation 7 with initialization $p_0 = 1$. Noticing that the integrand above is itself a polynomial, we can with a large enough number of precomputed nodes and weights replace the integral with a nominally exactly equivalent Gauss-Legendre quadrature

$$\nu_k = \sum_{q=0}^{N_q-1} w_q p_k(u_q) u_q^m \tag{9}$$

The mumber of quadrature points required is calculated as follows: for any azimuthal index m if the maximum radial order of interest is N we need $(N-m)/2+1$ pairs of recurrence coefficients for the polynomials $Q_j^m(u)$ up to order $(N-m)/2$. For this we need $2[(N-m)/2+1] = N-m+2$ moments. Noting that the weight function is u^m the maximum order polynomial required is $N+1$. Gauss-Legendre quadrature with N_q nodes is nominally exact for polynomials of order up to $2N_q - 1$, so we need $N/2+1$ nodes. From numerical experiments I find that a few more nodes are useful (5 or so more than required), but many more waste CPU cycles and result in larger rounding errors. Also note that p_k is orthogonal to u^m for $k \ge m+1$, so there are at most $m+1$ non-zero modified moments. Often then only a handful of quadratures need to be computed.

Before proceeding with the remainder of the algorithm there are two special cases to check. First, if $m = M$ there is only one "polynomial", namely $Q_0^m = 1$ with normalization

$$\begin{aligned} c_0^m &= \textstyle\int_{\epsilon^2}^{1} u^m \mathrm{d}u \\ &= \tfrac{1-\epsilon^{2(m+1)}}{m+1} \end{aligned}$$

and, after substituting in equation 5 we obtain

$$R_m^m(\rho;\epsilon) = \left[\frac{1-\epsilon^2}{1-\epsilon^{2(m+1)}}\right]^{1/2} \rho^m \tag{10}$$

Mahajan gives an equivalent expression in a different form in [7] and in identical form in [8].

The second special case is $m = 0$ where we already have the desired recurrence coefficients (equation 7) and we can proceed directly to recursively calculate the polynomials $Q_k^0(u), k = 0, \ldots, N/2$. The normalization factors c_k^0 are also calculated recursively as

$$\begin{aligned} c_0^0 &= 1-\epsilon^2 \\ c_k^0 &= b_k \times c_{k-1}^0 \quad k = 1, \ldots, N/2 \end{aligned} \tag{11}$$

where the b_k are the recurrence coefficients in equation 7.

For the rest of the algorithm description I closely follow Gautschi [6, section 2.4]. Given a maximum radial order N and azimuthal order m we need to construct a set of $K+1 = (N-m)/2+1$ orthogonal polynomials through order K, which is tantamount to calculating the recurrence coefficients α_k, β_k for $k = 0, \ldots, K-1$. For brevity I am suppressing the superscript m in the following but keep in mind there is a separate sequence of polynomials and therefore a separate function call for each value of m in the desired matrix of values. The algorithm is initialized as follows.

$$\begin{aligned} \sigma_{-1,\ell} &= 0 && \ell = 1, \ldots, 2K-2 \\ \sigma_{0,\ell} &= \nu_\ell && \ell = 0, \ldots, 2K-1 \\ \alpha_0 &= a_0 + \nu_1/\nu_0 \\ \beta_0 &= \nu_0 \\ c_0 &= \beta_0 \end{aligned} \tag{12}$$

The continuation for $k = 1, \ldots, K-1$ is:

$$\begin{aligned} \sigma_{k\ell} &= \sigma_{k-1,\ell+1} - (\alpha_{k-1} - a_\ell)\sigma_{k-1,\ell} - \beta_{k-1}\sigma_{k-2,\ell} \\ &\quad + b_\ell \sigma_{k-1,\ell-1} && \ell = k, \ldots, 2K-k-1 \\ \alpha_k &= a_k + \sigma_{k,k+1}/\sigma_{k,k} - \sigma_{k-1,k}/\sigma_{k-1,k-1} \\ \beta_k &= \sigma_{k,k}/\sigma_{k-1,k-1} \\ c_k &= \beta_k \times c_{k-1} \end{aligned} \tag{13}$$

Again, the a_k, b_k are the recurrence coefficients for the shifted Legendre polynomials given in equation 7 above with a_k constant for all k. As a programming note, at each iteration k we only use $\sigma_{k-2,\ldots}, \sigma_{k-1,\ldots}$ to calculate $\sigma_{k,\ldots}$, so the variable `sigma` is declared as a 3 by $2K$ array, with the values pushed down at each iteration. With the recurrence coefficients in hand the polynomials Q_k^m are given by

$$\begin{aligned} Q_0^m(u) &= 1 \\ Q_1^m(u) &= u - \alpha_0 \\ Q_{k+1}^m(u) &= (u-\alpha_k)Q_k^m(u) - \beta_k Q_{k-1}^m(u) \quad k = 1, \ldots, K-1 \end{aligned} \tag{14}$$

Finally these are substituted into equation 5 and returned as a matrix.

3 A few notes on indexing

The final data product from this exercise is a matrix of Zernike annular polynomials, which for P coordinate pairs (ρ_p, θ_p) and N_Z distinct Zernikes will be dimension $P \times N_Z$. Naturally we need a mapping from the two indexes of the Zernikes to a single column index. There are a number of indexing schemes in use, and I currently support two of them.

First is an extension of the FRINGE set, which is popular for optical design and testing applications. This groups Zernikes by values of $o = n + m$ with $o = 0, 2, \ldots,$ up to a desired maximum order $\mathcal{O}$. The complete set will have $(\mathcal{O}/2 + 1)^2$ columns with the radial index n ranging from 0 to $\mathcal{O}$ and the azimuthal index m from 0 to $\mathcal{O}/2$. For the axisymmetric aberrations $(m = 0)$ the column indexes are

$$k = n^2/4 + n \quad n = 0, 2, \ldots, \mathcal{O}$$

while for $m \geq 1$ the cosine components are in columns

$$k = (n + m)^2/4 + n - m \quad n = m, m + 2, \ldots, \mathcal{O} - m$$

and the sine components in the following columns.

This is a popular indexing scheme and it's quite common to refer to aberrations by their 0 indexed location, so for example (Z1, Z2) are x and y tilt, Z3 is defocus, (Z4, Z5) are astigmatism, (Z6, Z7) coma, Z8 is (balanced) primary spherical aberration, etc. Components Z9 through Z15 correspond to the classical secondary aberrations from trefoil through secondary spherical aberration.

The second scheme, which I believe is an ISO/ANSI standard, groups aberrations by value of the radial index n, with n ranging from 0 up to a desired maximum N and for each n, m takes all values from either $1, 3, \ldots, n$ or $0, 2, \ldots, n$ depending on whether n is odd or even. There are a total of $(N + 1)(N + 2)/2$ columns in the complete set, approximately twice as many as in the FRINGE set with the same maximum radial order. The axisymmetric aberrations $(m = 0)$ are in columns

$$k = (n^2 + 2n)/2 \quad n = 0, 2, \ldots, N$$

For each $m \geq 1$ there are $\lfloor N/2 \rfloor + 1$ radial annular Zernikes up to maximum order N or $N - 1$ depending on whether m is even or odd. The *sine* components will be placed in columns

$$k = (n^2 + 2n - m)/2 \quad n = m, m + 2, \ldots$$

while the cosine components will be separated by m columns:

$$k = (n^2 + 2n + m)/2 \quad n = m, m + 2, \ldots$$

Although this is a very logical ordering it does have the disadvantage that the components of aberrations with the same azimuthal order are separated by one or more columns, except for the coma-like pairs $(m = 1)$ which are always in adjacent columns. In the context of optical testing this makes visual inspection of tables of aberration coefficients more difficult and there are some surprises for those accustomed to the FRINGE set. For example the components of trefoil bracket those of primary coma, and similarly the components of astigmatism of all orders bracket defocus and spherical aberration of the same orders.

In optical testing applications defects often have a fairly high degree of symmetry with "zonal" defects and relatively low order asymmetric aberrations often being the most significant. So while it might be necessary to use fairly high radial orders to capture, for example, edge runout using similarly large azimuthal orders could be overkill.

4 Numerical stability issues and examples

Gautschi claims that while Chebyshev's classical method of moments is hopelessly numerically unstable, the modified moments procedure *may* be stable. It's easy enough to check the first claim since the monomial moments are trivial to write down [1], and in fact it is hopeless. I would not exclude the possibility of deriving a symbolic expression for the recurrence coefficients however. This is after all how recurrences for the classical orthogonal polynomials were derived. More advanced computer algebra software than I have on hand would be essential though.

A first and very weak test of numerical stability is to compare to explicit formulas. The only tabulation I have found, by Mahajan [8], is complete through radial and azimuthal order 6 only, with a few higher radial order examples. This is sufficient to calculate the first 28 Zernike annular polynomials in ISO sequence. For this exercise I extracted polar coordinates from a uniform grid of points in a 1024×1024 array with 50% obstruction fraction, for a total of ≈ 615K coordinate pairs. I have a simple R function that evaluates Mahajan's formulas and creates a matrix of Zernike annular polynomial values, to which I compare the values from this algorithm. The standard deviation of the difference of all values in the matrixes returned by the two methods was $\approx 2.4 \times 10^{-15}$, while the range of differences was $\approx \pm 6 \times 10^{-14}$, so the agreement was typically about 14 digits and at worst around 13. This is well within expected double precision numerical accuracy. Of course this is mostly a test of code correctness. One wouldn't expect rounding errors to be excessive at these low orders.

A considerably more stringent test results from the fact that Zernike annular polynomials with an obstruction fraction of zero are just Zernike polynomials. I have well tested code that implements the algorithm described by Andersen [1] for either the extended FRINGE set or in ISO sequence with cartesian coordinates. As mentioned in the Introduction these return values with at least 12 digit precision in double precision arithmetic.

For this exercise I created matrices of Zernikes up to radial and azimuthal order 40 in ISO sequence on a grid of ≈ 320K coordinate pairs. This comprises 861 distinct aberrations. Again the quality metric I use is the standard deviation of the difference of the values calculated with the annular Zernike routines from those of the Zernike polynomial code, broken down by radial and azimuthal order. In the figure below this metric is plotted against the azimuthal order m and color coded by radial order n. The vertical scale is logarithmic to show the (inverse of) number of digits of precision. As mentioned in the Introduction there are currently two versions of the code. The first uses the Armadillo routine `eig_sym` to calculate the quadrature nodes and weights using the Golub-Welsch method, with all arithmetic done in double precision. The second uses the routine `fastgl` retrieved from sourceforge and described in [2]. The code was edited to use the `float128` type from the boost multiprecision library for all variable declarations. The source code lists all constants with 25 digits precision, which is considerably higher than double precision. `float128` arithmetic is also used for the recurrences in equations 12 and 13. For compatibility all results are then cast to doubles to fill in the Zernike matrices.

At first sight the performance of the double precision version seems somewhat disappointing, with steadily increasing departures from the Zernike code results up to azimuthal order $m = 20$, where there is only about 1 digit precision at the highest radial order of $n = 40$. The extended precision version has an extra 2 to 4 digits of precision for all orders. Interestingly, at high orders $(m, n \geq 20)$ the extended precision results are no worse than what was claimed for the Zernike recurrence [1].

But even in double precision there is only a gradual loss of precision rather than catastrophic failure. Figure 2 displays the worst case disagreement at $m = 20, n = 40$. The maps have the same basic shape, with the

[1] the monomial moments are $\mu_\ell^m = \int_{\epsilon^2}^1 u^l u^m \mathrm{d}u = \frac{1-\epsilon^{2(\ell+m+1)}}{\ell+m+1}$.

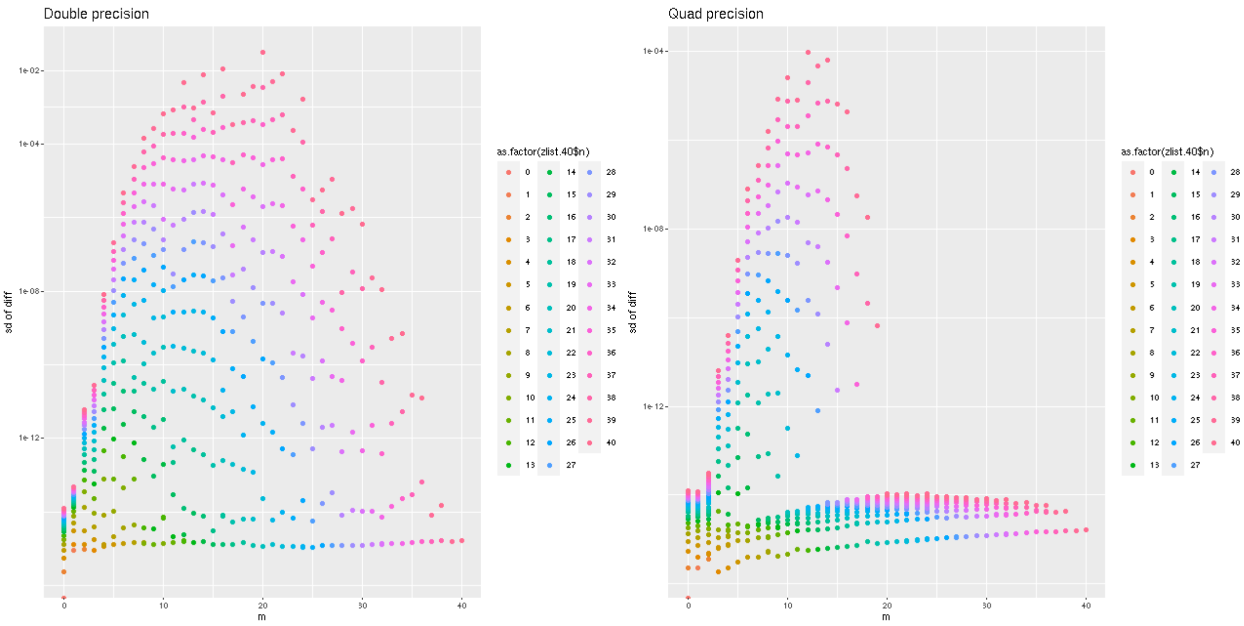

Figure 1: Standard deviation of the difference between annular Zernike computation and Zernike by aberration index up to order 40. (L) double precision arithmetic. (R) Quad precision (`float128`)

largest differences in the first ring of local minima and maxima.

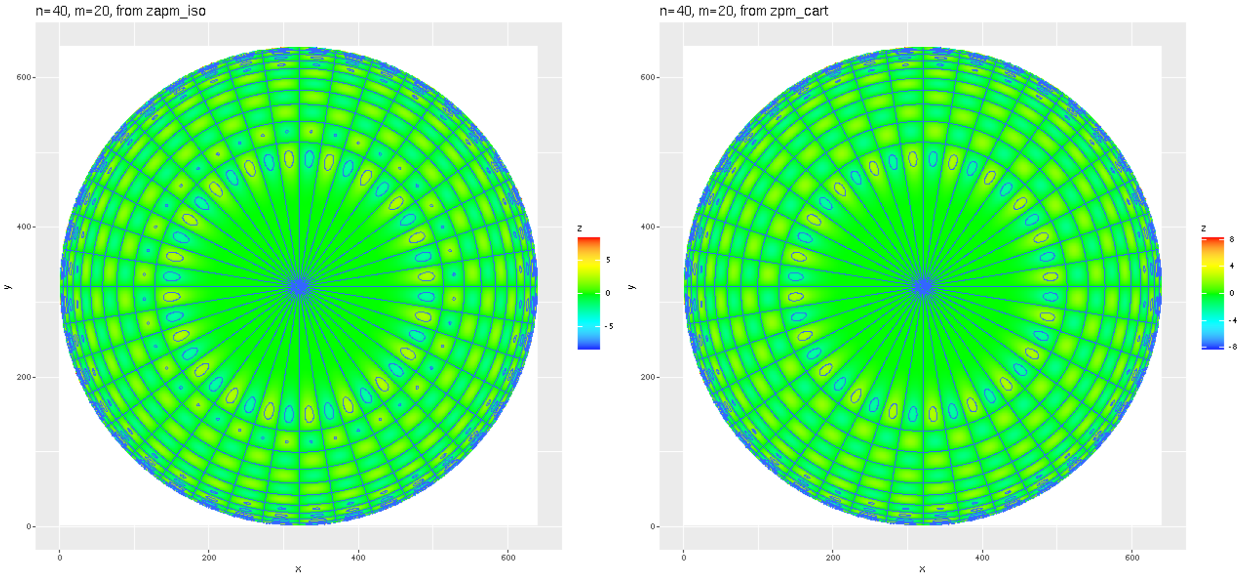

Figure 2: The worst match between `zapm_iso` and `zpm_cart`, n=40, m=20

A look at the profiles of the radial polynomials (figure 3) shows that they have the same number and to a good approximation location of zero crossings and local minima and maxima. Another interesting result is that the condition number of the 3 matrices, which in this context provides a measure of how close they are to being exactly orthogonal, is virtually the same for all 3 at ≈ 1.58. This means they are equally well suited to wavefront fitting by least squares. There would be small differences in aberration coefficients, but usually only those of low order aberrations are likely to be of interest.

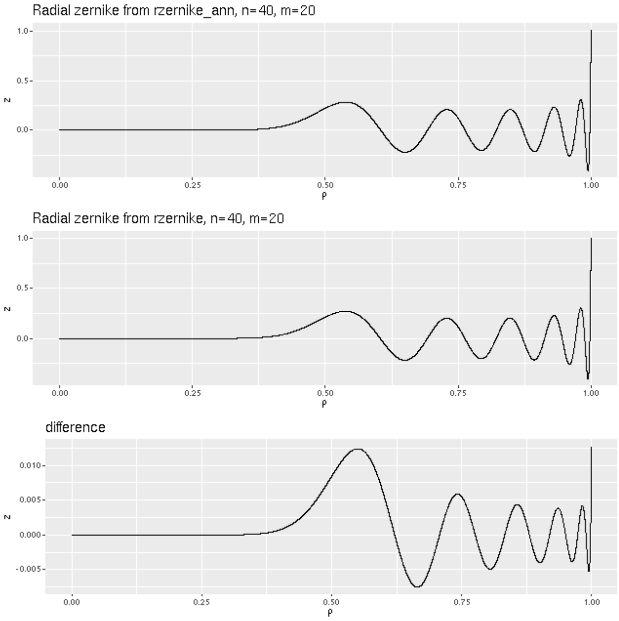

Figure 3: Radial polynomials from `zapm_iso` and `zpm_cart`, and their difference, n=40, m=20

For a more realistic case, figure 4 makes the same comparisons for a 40th order set in the extended FRINGE sequence (441 aberrations). This is the largest set that I have used in an actual wavefront fit, and is considerably more than would normally be needed in a fringe analysis application.

Normally one only needs dozens to a few hundred Zernike components for optical design and testing, and sets of this size are entirely unproblematic, with at least 12 or 13 digits precision even with double precision arithmetic.

I conclude with a few examples. First, figure 5 is a graph of several of the lower order radial Zernike and annular Zernike polynomials for the axisymmetric ($m = 0$) aberrations corresponding to defocus and various orders of balanced spherical aberration.

Since, as already mentioned, the annular polynomials are just Legendre polynomials shifted into the interval $[\epsilon^2, 1]$ they have the same shape as their Zernike counterparts but "squeezed" into the smaller interval. In particular they have the same number of zero crossings, amplitudes, and values at the endpoints. Thus their shape is quite different than their Zernike counterparts truncated to the range $[\epsilon, 1]$, and in a wavefront fitting exercise their coefficients can be quite different. This is one reason why it is preferable to use Zernike annular polynomials in optical testing of obstructed apertures.

On the other hand when $m = n$ the annular Zernikes have the same functional form as their Zernike counterparts truncated to the annular region, except for a constant scale factor to maintain unit variance. This is illustrated for trefoil ($m = 3, n = 3$) in figure 6.

5 Software availability

As mentioned in the introduction the software described in this note is part of an R package for manipulation of Zernike polynomials and fringe analysis maintained in a Github repository at github.com/mlpeck/zernike. The specific files referenced here are `zapm.cpp` and `zapm_128.cpp`.

The extended precision code also requires `fastgl_128.cpp` and `fastgl_128.h` [3].

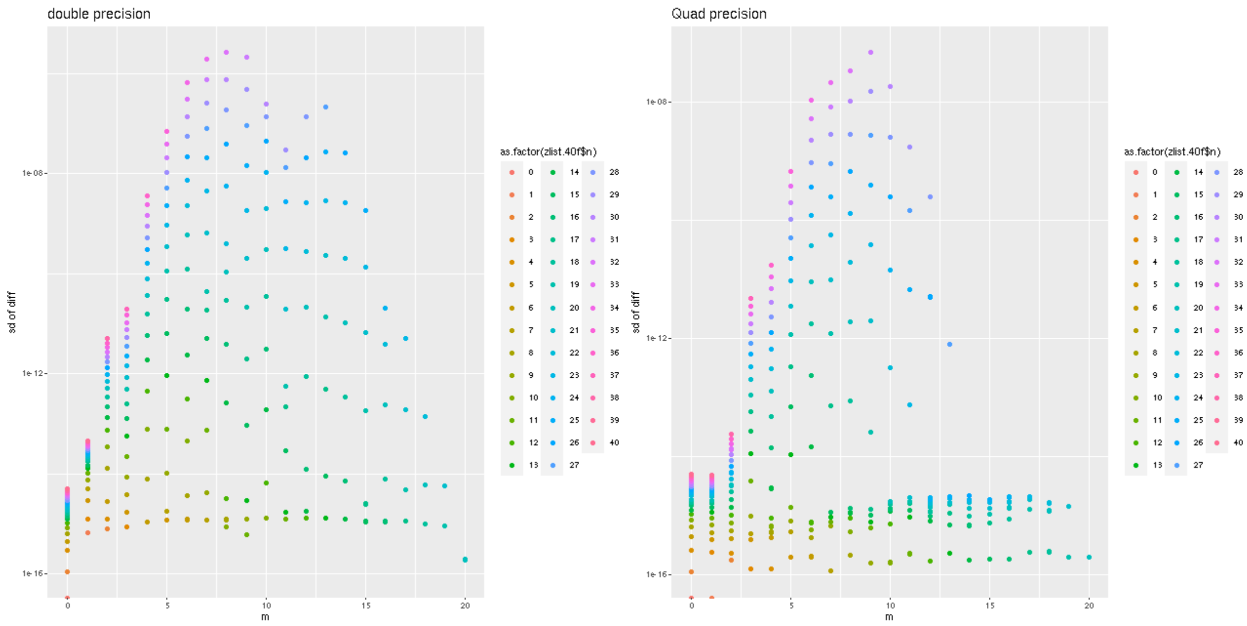

Figure 4: Standard deviation of the difference between annular Zernike computation and Zernike by aberration index up to radial order 40 in extended FRINGE sequence. (L) double precision arithmetic. (R) Quad precision (`float128`)

Usage examples written in R are included in the package, and fringe analysis routines by default call the annular Zernike functions for wavefront fitting of annular apertures. Some care was taken to avoid having to link to libraries with restrictive licenses.

References

[1] Torben B Andersen. Efficient and robust recurrence relations for the zernike circle polynomials and their derivatives in cartesian coordinates. *Optics Express, Vol. 26, Issue 15, pp. 18878-18896*, 26: 18878–18896, 7 2018. ISSN 1094-4087. doi: 10.1364/OE.26.018878.

[2] I. Bogaert. Iteration-free computation of gauss–legendre quadrature nodes and weights. *https://doi.org/10.1137/140954969*, 36, 5 2014. ISSN 10957200. doi: 10.1137/140954969. URL https://epubs.siam.org/doi/10.1137/140954969.

[3] I Bogaert. Fast gauss-legendre quadrature rules download — sourceforge.net, 2014. URL https://sourceforge.net/projects/fastgausslegendrequadrature/.

[4] Max Born and Emil Wolf. *Principles of Optics*. Cambridge University Press, 7 edition, 1999. ISBN 052164221.

[5] Olver et al. Nist digital library of mathematical functions, 2022. URL https://dlmf.nist.gov/18.2#iv.

[6] Walter Gautschi. On generating orthogonal polynomials. *https://doi.org/10.1137/0903018*, 3:289–317, 5 2012. ISSN 0196-5204. doi: 10.1137/0903018. URL https://epubs.siam.org/doi/10.1137/0903018.

[7] Virendra N Mahajan. Zernike annular polynomials for imaging systems with annular pupils. *J. Opt. Soc. Am.*, 71:75–85, 1981. ISSN 0030-3941. doi: 10.1364/JOSA.71.000075. URL http://www.opticsinfobase.org/abstract.cfm?URI=josa-71-1-75.

[8] Virendra N Mahajan. Zernike annular polynomials and optical aberrations of systems with annular pupils. *Applied optics*, 33:8125–8127, 1994. ISSN 1084-7529. doi: 10.1364/AO.33.008125.

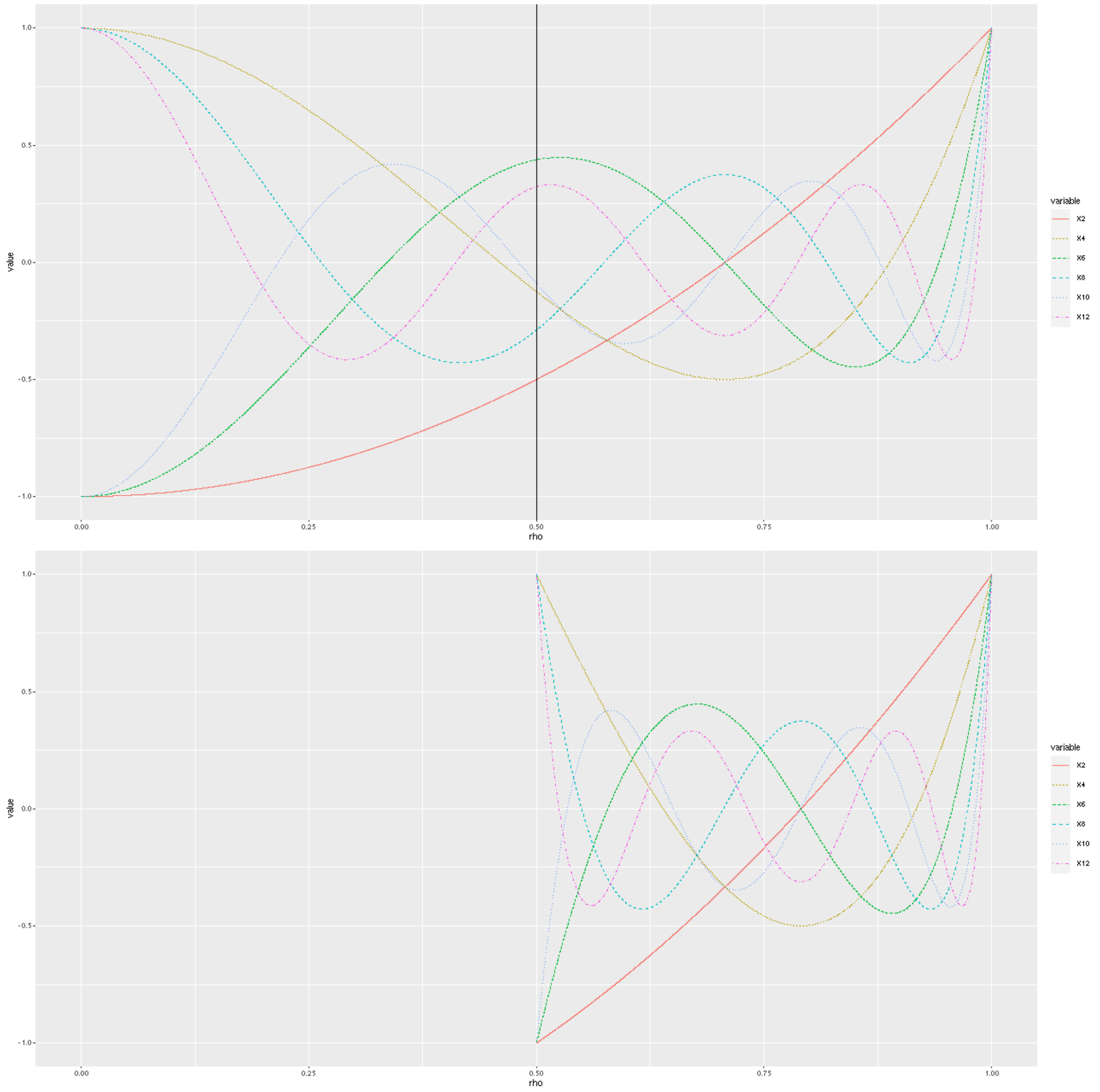

Figure 5: (T) radial Zernike polynomials for $m = 0, n = 2, \ldots, 12$. (B) the corresponding radial Zernike annular polynomials for $\epsilon = 0.5$

[9] Aluizio Prata and W. V. T. Rusch. Algorithm for computation of zernike polynomials expansion coefficients. *Applied Optics*, 28:749, 2 1989. ISSN 0003-6935. doi: 10.1364/AO.28.000749.

[10] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007. ISBN 0521880688. URL http://www.amazon.com/Numerical-Recipes-3rd-Scientific-Computing/dp/0521880688/ref=sr_1_1?ie=UTF8&s=books&qid=1280322496&sr=8-1.

[11] Barmak Shakibaei and Raveendran Paramesran. Recursive formula to compute zernike radial polynomials. *Optics Letters*, 38:2487, 7 2013. ISSN 0146-9592. doi: 10.1364/OL.38.002487.

[12] Berge Tatian. Aberration balancing in rotationally symmetric lenses. *J Opt Soc Am*, 64:1083–1091,

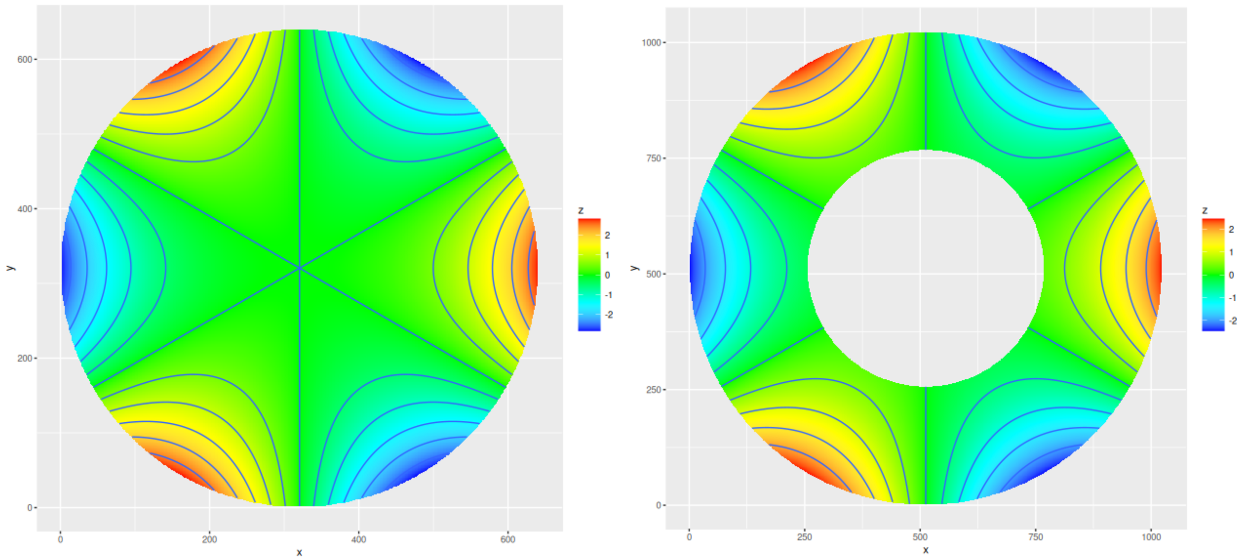

Figure 6: Trefoil: (L) on circle. (R) on annulus with $\epsilon = 0.5$

1974. ISSN 0030-3941. doi: 10.1364/JOSA.64.001083.