VANDERMONDE WITH ARNOLDI

PABLO D. BRUBECK, YUJI NAKATSUKASA, AND LLOYD N. TREFETHEN*

Abstract. Vandermonde matrices are exponentially ill-conditioned, rendering the familiar "polyval(polyfit)" algorithm for polynomial interpolation and least-squares fitting ineffective at higher degrees. We show that Arnoldi orthogonalization fixes the problem. This amounts to on-the-fly construction of discrete orthogonal polynomials by Stieltjes orthogonalization.

Key words. interpolation, least-squares, Vandermonde matrix, Arnoldi, polyval, polyfit, Fourier extension

AMS subject classifications. 41A05, 65D05, 65D10

1. Introduction. Fitting polynomials to data by means of Vandermonde matrices is a notoriously unstable algorithm. In this note we show how the problem can be fixed by coupling the Vandermonde construction with Arnoldi orthogonalization.

Let x be a column vector of m distinct numbers and f a column vector of m data values. Given $n \le m-1$, we wish to compute the polynomial

$$p(x) = \sum_{k=0}^{n} c_k x^k \tag{1.1}$$

with coefficients defined by a system of equations $Ac \approx f$,

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & \cdots & x_2^n \\ \vdots & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} \approx \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}. \tag{1.2}$$

If $n = m-1$, A is square and $\approx$ is interpreted as equality: p interpolates the data. If $n < m-1$, A has more rows than columns, and the system is interpreted in the least-squares sense. Regardless of n, A is of full rank since the points x_j are distinct, so c is uniquely defined. At least this is true mathematically, though in cases of interest, c may be nonunique numerically.

Once (1.2) has been solved, the evaluation of p at a set of points $s_1, \ldots, s_M$ amounts to the computation of the matrix-vector product

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix} = \begin{pmatrix} 1 & s_1 & \cdots & s_1^n \\ 1 & s_2 & \cdots & s_2^n \\ \vdots & \vdots & & \vdots \\ 1 & s_M & \cdots & s_M^n \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}. \tag{1.3}$$

2. Example 1: interpolation in Chebyshev points. Consider polynomial interpolation of $f(x) = 1/(1+25x^2)$ in $n+1$ Chebyshev points $x_j = \cos(j\pi/n)$, $0 \le j \le n$. This is a well-conditioned problem, satisfying $\|p\|/\|f\| < 1 + (2/\pi)\log(n+1)$

*brubeckmarti@maths.ox.ac.uk, nakatsukasa@maths.ox.ac.uk, and trefethen@maths.ox.ac.uk, Mathematical Institute, University of Oxford, Oxford, OX2 6GG, UK.

in the ∞-norm [26, Thm. 15.2]. Yet if we apply (1.2) and (1.3) as written, the result is a failure. Specifically, we execute in MATLAB the codes

```
function c = polyfit(x,f,n)
A = x.^(0:n);
c = A\f;

function y = polyval(c,s)
n = length(c)-1;
B = s.^(0:n);
y = B*c;
```

where s is a vector of 1000 points in $[-1,1]$. The left column of images in Figure 2.1 shows the results. For $n = 80$, the interpolant is plainly inaccurate, and the plot of error against n shows that after $n = 40$ or so, only 1–3 digits of accuracy are achieved. (The errors vary widely in the range about 10^{-4} to 10^{-2}, and the reader will very likely see different details on his or her own machine. If one does the same experiment with the versions of `polyfit` and `polyval` provided in MATLAB, the results are much worse, a consequence of the order of the columns of the Vandermonde matrix being reversed.)

On the other hand suppose we execute the following alternative codes, which will be explained in Section 4. The "A" stands for Arnoldi.

```
function [d,H] = polyfitA(x,f,n)
m = length(x);
Q = ones(m,1);
H = zeros(n+1,n);
for k = 1:n
    q = x.*Q(:,k);
    for j = 1:k
        H(j,k) = Q(:,j)'*q/m;
        q = q - H(j,k)*Q(:,j);
    end
    H(k+1,k) = norm(q)/sqrt(m);
    Q = [Q q/H(k+1,k)];
end
d = Q\f;

function y = polyvalA(d,H,s)
M = length(s);
W = ones(M,1);
n = size(H,2);
for k = 1:n
    w = s.*W(:,k);
    for j = 1:k
        w = w - H(j,k)*W(:,j);
    end
    W = [W w/H(k+1,k)];
end
```

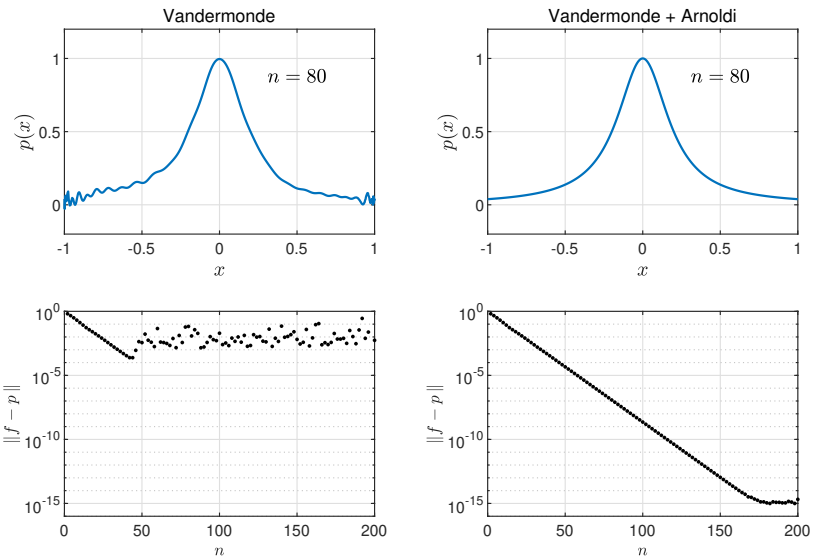

FIG. 2.1. *On the left, the degree n Chebyshev interpolant to $f(x) = 1/(1 + 25x^2)$ computed unstably by direct application of* (1.2) *and* (1.3) *via the codes* `polyfit` *and* `polyval` *for $n = 80$ (above) and its error for even values of n from 2 to 200 (below). (The results computed by the MATLAB versions of* `polyfit` *and* `polyval` *would be worse.) On the right, the same computations with the Arnoldi-based codes* `polyfitA` *and* `polyvalA`.

```
y = W*d;
```

The right column of Figure 2.1 shows the results, now converging cleanly down to machine precision. For this problem, other stable algorithms are also known involving the barycentric interpolation formula or the computation of Chebyshev series coefficients of the polynomial interpolant [6, 18, 24, 26].

3. Example 2: Least-squares approximation on two intervals. Our second example moves from interpolation to least-squares. This time, the problem is to approximate the function $\text{sign}(x)$ on $[-1, -1/3] \cup [1/3, 1]$ by a polynomial of degree n. We discretize the domain by 500 equispaced points in each interval and call `polyfit/polyval` and `polyfitA/polyvalA` to compute least-squares fits. Here as in most of our examples, we are working on discrete sets chosen to have enough points to approximate a continuum effectively. (If 500 is increased to 5000, the results look the same.) There are interesting and important questions associated with how to optimize such discretizations, which generally involves clustering of sample points near endpoints and corners, but we do not discuss such matters in this paper.

The computed results are displayed in Figure 3.1 in the same pattern as in Figure 2.1, again revealing a great disparity between algorithms. A new feature, however, is that the error with `polyfit/polyval` now goes down to order $O(10^{-5})$ before stagnating at that level. Ill-conditioned least-squares computations, at least as carried out by the MATLAB backslash command, are often sufficiently regularized that the results may look very good in the "eyeball norm." For some purposes, this may be all the accuracy one needs, but the figure shows that it falls far short of what is achievable.

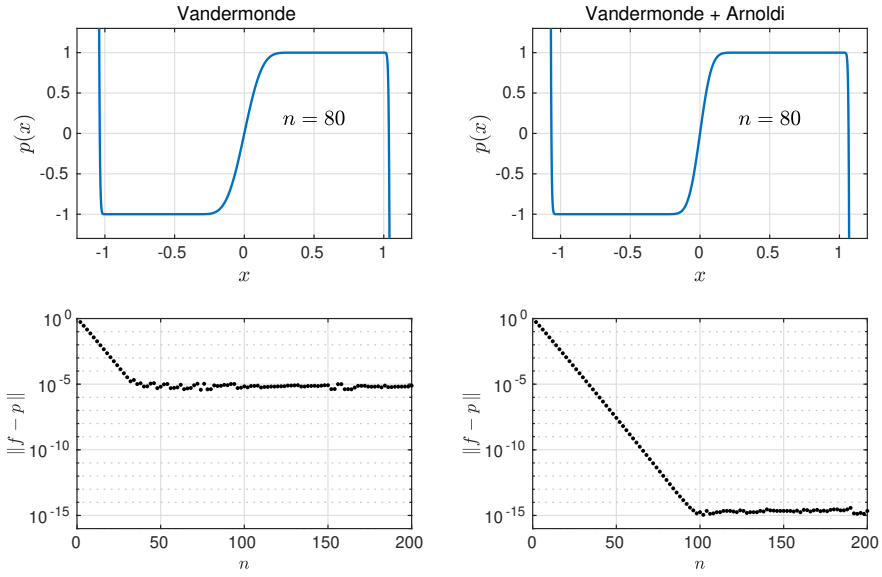

FIG. 3.1. *Images as in Fig.* 2.1 *but now for a least-squares problem: polynomial fitting to* sign(x) *on* 500 *equispaced points each in the two intervals* $[-1, -1/3]$ *and* $[1/3, 1]$. *The unstable algorithm stagnates at* 5 *digits of accuracy, which is enough that to the eye, the computation appears successful.*

4. Vandermonde with Arnoldi. We now explain the Vandermonde with Arnoldi algorithm.

First, a word about Vandermonde without Arnoldi, which solves the equation

$$Ac \approx f. \tag{4.1}$$

When the code `polyfit` of Section 2 is executed, exactly what happens depends on what lies behind the backslash operator—which in the world of MATLAB is a time-dependent business. If A is rectangular, the algorithm will probably be some variant of QR factorization, with regularizing effects as just mentioned (see [1] and also Remark 2.5 of [3]). If A is square, the algorithm will probably be Gaussian elimination with partial pivoting.

The introduction of Arnoldi orthogonalization follows a standard idea in the area of Krylov matrix iterations [13]. The columns $1, x, x^2, \dots$ of (1.2) can be regarded as vectors $q_0, Xq_0, X^2q_0, \dots$, where $q_0 = (1, \dots, 1)^T$ and $X = \mathrm{diag}(x_1, \dots, x_m)$. In the Arnoldi process, rather than computing these vectors as written and then orthogonalizing the result in a QR factorization, one orthogonalizes at each step to obtain a sequence of orthogonal vectors $q_0, q_1, q_2, \dots$ spanning the same spaces. The code `polyfitA` does this in a standard fashion, its only unusual feature being that q_k is normalized to have 2-norm $m^{1/2}$ rather than 1, so that the entries individually are of scale 1 (a matter of convenience). After n steps, orthogonal vectors $q_0, \dots, q_n$ and an $(n+1) \times n$ upper-Hessenberg matrix H have been computed such that

$$XQ_- = QH, \tag{4.2}$$

where Q is the $m \times (n+1)$ matrix with columns $q_0, \dots, q_n$ and Q_- is the same matrix without the final column. Equation (4.1) is now solved in its equivalent form

$$Qd \approx f, \tag{4.3}$$

and the resulting Qd is the m-vector of values $p(x_j)$ for the same polynomial p of degree n. The coefficient vectors in the two different representations are related mathematically by

$$d = Rc \tag{4.4}$$

where $R = Q^T A/m$ is the upper-triangular matrix such that $A = QR$.

In the usual Arnoldi application, further operations would now be carried out on Q and H for purposes such as estimating eigenvalues or solving systems of equations. For us, however, all that matters is the matrix H. The entries of column $k+1$ of H are the coefficients employed by `polyfitA` to orthogonalize Xq_k against $q_0, \ldots, q_k$. In `polyvalA`, the same operations are applied with a new operator $S = \mathrm{diag}(s_1, \ldots, s_M)$ in place of the original $X = \mathrm{diag}(x_1, \ldots, x_m)$. We call the resulting matrices W_- and W, satisfying

$$SW_- = WH. \tag{4.5}$$

Note that the columns of W_- and W are not orthogonal, merely (in many cases) approximately so. If Wd is now computed, the result is the M-vector of values $p(s_j)$.

All this is closely related to orthogonal polynomials, where the Arnoldi process is known as Stieltjes orthogonalization [9, 11, 12, 17, 24]. The entries of H are the recurrence coefficients for a sequence of polynomials orthogonal with respect to the uniform discrete measure on $\{x_1, \ldots, x_m\}$. If x is a discretization of a real or complex set Γ, they can be expected to approximate the recurrence coefficients for orthogonal polynomials defined by a continuous inner product on Γ. This is what makes the Vandermonde with Arnoldi algorithm so effective: it is constructing an approximation to orthogonal polynomials, hence a well-conditioned basis. The ill-conditioning of the Vandermonde basis, by contrast, has an elementary explanation in cases where the points x_j are unequal in size. If $|x_j|$ varies with j, then the powers $|x_j^k|$ vary exponentially. This means that function information associated with smaller values of x_j will only be resolvable through exponentially large expansion coefficients, and accuracy will quickly be lost in floating point arithmetic. The wiggles for $|x| \approx 1$ in the first image of Fig. 2.1, for example, are the result of exponentially large coefficients of terms x^k being required in an attempt to resolve the function for values $|x| < 1$.

Like `polyfit`, `polyfitA` requires $O(mn^2)$ operations, though with a larger constant. Subsequent evaluations, however, cost just $O(Mn)$ operations with `polyval` but $O(Mn^2)$ with `polyvalA`.

Note that since the columns of Q are orthogonal with norm $m^{1/2}$, the last line of `polyfitA` could be replaced by `d = Q'*f/m`, a mathematically equivalent operation involving just $O(mn)$ instead of $O(mn^2)$ operations. We have not done this since experiments indicate that accuracy is impaired. (On the other hand, this becomes an extremely stable alternative if each column of Q is computed with not just one but two rounds of orthogonalization against previous columns.)

If the numbers x_j are real, then X is real symmetric and the recurrence reduces to three terms at each step: H is tridiagonal and Arnoldi is equivalent to Lanczos [13]. In principle this could reduce the operation counts to $O(mn)$ for `polyfitA` (less than for `polyfit`!) and $O(Mn)$ for `polyvalA`. There are probably applications for which it would be worthwhile to exploit this structure, but there is a risk of reduced accuracy since the columns of V computed numerically may lose orthogonality.

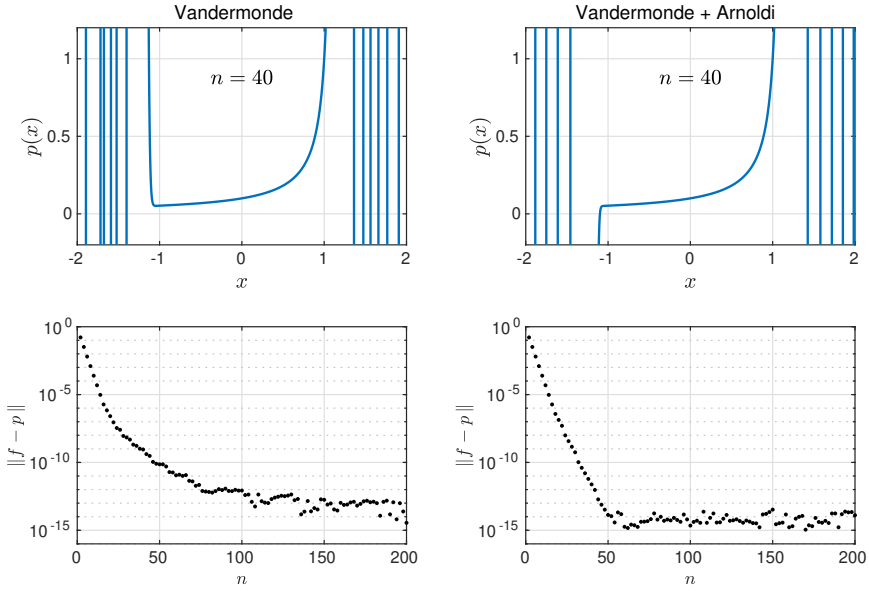

FIG. 5.1. *A Fourier extension example from* [1]*, with* $f(x) = 1/(10 - 9x)$ *approximated over* $[-1, 1]$ *by Fourier series scaled to the larger interval* $[-2, 2]$*. This is equivalent to approximation by powers* z^k *over just half of the unit circle, leading to exponential ill-conditioning of the Vandermonde matrix.*

5. Example 3: Fourier extension. Our third example comes from [1]. Suppose we wish to approximate $f(x) = 1/(10-9x)$ over the interval $[-1, 1]$ using Fourier series scaled to the larger interval $[-2, 2]$:

$$f(x) \approx \sum_{k=-n}^{n} c_k e^{ik\pi x/2}.$$

Since f is real, the coefficients will satisfy $c_{-k} = \overline{c}_k$, and we can write (with a revised definition of c_k)

$$f(x) \approx \mathrm{Re}\Big(\sum_{k=0}^{n} c_k e^{ik\pi x/2}\Big).$$

This one-dimensional problem is a model of Fourier extensions in higher dimensions, which can be used to approximate functions on irregular domains by embedding them in boxes. The difficulty is that the bases are exponentially ill-conditioned.

The Vandermonde structure is revealed if we define $z = e^{i\pi x/2}$, giving

$$f(x) \approx \mathrm{Re}\Big(\sum_{k=0}^{n} c_k z^k\Big), \tag{5.1}$$

or with $c_k = a_k + ib_k$,

$$f(z) \approx \sum_{k=0}^{n} \big(a_k \mathrm{Re} z^k - b_k \mathrm{Im} z^k\big). \tag{5.2}$$

Our aim is to find coefficients $\{a_k\}$ and $\{b_k\}$ such that (5.2) is satisfied in the least-squares sense in a set of m points on Γ, the right half of the unit circle, with $m \gg 2n+1$. We use 1000 points on Γ clustered at the endpoints in a Chebyshev distribution. The calculation is done by modifying (1.2) to the form

$$\text{(5.3)}\quad \text{Re}\begin{pmatrix} 1 & \cdots & z_1^n \\ 1 & \cdots & z_2^n \\ \vdots & & \vdots \\ 1 & \cdots & z_m^n \end{pmatrix}\begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} - \text{Im}\begin{pmatrix} z_1 & \cdots & z_1^n \\ z_2 & \cdots & z_2^n \\ \vdots & & \vdots \\ z_m & \cdots & z_m^n \end{pmatrix}\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \approx \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix}.$$

In `polyfit`, the final line is changed to

```
c = [real(A) imag(A(:,2:n+1))]\f;
c = c(1:n+1) - 1i*[0; c(n+2:2*n+1)];
```

and similarly in `polyfitA` with `d` and `Q` in place of `c` and `A`. We also now take the real parts of the results computed by `polyval` and `polyvalA`.

Figure 5.1 shows results in the same format as the previous two examples. The function being approximated, $f(x) = 1/(10-9x)$, has a pole in $[-2,2]$, a situation representative of applications of Fourier extension in cases where where one cannot assume analyticity beyond the boundary of the approximation domain. The Fourier extension appears to converge root-exponentially when implemented with `polyfit` and `polyval`; compare Fig. 5 of [1]. With `polyfitA` and `polyvalA`, it converges exponentially. We note that the success of this computation, and its close relationship with the previous example after the change of variables $z = e^{i\pi x/2}$, suggest that in the end there may not be much practical difference between Fourier extensions and approximations by ordinary algebraic polynomials.

6. Example 4: Laplace equation and conformal mapping. Our final example concerns the solution of the 2D Laplace equation by approximation of the solution by the real part of a complex polynomial [27]. To illustrate, let Ω be the region bounded by a Jordan curve Γ in the complex plane enclosing the point 0, and let g be the unique analytic function that maps Ω conformally onto the unit disk with $g(0) = 0$ and $g'(0) > 0$. We can write

$$\text{(6.1)}\qquad g(z) = ze^{h(z)} = ze^{u(z)+iv(z)}$$

where $h(z) = u(z) + iv(z)$ is an analytic function with $\text{Im}h(0) = 0$. Since $|g(z)| = 1$ for $z \in \Gamma$, we have $|h(z)| = -\log(|z|)$ for $z \in \Gamma$, that is,

$$\text{(6.2)}\qquad u(z) = -\log(|z|), \quad z \in \Gamma.$$

Thus the conformal mapping problem reduces to a Laplace problem: find a harmonic function u in Ω satisfying the boundary condition (6.2), let v be its harmonic conjugate with $v(0) = 0$, and then g is given by (6.1). Following [28, sec. 2], we compute h by approximating it by a polynomial:

$$\text{(6.3)}\qquad h(z) \approx \sum_{k=0}^{n} c_k z^k,$$

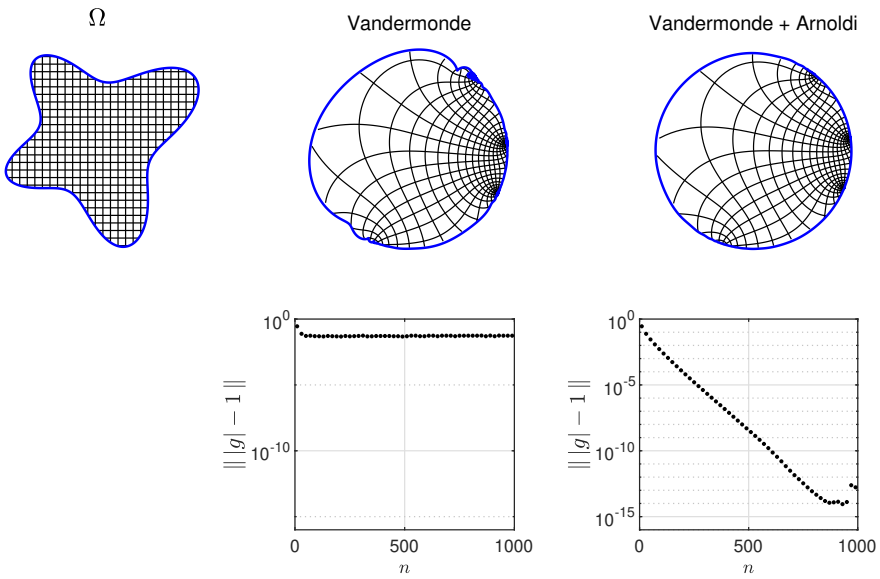

FIG. 6.1. *Conformal mapping of a blob onto the unit disk by the polynomial expansion method of* (6.1)–(6.4). *The two upper-right images correspond to* $n = 200$.

or if $c_k = a_k + ib_k$,

$$u(z) \approx \sum_{k=0}^{n} \big(a_k \mathrm{Re} z^k - b_k \mathrm{Im} z^k\big), \quad v(z) \approx \sum_{k=1}^{n} \big(b_k \mathrm{Re} z^k + a_k \mathrm{Im} z^k\big). \tag{6.4}$$

We find coefficients $\{a_k\}$ and $\{b_k\}$ such that (6.2) is satisfied in the least-squares sense in a set of m points on Γ with $m \gg 2n + 1$. This is done again by solving (5.3).

Figure 6.1 shows the result of such a computation involving a blob-shaped region Ω. The Vandermonde solution quickly stagnates, whereas Vandermonde with Arnoldi is successful. In this example as in the others, we have been careful to choose variables x and z with maximum absolute value equal to 1 so that the columns of the matrices (1.2) and (5.3) are uniformly scaled. Without such scaling, the Vandermonde approach without Arnoldi orthogonalization fails more dramatically, but that would perhaps be an unfair comparison.

7. Discussion. As we have mentioned, the combinination of Vandermonde with Arnoldi is not new: this is the Stieltjes procedure in the theory of orthogonal polynomials, and its numerical advantages have been advocated at least as far back as Forsythe in 1957 [10]. The work of Stylianopoulos has focused particularly compellingly on the Arnoldi idea, which he backs up by stability estimates related to leading coefficients of orthogonal polynomials [16, 22, 23]. Among undoubtedly many other related publications we note [2, 7, 8, 15, 19, 21, 25, 29]. Themes of these works include discrete and continuous orthogonal polynomials on the real line and in the complex plane, quadrature formulas, and the exploitation of three-term recurrence relations where possible. (*Bergman polynomials* are defined by orthogonality with respect to area measure and *Szegő polynomials* by orthogonality over subsets of the unit circle.) However, the emphasis in the literature is often on construction of orthogonal

polynomials on a continuous domain as an end in itself. In this note we have aimed to show that instead, discrete orthogonal polynomials can be calculated on the fly as a means to enable further computations. They are only approximately orthogonal on a continuum, but in a numerical application, this is all one needs.

There is an interesting literature on Vandermonde matrices, which are always exponentially ill-conditioned unless the nodes are uniformly distributed on the unit circle as in the well known case of the Fast Fourier Transform matrix [4, 5, 20]. The method presented here bypasses this discussion, since it bypasses the use of Vandermonde matrices.

For the specific application of polynomial interpolation, we have not found the Vandermonde with Arnoldi technique in the literature. Here the use of the barycentric formula is the established stable technique [6, 18], as mentioned at the end of Section 2.

Perhaps most important is the rectangular case, the problem of least-squares fitting with an ill-conditioned basis. As emphasized by Adcock and Huybrechs [1], problems of this nature are widespread. Our own awareness of the importance of Arnoldi orthogonalization came through the development of "lightning solvers" for the Laplace equation. Here a polynomial term plays just a supporting role in the computation, and in writing [14], we did not recognize that our computations were sometimes stagnating because this term was being treated by Vandermonde without Arnoldi. The `laplace` code subsequently made available at `people.maths.ox.ac.uk/trefethen/` remedies the problem by including Arnoldi orthogonalization, as does the newly introduced Chebfun code `conformal(...,'poly')`.

Arnoldi stabilization of Vandermonde computations has many applications beyond those we have illustrated of interpolation, least-squares fitting, Fourier extension, and solution of the Laplace equation. In work not yet published, we have applied these ideas to the biharmonic equation for problems of Stokes flow, where Hermite (confluent) interpolation conditions arise in connection with Neumann boundary conditions; to multivariate polynomial and Fourier approximation on arbitrary domains; and to computation of minimal polynomials on complex sets defined by the minimax rather than least-squares criterion via Lawson iteration (iteratively reweighted least-squares), a problem related to the computation of capacities. Other extensions under consideration involve the Laplace eigenvalue problem and the Helmholtz equation, cases where the successive powers of a variable give way to a sequence of Hankel functions. In any application where a matrix of Vandermonde or closely related form may appear, it seems likely that introducing Arnoldi orthogonalization may be a good idea.

Acknowledgments. We thank Bernd Beckermann, Jean-Paul Berrut, Timo Betcke, Amit Hochman, Lyle Ramshaw, Lothar Reichel, and Olivier Sète for helpful comments.

REFERENCES

[1] B. Adcock and D. Huybrechs, *Frames and numerical approximation,* SIAM Rev., 61 (2019), 443–473.

[2] A. P. Austin, et al., *Multivariate rational approximation,* arXiv:1912.02272, 2020.

[3] A. H. Barnett, G. R. Marple, S. Veerapeneni, and L. Zhao, *A unified integral equation scheme for doubly periodic Laplace and Stokes boundary value problems in two dimensions,* Commun. Pure Appl. Math., 71 (2018), 2334–2380.

[4] B. BECKERMANN, *The condition number of real Vandermonde, Krylov and positive definite Hankel matrices,* Numer. Math., 85 (2000), pp. 553–577.
[5] B. BECKERMANN, G. H. GOLUB, AND G. LABAHN, *On the numerical condition of a generalized Hankel eigenvalue problem,* Numer. Math., 106 (2007), pp. 41–68.
[6] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation,* SIAM Rev., 46 (2004), pp. 501–517.
[7] T. BETCKE, *The generalized singular value decomposition and the method of particular solutions,* SIAM J. Sci. Comp., 30 (2008), pp. 1278–1295.
[8] Å. BJÖRCK AND V. PEREYRA, *Solution of Vandermonde systems of equations,* Math. Comput., 24 (1970), pp. 893–903.
[9] P. J. DAVIS, *Interpolation and Approximation,* Dover, 1975.
[10] G. E. FORSYTHE, *Generation and use of orthogonal polynomials for data-fitting with a digital computer,* J. SIAM, 5 (1957), pp. 74–88.
[11] D. GAIER, *Lectures on Complex Approximation,* Birkhäuser, 1987.
[12] W. GAUTSCHI, *Orthogonal Polynomials: Computation and Approximation,* Oxford U. Press, 2004.
[13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations, 4th ed.,* Johns Hopkins, 2013.
[14] A. GOPAL AND L. N. TREFETHEN, *Solving Laplace problems with corner singularities via rational functions,* SIAM J. Numer. Anal., 57 (2019), pp. 2074–2094.
[15] W. B. GRAGG AND L. REICHEL, *On the application of orthogonal polynomials to the iterative solution of linear systems of equations with indefinite or non-hermitian matrices,* Lin. Alg. Applics., 88/89 (1987), pp. 349–371.
[16] B. GUSTAFSSON, M. PUTINAR, E. B. SAFF, AND N. STYLIANOPOULOS, *Bergman polynomials on an archipelago: estimates, zeros and shape reconstruction,* Adv. Math. 222 (2009), pp. 1405–1460.
[17] P. HENRICI, *Applied and Computational Complex Analysis, v. 3,* Wiley, 1986.
[18] N. J. HIGHAM, *The numerical stability of barycentric Lagrange interpolation,* IMA J. Numer. Anal. 24 (2004), pp. 547–556.
[19] R. PACHÓN, P. GONNET, AND J. VAN DEUN, *Fast and stable rational interpolation in roots of unity and Chebyshev points,* SIAM J. Numer. Anal., 50 (2012), pp. 1713–1734.
[20] V. Y. PAN, *How bad are Vandermonde matrices?,* SIAM J. Matrix Anal. Applics., 37 (2016), pp. 676–694.
[21] O. SÈTE AND J. LIESEN, *Properties and examples of Faber–Walsh polynomials,* Comput. Meth. Funct. Th., 17 (2017), pp. 151–177.
[22] N. STYLIANOPOULOS, *An Arnoldi Gram–Schmidt process and Hessenberg matrices for orthonormal polynomials,* slides from talk at Banff International Research Station, `http://www.mas.ucy.ac.cy/~nikos/Banff2010.pdf`.
[23] N. STYLIANOPOULOS, *Strong asymptotics for Bergman polynomials over domains with corners and applications,* Constr. Approx., 38 (2013), pp. 59–100.
[24] G. SZEGŐ, *Orthogonal Polynomials,* Amer. Math. Soc., 1975.
[25] J. M. TAYLOR, *The condition of gram matrices and related problems,* Proc. Roy. Soc. Edinb. A, 80 (1978), pp. 45–56.
[26] L. N. TREFETHEN, *Approximation Theory and Approximation Practice, Extended Edition,* SIAM, 2019.
[27] L. N. TREFETHEN, *Series solution of Laplace problems,* ANZIAM J., 60 (2018), pp. 1–26.
[28] L. N. TREFETHEN, *Numerical conformal mapping with rational functions,* Comp. Meth. Funct. Th., to appear.
[29] X. YE, Y. XI, AND Y. SAAD, *Preconditioning via GMRES in polynomial space,* preprint, 2019.