Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Linear Algebra and its Applications 417 (2006) 382–396
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials[☆]

James Demmel[a], Plamen Koev[b,*,1]

[a]*Computer Science Division and Mathematics Department, University of California, Berkeley, CA 94720, United States*

[b]*Department of Mathematics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States*

Accepted 16 September 2005
Available online 10 November 2005
Submitted by R.A. Brualdi

Abstract

We present a new $O(n^3)$ algorithm for computing the SVD of an $n \times n$ polynomial Vandermonde matrix $V_P = [P_{i-1}(x_j)]$ to high relative accuracy in $O(n^3)$ time. The P_i are orthonormal polynomials, deg $P_i = i$, and x_j are complex nodes. The small singular values of V_P can be arbitrarily smaller than the largest ones, so that traditional algorithms typically compute them with no relative accuracy at all.

We show that the singular values, even the tiniest ones, are usually well-conditioned functions of the data x_j, justifying this computation.

We also explain how this theory can be extended to other polynomial Vandermonde matrices, involving polynomials that are not orthonormal or even orthogonal.

[☆] This material is based in part upon work supported by the LLNL Memorandum Agreement No. B504962 under DOE Contract No. W-7405-ENG-48, DOE Grants No. DE-FG03-94ER25219, DE-FC03-98ER25351 and DE-FC02-01ER25478, NSF Grant No. ASC-9813362, and Cooperative Agreement No. ACI-9619020.

* Corresponding author.
E-mail addresses: demmel@cs.berkeley.edu (J. Demmel), plamen@math.mit.edu (P. Koev).

[1] This work was performed while the author was a graduate student at the Mathematics Department at the University of California, Berkeley.

0024-3795/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2005.09.014

Keywords: Vandermonde matrix; High relative accuracy; Singular value decomposition; Orthogonal polynomial

1. Introduction

A *polynomial Vandermonde matrix involving orthonormal polynomials* is a matrix of the form:

$$V_P(x) = \begin{bmatrix} P_0(x_1) & P_1(x_1) & \dots & P_{n-1}(x_1) \\ P_0(x_2) & P_1(x_2) & \dots & P_{n-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ P_0(x_n) & P_1(x_n) & \dots & P_{n-1}(x_n) \end{bmatrix},$$

where P_i are real polynomials, orthonormal on some interval $[a, b]$, $\deg P_i = i$, and $x = (x_1, x_2, \dots, x_n)$ is a complex n-vector.

The orthogonal polynomials are useful in the solution of various mathematical and physical problems and provide a natural way to solve, expand, and interpret solutions to many types of important differential equations [1,15].

Let $V_P = W \cdot \Sigma \cdot Z^*$ be the singular value decomposition (SVD) of $V_P(x)$, where W and Z are unitary,[2] $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$, and $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$ are the singular values.

Our first major contribution in this paper is a new algorithm for computing the SVD of $V_P(x)$ accurately and efficiently. By *accurately* we mean [3,4]:

- The error $|\sigma_i - \hat{\sigma_i}|$ in each computed singular value $\hat{\sigma_i}$ is bounded by $\mathrm{O}(\epsilon)\sigma_i$, where ϵ is the machine precision, i.e., the relative error is small;
- The angle $\theta(w_i, \hat{w}_i)$ between the true left singular vector w_i, corresponding to a simple singular value σ_i, and the computed singular vector $\hat{w}_i$ is bounded by $\mathrm{O}(\epsilon)/\mathrm{relgap}_i$, where $\mathrm{relgap}_i = \min_{i \neq j} |\sigma_j - \sigma_i|/(\sigma_i + \sigma_j)$ is the relative gap between σ_i and the nearest other singular value. An analogous statement holds for the computed right singular vectors $\hat{z}_i$.

By *efficiently* we mean that the cost is $\mathrm{O}(n^3)$, *independent of the condition number* $\kappa(G) = \|G\| \cdot \|G^{-1}\|$.

In contrast, conventional SVD algorithms deliver only high *absolute* accuracy, meaning the tiny singular values may be lost to roundoff. If we attempted to compute the SVD to high relative accuracy using conventional algorithms, we would have to use extended precision arithmetic, whose precision (and cost) grows with $\kappa(G)$.

[2] We use $W\Sigma Z^{\mathrm{T}}$ for the SVD instead of the traditional $U\Sigma V^{\mathrm{T}}$; we use the letters V and U for other purposes.

We now briefly survey the properties of orthogonal polynomials. A set of polynomials $P = \{P_i\}_{i=0}^n$ is called *orthogonal* on an interval $[a, b]$ if for a *weight function* $w(t)$ ($w(t) \geqslant 0$, $t \in [a, b]$), we have $\langle P_i, P_j \rangle = 0$, $i \neq j$, where

$$\langle f, g\rangle \equiv \int_a^b w(t) f(t) g(t)\, \mathrm{d}t.$$

Let

$$c_i \equiv \langle P_i, P_i \rangle, \quad i = 0, 1, 2, \ldots \tag{1}$$

If $c_i = 1$ for all i, then P is a set of *orthonormal* polynomials. A set of orthogonal polynomials can always be normalized: if $\{P_i\}_{i=0}^\infty$ are orthogonal, then $\{P_i/\sqrt{c_i}\}_{i=0}^\infty$ are orthonormal. All orthogonal polynomials satisfy a *three-term recurrence*:

$$\begin{aligned} &P_0(x) = a_0, \quad P_1(x) = d_0 x + b_0, \\ &P_{i+1}(x) = d_i (x - b_i) P_i(x) - a_i P_{i-1}(x), \end{aligned} \tag{2}$$

where $a_i, d_i \neq 0$ for all i. The roots of P_i are real and simple.

The properties of some well-known classes of orthogonal polynomials are given in Table 1 [1,15].

For our error analysis we use the standard model of floating point arithmetic, in which we assume only that the relative error of any arithmetic operation is small [11, Section 2.2]:

$$\mathrm{fl}(c \odot d) = (c \odot d)(1 + \delta),$$

where $\odot \in \{+, -, *, /\}$ and $|\delta| < \epsilon$ for some fixed ϵ, called *machine precision*. We also assume that no underflow or overflow occurs. This model implies that products, quotients, and sums of like-signed quantities can be computed accurately (i.e., with low relative error), but expressions involving cancellation may not be (for example, the sum of three numbers can provably not be computed accurately in this model [6]). However, if c and d are inputs (and so can be considered exact) then $\mathrm{fl}(c \pm d) = (c \pm d)(1 + \delta)$ is computed accurately.

The trick in achieving high relative accuracy is to avoid subtractions of approximate intermediate results in those parts of the algorithm where subtractive cancellation may lead to loss of significant digits.

Table 1
Properties of some orthogonal polynomials

Type	Interval	$w(x)$	c_j	d_j	b_j	a_j
Chebyshev, first kind	$[-1, 1]$	$1/\sqrt{1-x^2}$	π, $c_0 = \frac{\pi}{2}$	2, $d_0 = 1$	0	1
Chebyshev, second kind	$[-1, 1]$	$\sqrt{1-x^2}$	$\frac{\pi}{2}$	2	0	1
Legendre ($P_j(1) \equiv 1$)	$[-1, 1]$	1	$\frac{2}{2j+1}$	$\frac{2j+1}{j+1}$	0	$\frac{j}{j+1}$
Laguerre	$[0, \infty)$	e^{-x}	1	$-\frac{1}{j+1}$	$2j+1$	$\frac{j}{j+1}$
Hermite	$(-\infty, \infty)$	e^{-x^2}	$\sqrt{\pi}2^j j!$	2	0	$2j$

We compute the SVD of $V_P(x)$ to high relative accuracy as follows. We first write $V_P(x) = C \cdot Q$ as a product of the scaled Cauchy matrix

$$C = \text{diag}(h_1, \ldots, h_n) \cdot \left[\frac{1}{x_i - y_j}\right]_{i,j=1}^{n} \cdot \text{diag}(g_1, \ldots, g_n) \tag{3}$$

and an orthogonal matrix Q (see Section 2 for details). This decomposition is a straightforward consequence of the Lagrange interpolation formula and the discrete orthogonality property of the orthonormal polynomials P_i. Then, following the idea of Demmel [3], we compute the decomposition $C = LDU$ resulting from Gaussian elimination with complete pivoting (GECP). We exploit the structure of C to perform GECP using only multiplications, divisions, and differences of initial data, thus preserving the relative accuracy. The resulting decomposition $V_P(x) = L \cdot D \cdot (UQ)$ is a *rank-revealing decomposition* (RRD) [4], i.e., the matrices L and UQ are well conditioned and computed with small norm error, and each entry of D is computed with small relative error. Finally, given this RRD of $V_P(x)$, we apply Algorithm 3.1 from [4] to compute the SVD to high relative accuracy.

The case $x_i = y_j$ in (3) turns out to be a removable singularity. Also, Algorithm 2.1 suffers no instability in computing the decomposition $C = LDU$ in that case. This is our second major contribution in this paper. In particular, we answer an open question at the end of Section 5 of [3]: We do not need a high precision table of the roots of unity in order to compute an accurate SVD of the ordinary Vandermonde matrix, $V = [x_i^{j-1}]_{i,j=1}^{n}$ when some of the nodes x_i coincide with the (floating point representations of the) roots of unity.

We can compute the SVD of $V_P(x)$ to high relative accuracy even when the P_i are not orthonormal or not even orthogonal. In particular, this is the case when P are the (unnormalized) Chebyshev, Legendre, or Laguerre orthogonal polynomials, or the monomials x^i. We describe the conditions under which this is possible in Section 3.

In Section 4, we analyze the computational problems in computing the roots of P_n and the Christoffel numbers.

Our third major contribution is to show that the singular values of $V_P(x)$, even the tiniest ones, are well conditioned functions of the data (see Section 5).

Finally, in Section 6 we present numerical experiments.

2. Main algorithm

In this section we present our main algorithm for computing the SVD of $V_P(x)$ to high relative accuracy. We first show that $V_P(x)$ can be written as a product of a scaled Cauchy matrix and an orthogonal matrix.

Let $P = \{P_i\}_{i=0}^{\infty}$ be a set of polynomials, deg $P_i = i$, orthonormal on some real interval $[a, b]$. Let $y = (y_1, y_2, \ldots, y_n)$ be a vector of n pairwise distinct complex numbers.

We use the Lagrange interpolation formula to interpolate the value of $P_l(x_i)$ at the points $y_1, y_2, \ldots, y_n$:

$$P_l(x_i) = \sum_{j=1}^{n} E_{ij} P_l(y_j),$$

where

$$E_{ij} = \prod_{k=1,\ k\neq j}^{n} \frac{x_i - y_k}{y_j - y_k}$$

for $i, j = 1, 2, \ldots, n$. Therefore

$$V_P(x) = E \cdot V_P(y). \tag{4}$$

Let $y_1, y_2, \ldots, y_n$ now be the distinct roots of P_n, and let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the Christoffel numbers. Define $\Lambda \equiv \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$.

We use the *discrete orthogonality property* [8] of the orthonormal polynomials:

$$\sum_{i=1}^{n} \lambda_i P_m(y_i) P_k(y_i) = \delta_{mk}$$

for all $k, m < n$. It implies that the matrix

$$Q \equiv \Lambda^{1/2} \cdot V_P(y) \tag{5}$$

is orthogonal.

Let $C \equiv E \cdot \Lambda^{-1/2}$, i.e.,

$$C_{ij} = \lambda_j^{-1/2} \cdot \prod_{k\neq j} \frac{x_i - y_k}{y_j - y_k} \tag{6}$$

for $i, j = 1, 2, \ldots, n$. The matrix C is a *scaled Cauchy matrix*:[3]

$$C = \mathrm{diag}(h_1, \ldots, h_n) \cdot \left[\frac{1}{x_i - y_j}\right]_{i,j=1}^{n} \cdot \mathrm{diag}(g_1^{-1}, \ldots, g_n^{-1}), \tag{7}$$

where $h_i = \prod_{k=1}^{n}(x_i - y_k)$ and $g_i = \lambda_i^{1/2} \prod_{k=1,\ k\neq i}^{n}(y_i - y_k)$, $i = 1, 2, \ldots, n$.

Now from (4),

$$V_P(x) = E \cdot V_P(y) = E \cdot \Lambda^{-1/2} \cdot Q = C \cdot Q. \tag{8}$$

Namely, $V_P(x)$ equals the product of the scaled Cauchy matrix C and the orthogonal matrix Q.

The first step in computing the SVD of $V_P(x)$ is to compute the LDU decomposition of C resulting from GECP to high relative accuracy. We follow the idea of Demmel [3, Algorithm 3]. Let $C^{(k)}$ be the kth Schur complement of C. We use the traditional update formula in GECP only when $C_{ij}^{(k-1)} = 0$:

[3] The expression (7) clearly reveals the scaled Cauchy structure of C, but contains an unnecessary removable singularity at $x_i = y_j$; therefore we use (6) in our computations.

$$C_{ij}^{(k)} = C_{ij}^{(k-1)} - \frac{C_{ik}^{(k-1)} C_{kj}^{(k-1)}}{C_{kk}^{(k-1)}} = -\frac{C_{ik}^{(k-1)} C_{kj}^{(k-1)}}{C_{kk}^{(k-1)}}. \tag{9}$$

When $C_{ij}^{(k-1)} \neq 0$, we use the equivalent expression (see, e.g., [3, Section 4], or (12) below):

$$C_{ij}^{(k)} = C_{ij}^{(k-1)} \frac{(x_i - x_k)(y_j - y_k)}{(x_i - y_k)(y_j - x_k)}. \tag{10}$$

Both expressions (9) and (10) involve only multiplications, divisions, and subtractions of initial data, and thus preserve the relative accuracy.

The use of the update (9) when $C_{ij}^{(k-1)} = 0$ is the only difference between our Algorithm 2.1 below and Algorithm 3 from [3]. This small modification makes our algorithm stable even when $x_i = y_j$ for some i and j as we now prove. This is our second major contribution in this paper.

Lemma 2.1. *Let $C^{(k)}$ be the* k*th Schur complement of the matrix C defined in* (6). *Let $C_{kk}^{(k-1)} \neq 0$. If $C_{ij}^{(k-1)} \neq 0$ for some $i > k,\ j > k$, then $x_i \neq y_k$ and $x_k \neq y_j$. Thus the formula* (10) *may be used.*

Proof. We use the well-known formula for the determinant of the Cauchy matrix

$$\det \left[\frac{1}{x_i - y_j} \right]_{i,j=1}^{n} = \frac{\prod_{i<j}(x_j - x_i)(y_i - y_j)}{\prod_{i,j}(x_i - y_j)} \tag{11}$$

(see, e.g., [3]). Now $C_{kk}^{(k-1)} \neq 0$ implies $\det C(1:k, 1:k) \neq 0$.[4] From (7) and (11),

$$\det C(1:k, 1:k) = \left(\prod_{1 \leqslant r < s \leqslant k} (x_s - x_r)(y_r - y_s) \right) \cdot \left(\prod_{r=k+1}^{n} \prod_{s=1}^{k} (x_s - y_r) \right) \times \prod_{s=1}^{k} g_s^{-1}.$$

Therefore $x_k \neq y_j$. Analogously, after simplifying,

$$\begin{aligned} C_{ij}^{(k-1)} &= \frac{\det C([1:k-1, i], [1:k-1, j])}{\det C(1:k-1, 1:k-1)} \\ &= \lambda_j^{-1/2} \cdot \prod_{r=k, r \neq j}^{n} \frac{x_i - y_r}{y_j - y_r} \cdot \prod_{r=1}^{k-1} \frac{x_i - x_r}{y_j - x_r}. \end{aligned} \tag{12}$$

Therefore $C_{ij}^{(k-1)} \neq 0$ implies $x_i \neq y_k$. Additionally, (12) immediately yields (10). □

[4] We adopt MATLAB [12] notation for submatrices.

Algorithm 2.1 (*High accuracy GECP on the scaled Cauchy matrix C*). Let C be defined as in (6). The following algorithm performs GECP on C to high relative accuracy.

```
Form the matrix C as defined in (6)
for k = 1 : n − 1
    Find the largest absolute entry in C(k : n, k : n)
    Swap rows and columns of C, and entries of x and y,
            so C_kk is largest
    if C_kk = 0, quit
    for i = k + 1 : n and j = k + 1 : n
        if C_ij = 0 then
            C_ij = −(C_ik C_kj)/C_kk
        else
            C_ij = C_ij ((x_i − x_k)(y_j − y_k))/((x_i − y_k)(y_j − x_k))
D = diag(C)
L = I + tril(C) · D^{−1}
U = I + D^{−1} · triu(C)
```

Next, we present our main algorithm.

Algorithm 2.2 (*High accuracy SVD of $V_P(x)$*). The following algorithm computes the SVD of $V_P(x)$, where $P = \{P_i\}_{i=0}^n$ is a set of orthonormal polynomials, deg $P_i = i$, and x_i are complex nodes.

1. Compute the roots $y_1, y_2, \ldots, y_n$ of P_n;
2. Form the matrices Q, C as defined in (5), (6), respectively;
3. Compute $C = LDU$ using Algorithm 2.1;
4. Use Algorithm 3.1 from [4] to compute the SVD of $V_P(x) = L \cdot D \cdot Y$, where $Y = UQ$.

Theorem 2.1. *Algorithm 2.2 computes the SVD of $V_P(x)$ to high relative accuracy, provided all λ_i are computed to high relative accuracy, $Q = \Lambda^{1/2} \cdot V_P(y)$ is close to orthogonal, and is formed with small norm error. The cost is* $\mathrm{O}(n^3)$.

Proof. The expressions (6), (9), and (10) involve only multiplications, divisions, and subtractions of initial data, thus each entry of L, D, and U is computed to high relative accuracy. The matrices L and U are well conditioned in practice. Since Q is close to orthogonal and is computed with small norm error, the matrix $Y = UQ$ is also well-conditioned and computed with small norm error. The decomposition $V_P(x) = L \cdot D \cdot Y$ is therefore a rank-revealing decomposition [4]. Thus, the results of [4] guarantee that the SVD of $V_P(x)$ is computed to high relative accuracy. □

3. Non-orthonormal polynomials

The SVD of V_P can still be computed to high relative accuracy in many cases when the set of polynomials P is not orthonormal or not even orthogonal. Recalling (4), $V_P(x) = E \cdot V_P(y)$. We can compute the SVD of $V_P(x)$ to high relative accuracy as long as there exist a set of points y such that $V_P(y) = D' \cdot M$, where D' is diagonal and M is well-conditioned. The matrix ED' (just like C in (6)) is a scaled Cauchy matrix, and we can compute its LDU decomposition $ED' = LDU$ analogously to Algorithm 2.1. Therefore $V_P(x) = L \cdot D \cdot (U \cdot M)$ is an RRD of $V_P(x)$, and we can again invoke Algorithm 3.1 from [4] to compute the SVD of $V_P(x)$ to high relative accuracy. Of course, D' must be computable with small relative error componentwise and M must be computable with small norm error.

For orthonormal polynomials (e.g., Chebyshev of the second kind or Laguerre), y can be chosen so M is orthogonal. What if the polynomials P are orthogonal, but not orthonormal? That is

$$V_P(y) = \Lambda^{-1/2} \cdot Q \cdot F^{1/2},$$

where Q is orthogonal, and $F = \mathrm{diag}(c_0, c_1, \ldots, c_{n-1})$ (c_i are defined as in (1)). The matrix $Q \cdot F^{1/2}$ has singular values $c_i^{1/2}$. Therefore if the c_i are of not widely varying magnitude (e.g., in the case of the Chebyshev polynomials of the first kind or the Legendre polynomials), then $Q \cdot F^{1/2}$ is well conditioned and we can still compute the SVD of $V_P(x)$ to high relative accuracy in $O(n^3)$ time.

In fact, we can tolerate a much wilder behavior of the c_i's and still compute the SVD of $V_P(x)$ in polynomial time by using extended precision arithmetic. In the case of Hermite polynomials, for example, the matrix $B = Q \cdot F^{1/2}$ would seem very ill-conditioned: $\kappa(B) = \sqrt{c_{n-1}/c_0} = \sqrt{2^{n-1}(n-1)!}$ (see Table 1). From a complexity point of view, however, our algorithm will still complete in polynomial time. We will simply run our algorithm in extended precision carrying a little more than $\log_2(\kappa(B)) \approx (n - 1 + (n-1)\log_2((n-1)/e))/2 = O(n \log_2 n)$ digits. In general, addition in $k = O(n \log_2 n)$ digit arithmetic costs $O(k)$ and multiplication costs $O(k^2)$ if a straightforward implementation is used [13], or $O(k \log k \log\log k)$, if [14] is used. Either way, the total cost of computing the SVD is still polynomial and does not exceed $O(n^5 \log_2^2 n)$. This analysis, however, goes beyond the scope of this paper and we will not pursue it further.

The situation when P are not orthogonal must be considered on a case-by-case basis. A perfect example in this regard is the ordinary Vandermonde matrix $V(x) = [x_i^{j-1}]_{i,j=1}^n$. For the roots of unity $y_j = \mathrm{e}^{\frac{2\pi\sqrt{-1}(j-1)}{n}}$, $j = 1, 2, \ldots, n$, we have

$$V(y) = \left[\mathrm{e}^{\frac{2(i-1)(j-1)\pi\sqrt{-1}}{n}}\right]_{i,j=1}^n$$

is the *unitary* matrix of the *discrete Fourier transform* (DFT). Thus $V(x) = C \cdot Q$ as in (8), where $Q \equiv V(y)$ and $\Lambda = I$. Now we can proceed as in Section 2. See also [3].

4. Computing y, Λ, and $V_P(y)$

In this section we assume that P_i are orthonormal polynomials and consider the problem of an accurate computation of the roots $y_1, y_2, \ldots, y_n$ of P_n, the Christoffel numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$, and the orthogonal matrix $Q = \Lambda^{1/2} \cdot V_P(y)$. We argue that this computation should not be considered a part of our algorithm because it need only be done once and for all for each class of orthonormal polynomials.

Computing y and Λ is equivalent to computing the Gauss–Christoffel quadrature formulas—a problem that has been studied in detail in [7,10]. This problem may be very ill-conditioned if all that is known about the orthonormal polynomials is the respective weights $w(x)$. If the coefficients of the three-term recurrence (2) are available, then the problem becomes a lot easier [9,10]; the nodes y may be computed as eigenvalues of a tridiagonal matrix [7]. The Christoffel numbers are then the squares of the leading components of the corresponding eigenvectors. For Chebyshev, Legendre, Hermite, and Laguerre orthonormal polynomials this problem is very well conditioned and allows for accurate computation of y and Λ using existing methods.

When $b_j = 0$ (e.g., for Chebyshev, Legendre, and Hermite polynomials), the tridiagonal eigenproblem reduces to the bidiagonal SVD problem—a problem solved by Demmel and Kahan to high relative accuracy regardless of condition numbers [5].

For the Chebyshev polynomials of the first kind

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1, 1]$$

and the Chebyshev polynomials of the second kind

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sin(\arccos x)}, \quad x \in [-1, 1],$$

we have exact formulas for y and $V_P(y)$. The roots of $T_n(x)$ and $U_n(x)$ are $y_i = \cos\frac{(2i-1)\pi}{2n}$ and $y_i = \cos\frac{i\pi}{n+1}$, respectively. The matrices

$$V_T(y) = \left[\cos\frac{2(i-1)(2j-1)\pi}{n}\right]_{i,j=1}^{n} \quad \text{and} \quad V_U(y) = \left[\frac{\sin\frac{(i-1)j\pi}{n+1}}{\sin\frac{j\pi}{n+1}}\right]_{i,j=1}^{n}$$

are computable to high relative accuracy componentwise. The Chebyshev polynomials of the first kind are not orthonormal, but $\kappa(V_T(y)) = \kappa(V_U(y)) = \sqrt{2}$, which means that we can still compute the SVD of any $V_T(x)$ and $V_U(x)$ to high relative accuracy. If we have the orthonormal Chebyshev polynomial bases T' and U' instead of T and U, then we can obtain the Christoffel numbers by scaling the rows of $V_{T'}(y)$ and $V_{U'}(y)$ and thus obtaining orthogonal matrices.

5. Perturbation theory

We have presented an algorithm that computes an accurate SVD, independent of how sensitive the output is to small changes in the input. It is hard to complain about this accuracy, but it is still of interest to know how sensitive the output is to changes in the input. This is because the computation is justified in many applications only when the sensitivity is low enough, since the input is rarely exactly known.

In this section we show that the sensitivity of the singular values of $V_p(x)$ depends on the *relative* separations $|x_i - x_j|/(|x_i| + |x_j|)$ between the $\{x_i\}$, not the *absolute* separations $|x_i - x_j|$.

The smallest relative gap

$$\text{rel\_gap}_x \equiv \min_{i \neq j} \frac{|x_i - x_j|}{|x_i| + |x_j|}$$

between any pair of x_i's is lower bound on the sensitivity of *any* polynomial Vandermonde SVD (or inversion or determinant or linear equation solving) problem, not just for orthogonal polynomials, because a relative change of $\eta = \text{rel\_gap}_x$ in some x_i can make it equal to some other x_j, making the matrix singular and the smallest singular value zero, an $\epsilon = 100\%$ relative change. Another way to look at this is that the rel_gap_x is the (relative) distance from $V_P(x)$ to the nearest singular matrix, or ill-posed problem [2], since the matrix is singular if and only if two x_i are equal. Then the condition number is the reciprocal of this distance.

More specifically, in Theorem 5.1 we prove this claim under the additional assumption that the relative gap between any x_i and y_j (defined as rel_gap_{xy} below) is not too small. We conjecture that this technical assumption is unnecessary, a conjecture which is supported by numerical experiments in Section 6. We also prove the claim without any assumptions for ordinary Vandermonde matrices in Theorem 5.2. Interestingly, it is easier to compute all the singular values of an ordinary Vandermonde matrix to high relative accuracy than it is to compute all the entries of its inverse (it is in fact impossible in our model of arithmetic because the inverse can contain expressions of the form $x_i + x_j + x_k$ [6]).

Theorem 5.1. *Let P be a basis of orthonormal polynomials, let $y_1, y_2, \ldots, y_n$ be the roots of $P_n(x)$, and let $x_1, x_2, \ldots, x_n, x'_1, x'_2, \ldots, x'_n$ be such that*

$$\begin{aligned} \text{rel\_gap}_x &\equiv \min_{i \neq j} \frac{|x_i - x_j|}{|x_i| + |x_j|} \geqslant \eta \\ \text{rel\_gap}_{xy} &\equiv \min_{i,j} \frac{|x_i - y_j|}{|x_i|} \geqslant \zeta \\ \max_i \frac{|x_i - x'_i|}{|x_i|} &\equiv \theta \end{aligned} \tag{13}$$

where $\theta \ll \eta$, $\zeta \leqslant 1$. Let $V_P(x) = W \cdot \Sigma \cdot Z^{\mathrm{T}}$ and $V_P(x') = W' \cdot \Sigma' \cdot (Z')^{\mathrm{T}}$ be the SVDs of $V_P(x)$ and $V_P(x')$, respectively.

If $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ *and* $\Sigma' = \mathrm{diag}(\sigma_1', \sigma_2', \ldots, \sigma_n')$, *then the following bound is valid*

$$|\sigma_i - \sigma_i'| \leqslant \mathrm{O}(\delta)\sigma_i,$$

where $\delta = \frac{n\theta(1/\zeta+1/\eta)}{1-n\theta(1/\zeta+1/\eta)}$, *i.e., small relative changes in the* x_i*'s only cause small relative changes in the singular values for as long as the relative gaps between the* x_i*'s, and the relative gaps between the* x_i*'s and the* y_j*'s are large enough.*

Proof. We use our algorithm to prove the perturbation theory. Let C and C' be the Cauchy matrices (6) corresponding to $V_P(x)$ and $V_P(x')$, respectively, let and $C = LDU$ and $C' = L'D'U'$ be their LDU decompositions. The matrices C and $V_P(x)$ have the same singular values and so do the matrices C' and $V_P(x')$.

We now prove that the entries of L', D' and U' are small relative perturbations of the entries of L, D and U, respectively. The inequalities (13) imply

$$x_i' - x_j' = (1 + \delta_{ij})(x_i - x_j), \quad y_i - x_j' = (1 + \delta_{ij}')(y_i - x_j),$$

where $|\delta_{ij}| \leqslant \theta/\eta$ and $|\delta_{ij}'| \leqslant \theta/\zeta$. Now from (12) we obtain

$$u_{ij}' = (C')_{ij}^{(i)} = u_{ij} \prod_{r=1}^{i} \frac{1 + \delta_{ir}}{1 + \delta_{jr}'} \prod_{r=i+1}^{n} (1 + \delta_{ir}') = u_{ij}(1 + \bar{\delta}),$$

where $|\bar{\delta}| \leqslant \frac{\theta(i/\eta+n/\zeta)}{1-\theta(i/\eta+n/\zeta)} \leqslant \frac{n\theta(1/\zeta+1/\eta)}{1-n\theta(1/\zeta+1/\eta)} = \delta$. By analogy we also obtain

$$|l_{ij} - l_{ij}'| \leqslant \delta|l_{ij}|, \quad |d_{ii} - d_{ii}'| \leqslant \delta|d_{ii}| \quad \text{and} \quad |u_{ij} - u_{ij}'| \leqslant \delta|u_{ij}|.$$

Since the matrices L and U are assumed to be well conditioned in practice, we can use Theorem 2.1 from [4] to conclude that

$$|\sigma_i - \sigma_i'| \leqslant \mathrm{O}(\delta)\sigma_i.$$

The modest constant hidden in the above big-O notation can also be found in [4, Theorem 2.1]. □

We also conjecture that the above theorem is still valid independent of the relative gaps between the x_i's and the y_j's and successfully test this conjecture in the numerical experiments below.

Theorem 5.2. *Let* $V(x)$ *and* $V(x')$ *be ordinary Vandermonde matrices* (*i.e.*, $P_i(x) = x^i$). *Define* $\mathrm{rel\_gap}_x$, θ, σ_i, *and* σ_i' *as in Theorem* 5.1. *Then if* $\theta \ll \mathrm{rel\_gap}_x \leqslant \frac{\pi}{2n^2}$, *we have*

$$|\sigma_i - \sigma_i'| \leqslant \mathrm{O}(n\theta/\mathrm{rel\_gap}_x)\sigma_i.$$

Proof. The proof is very similar to that of Theorem 5.1. Let $y_j \equiv \mathrm{e}^{2\sqrt{-1}\pi(j-1)/n}$. Note that if we multiply each x_i (and x_i') by ω, where $|\omega| = 1$, then $\mathrm{rel\_gap}_x$, θ, σ_i, and σ_i' do not change, but rel_gap_{xy} may. In particular, we can choose ω to make rel_gap_{xy} at least about π/n^2, a worst case that occurs when the x_i are evenly spaced on the unit circle with angular separation $2\pi/n^2$. The rest of the proof follows as before. □

6. Numerical experiments

We ran extensive numerical experiments to verify the correctness of our SVD algorithm and we present here one such experiment. We start with a 20×20 Chebyshev–Vandermonde matrix A (with the orthonormalized Chebyshev polynomials of the first kind) and uniformly distributed random nodes in $[0, 0.2]$—within the interval of orthogonality. The resulting matrix has singular values, computed using our algorithm as implemented in MATLAB [12], ranging over 35 orders of magnitude. We also computed the singular values using 60-digit arithmetic in Mathematica and got the same result to 14 digits. For comparison, the singular values computed by the traditional SVD algorithm have relative error exceeding one when they are less than $\sigma_1\epsilon$, as expected. We present the results of this experiment in Table 2 and Fig. 1.

We also tested the predictions of Theorem 5.1 for the sensitivity of the SVD with respect to small perturbations of the initial data. We ran two tests.

In the first test we took our test data below, for which the minimum relative gap between the x_i's (rel_gap_x) is 8.05×10^{-5} and the minimum relative gap between the x_i's and the y_j's (rel_gap_{xy}) is 1.20×10^{-2}. We introduced random perturbation in the 10th digit of each of the x_i's, which resulted in a 5th digit perturbation in the relative gaps between the x_i's. We then computed the SVD of the thus perturbed matrices using Algorithm 2.2 and in Mathematica [16] using 60-digit arithmetic. The singular values computed using each of these methods agreed with the singular values of the unperturbed matrix to 5 digits, as predicted by Theorem 5.1.

In our second test we tested the sensitivity of the algorithm to relative changes in the x_i's when the relative gaps between the x_i's and the y_j's are small. We replaced the odd nodes $x_1, x_3, \ldots, x_{19}$ with $y_1, y_3, \ldots, y_{19}$, ensuring that the relative gaps between the x_i and the y_j are zero. The minimum relative gap among the x_i's is 3.61×10^{-3}. As in our previous test we introduced a random perturbation in the 10th digit in each x_i and computed the SVD of the Chebyshev–Vandermonde matrix corresponding to the perturbed nodes. The singular values of the perturbed matrix agreed with the singular values of the unperturbed matrix to 7 digits, as expected, i.e., the prediction of Theorem 5.1 holds whether ζ is small or large.

Both numerical tests confirm our conjecture that the SVD is only sensitive to changes in the relative gaps (η in Theorem 5.1) between the x_i's, but is not sensitive to changes in the relative gaps (ζ in Theorem 5.1) between the x_i's and the y_j's.

Table 2
The nodes and singular values of a 20 × 20 Chebyshev–Vandermonde matrix

Nodes $x_i \in [0, 0.2]$	Mathematica	Algorithm 2.2	Traditional
1.754545429110508e−1	3.9431633617865161e+00	3.9431633617865183e+00	3.9431633617865129e+00
1.785123568004681e−1	1.8637975498148207e+00	1.8637975498148214e+00	1.8637975498148200e+00
1.822530760791615e−1	5.8378797599607701e−01	5.8378797599607724e−01	5.8378797599607712e−01
1.337265877473653e−1	8.7323445025621033e−02	8.7323445025621074e−02	8.7323445025621046e−02
1.318492786266806e−1	8.4416423668173899e−03	8.4416423668173951e−03	8.4416423668173465e−03
1.399501459680857e−2	4.9924169675294070e−04	4.9924169675294081e−04	4.9924169675289918e−04
1.192756955101435e−1	2.1594947468609303e−05	2.1594947468609320e−05	2.1594947468623757e−05
1.466911661412304e−1	7.2784564714692648e−07	7.2784564714692700e−07	7.2784564716638695e−07
1.037535735742760e−1	2.7518442142941135e−08	2.7518442142941122e−08	2.7518442191715476e−08
1.791600477478306e−1	5.4735925469678211e−10	5.4735925469678263e−10	5.4735930523441333e−10
1.791744650424257e−1	1.6277393913826226e−11	1.6277393913826245e−11	1.6277437245500341e−11
9.045924651005480e−2	2.8485946493449978e−13	2.8485946493449983e−13	2.8487032864800328e−13
7.941574738520146e−2	2.4158345438031927e−15	2.4158345438031939e−15	2.4187910790488627e−15
1.200488382611424e−1	1.8338131347839962e−17	1.8338131347839956e−17	6.2473401943443176e−16
1.932439387160720e−1	1.8567109268982685e−19	1.8567109268982687e−19	8.1944522168311346e−17
1.736232440468190e−1	6.8715776044237070e−22	6.8715776044237004e−22	6.1039995944973018e−17
1.468702150155248e−1	7.1569407338259544e−24	7.1569407338259603e−24	3.0843251283083951e−17
7.419133543499865e−2	5.4140304287851849e−27	5.4140304287851849e−27	2.3979357147069272e−17
1.669780081142836e−1	8.9977317668911131e−29	8.9977317668911198e−29	2.0047699810203855e−17
1.020762301650144e−1	4.2539407269598119e−35	4.2539407269598130e−35	7.3091886573513421e−18

The digits that disagree with the results from Mathematica are underlined.

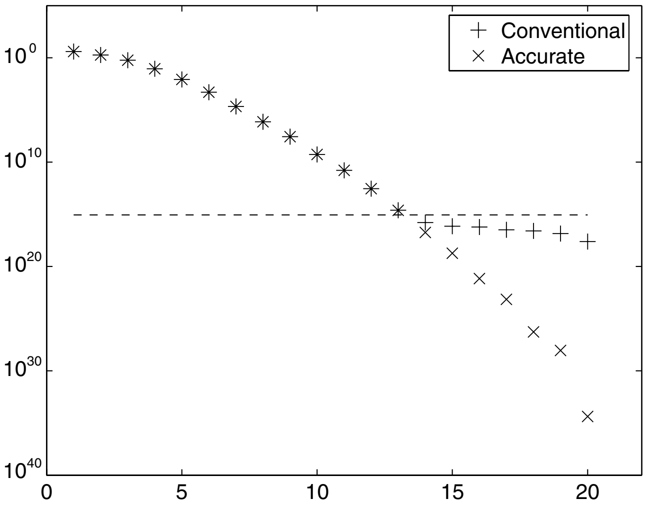

Fig. 1. Singular values of a 20 × 20 Chebyshev–Vandermonde matrix. Data below the dashed line may be inaccurate for the conventional algorithm.

Acknowledgments

The authors would like to thank Gene Golub, Alberto Grunbaum, and Ming Gu for several useful discussions in the process of preparation of this work as well as the two anonymous referees for making useful remarks and shortening the proof of Lemma 2.1.

References

[1] M. Abramowitz, I. Stegun (Eds.), Handbook of Mathematical Functions, NBS, Appl. Math. Ser., vol. 55, US Government Printing Office, Washington, DC, 1965.

[2] J. Demmel, On condition numbers and the distance to the nearest ill-posed problem, Numer. Math. 51 (3) (1987) 251–289.

[3] J. Demmel, Accurate singular value decompositions of structured matrices, SIAM J. Matrix Anal. Appl. 21 (2) (1999) 562–580.

[4] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, Z. Drmač, Computing the singular value decomposition with high relative accuracy, Linear Algebra Appl. 299 (1–3) (1999) 21–80.

[5] J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, SIAM J. Sci. Statist. Comput. 11 (5) (1990) 873–912.

[6] J. Demmel, P. Koev, Necessary and sufficient conditions for accurate and efficient rational function evaluation and factorizations of rational matrices, Structured Matrices in Mathematics, Computer Science, and Engineering, II (Boulder, CO, 1999), Contemp. Math., vol. 281, Amer. Math. Soc., Providence, RI, 2001, pp. 117–143.

[7] W. Gautschi, Construction of Gauss–Christoffel quadrature formulas, Math. Comp. 22 (1968) 251–270.

[8] W. Gautschi, The condition of Vandermonde-like matrices involving orthogonal polynomials, Linear Algebra Appl. 52/53 (1983) 293–300.
[9] W. Gautschi, Is the recurrence relation for orthogonal polynomials always stable? BIT 33 (1993) 277–284.
[10] W. Gautschi, Orthogonal polynomials: applications and computation, Acta Numerica (1996) 45–119.
[11] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM, Philadelphia, 2002.
[12] The MathWorks, Inc., Natick, MA. MATLAB Reference Guide, 1992.
[13] D. Priest, Algorithms for arbitrary precision floating point arithmetic, in: P. Kornerup, D. Matula (Eds.), Proceedings of the 10th Symposium on Computer Arithmetic, IEEE Computer Society, Grenoble, France, 1991, pp. 132–145.
[14] A. Schönhage, V. Strassen, Schnelle Multiplikation grosser Zahlen, Computing (Arch. Elektron. Rechnen) 7 (1971) 281–292.
[15] G. Szegö, Orthogonal Polynomials, American Mathematical Society, 1967.
[16] S. Wolfram, Mathematica: A System for Doing Mathematics by Computer, Addison-Wesley, Reading, MA, 1988.