MATHEMATICS OF COMPUTATION
VOLUME 57, NUMBER 196
OCTOBER 1991, PAGES 703–721

CHEBYSHEV-VANDERMONDE SYSTEMS

LOTHAR REICHEL AND GERHARD OPFER

Dedicated to the memory of Lothar Collatz

ABSTRACT. A Chebyshev-Vandermonde matrix

$$V = [p_j(z_k)]_{j,k=0}^{n} \in \mathbb{C}^{(n+1)\times(n+1)}$$

is obtained by replacing the monomial entries of a Vandermonde matrix by Chebyshev polynomials p_j for an ellipse. The ellipse is also allowed to be a disk or an interval. We present a progressive scheme for allocating distinct nodes z_k on the boundary of the ellipse such that the Chebyshev-Vandermonde matrices obtained are reasonably well-conditioned. Fast progressive algorithms for the solution of the Chebyshev-Vandermonde systems are described. These algorithms are closely related to methods recently presented by Higham. We show that the node allocation is such that the solution computed by the progressive algorithms is fairly insensitive to perturbations in the right-hand side vector. Computed examples illustrate the numerical behavior of the schemes. Our analysis can also be used to bound the condition number of the polynomial interpolation operator defined by Newton's interpolation formula. This extends earlier results of Fischer and the first author.

1. INTRODUCTION

Let E_ρ, for some $\rho \in [0, 1]$, be the closed ellipse with boundary curve

$$\partial E_\rho := \{e^{it} + \rho e^{-it} : 0 \le t \le 2\pi\}, \qquad i := \sqrt{-1}. \tag{1.1}$$

Define the polynomials in $z = w + \rho w^{-1}$,

$$\begin{cases} p_0(z) := 1, \\ p_j(z) := w^j + (\rho/w)^j, \qquad j = 1, 2, 3, \ldots. \end{cases} \tag{1.2}$$

It can easily be shown (see, e.g., Smirnov and Lebedev [17, Chapter 5]) that the p_j are *Chebyshev polynomials* for E_ρ with leading coefficient one, i.e., among all monic polynomials of degree j, p_j is the unique polynomial of minimum uniform norm on E_ρ.

Received September 26, 1988; revised May 16, 1989, November 6, 1989, April 18, 1990.

1980 *Mathematics Subject Classification* (1985 *Revision*). Primary 65F05, 65F35, 65D05, 65D32.

Key words and phrases. Vandermonde-like matrix, ordering of nodes, numerical conditioning, progressive algorithm, Newton interpolation formula.

The first author's research was supported in part by the IBM Bergen Scientific Centre and by NSF under Grant DMS-8704196.

Example 1.1. Let $\rho := 0$. Then E_0 is the closed unit disk, and the p_j are the monomials $p_j(z) = z^j$, $j \geq 0$. □

Example 1.2. Let $\rho := 1$. Then $\partial E_1 = E_1$ is the interval $[-2, 2]$, and the p_j are the 'ordinary' Chebyshev polynomials on $[-2, 2]$, i.e., $p_j(z) = 2\cos(j \arccos(z/2))$, $j \geq 1$. □

Chebyshev-Vandermonde matrices (henceforth abbreviated *CV matrices*) $V = V_\rho = [v_{jk}]_{j,k=0}^n$, $v_{jk} := p_j(z_k)$, arise naturally in polynomial interpolation problems when the basis (1.2) is used for Π_n, the set of polynomials of degree at most n. Let $(z_k, f_k) \in \mathbb{C}^2$, $0 \leq k \leq n$, be the given data, where the nodes z_k are assumed to be pairwise distinct. The computation of an interpolating polynomial $q_n \in \Pi_n$ such that $q_n(z_k) = f_k$, $0 \leq k \leq n$, in the form

$$q_n(z) = \sum_{j=0}^{n} a_j p_j(z) \tag{1.3}$$

can be accomplished by solving a *dual CV system*, namely

$$V_\rho^{\mathrm{T}} \mathbf{a} = \mathbf{f}, \tag{1.4}$$

where $\mathbf{a} := (a_0, a_1, \ldots, a_n)^{\mathrm{T}}$ and $\mathbf{f} := (f_0, f_1, \ldots, f_n)^{\mathrm{T}}$. *Primal CV systems*

$$V_\rho \mathbf{a} = \mathbf{g}, \qquad \mathbf{g} := (g_0, g_1, \ldots, g_n)^{\mathrm{T}}, \tag{1.5}$$

arise in the computation of weights of interpolatory quadrature rules with nodes z_k when the polynomial basis (1.2) is used. We note that the CV matrix V_ρ simplifies to an 'ordinary' Vandermonde matrix when $\rho = 0$ in (1.2) (cf. Example 1.1).

Our interest in the basis (1.2) and in fast solution methods for the linear systems of equations (1.4) and (1.5) stems from our ability to bound the growth with n of the condition numbers of the CV matrices V_ρ for certain progressively allocated nodes on ∂E_ρ. Introduce the condition number

$$\kappa_p(V_\rho) := \|V_\rho\|_p \|V_\rho^{-1}\|_p, \tag{1.6}$$

where $\| \ \|_p$ denotes the usual matrix p-norm on $\mathbb{C}^{(n+1)\times(n+1)}$ [12, p. 56]. We show in §3 that for our progressively determined nodes the condition number $\kappa_\infty(V_\rho)$ grows at most polynomially[1] with n for any (fixed) $\rho \in [0, 1)$. For $\rho = 1$, the condition number $\kappa_\infty(V_1)$ grows at most like $n^{O(\log n)}$. The latter

[1]This bound has for $\rho = 0$ recently been improved by A. Córdova, W. Gautschi, and S. Ruscheweyh (see Addendum at the end of this paper).

bound should be compared with recent results by Gautschi and Inglese [8], who show that for real nodes z_k the condition number $\kappa_\infty(V_0)$ generally grows at least like $O(2^{n/2})$ with n. Related results and examples can also be found in [9, 10, 11]. These results indicate that unless special care is taken when allocating the nodes z_k, the condition numbers of Vandermonde and CV matrices in general grow *exponentially* with n.

The numerical solution of the linear systems of equations (1.4) and (1.5) has received considerable attention when V_ρ is an 'ordinary' Vandermonde matrix, i.e., when $\rho = 0$. Then the systems (1.4) and (1.5) can be solved in $O(n^2)$ arithmetic operations by methods of Björck and Pereyra [1] and Tang and Golub [18]. This operation count compares favorably with the $O(n^3)$ arithmetic operations required for the solution of (1.4) or (1.5) by Gaussian elimination. Recently Higham [13, 14] presented (nonprogressive) algorithms for the solution of Vandermonde-like linear systems of equations involving polynomials that satisfy a three-term recurrence relation. These algorithms are obtained by modifying the nonprogressive algorithms for 'ordinary' Vandermonde systems in [1].

Our scheme for progressively determining nodes z_k makes it attractive to use progressive algorithms for the solution of the CV systems (1.4) and (1.5); i.e., the solution of (1.4) and (1.5) for $n = m+1$ is computed by modifying the solution obtained for $n = m$. Progressive algorithms allow us to conveniently solve (1.4) and (1.5) for increasing values of n until the computed interpolation polynomial q_n approximates a given function sufficiently accurately, or until the determined quadrature rule yields a small enough integration error. In §2 we modify progressive algorithms of Björck and Pereyra [1] in order to obtain progressive CV solvers that require $O(n^2)$ arithmetic operations and $O(n)$ storage locations for the solution of (1.4) and (1.5) for any $\rho \in [0, 1]$ and $V_\rho \in \mathbb{C}^{(n+1)\times(n+1)}$. If $\rho = 0$, then our progressive CV solvers simplify to the progressive Vandermonde solvers in [1].

The error propagation of CV solvers does not only depend on the condition number $\kappa_p(V_\rho)$, but also on the ordering of the nodes z_k. For instance, let $\rho = 0$ (unit disk case, cf. Example 1.1) and let the nodes z_k, $0 \le k \le n$, be some enumeration of the $n+1$ roots of unity $\{\exp(2\pi ik/(n+1))\}_{k=0}^{n}$. Then the Vandermonde matrix V_0 is a scalar multiple of an orthogonal matrix, and therefore $\kappa_2(V_0) = 1$. However, if $z_k = \exp(2\pi ik/(n+1))$, the error in the solution due to propagated roundoff errors grows rapidly with n (see Figures 4.2.2 and 4.2.4 of §4). On the other hand, the CV solvers yield a fairly small amplification of roundoff errors if the z_k are ordered in such a way that the nodes in each subset $\{z_k\}_{k=0}^{l}$, $0 \le l \le n$, are 'approximately uniformly distributed' on the unit circle (see Examples 4.2–4.3 of §4). Such an ordering is given by

$$z_k := \exp(2\pi i c_k), \qquad 0 \le k \le n, \tag{1.7}$$

where $\{c_k\}_{k=0}^{\infty}$ is the *van der Corput sequence* defined as follows. Let the nonnegative integer k have the binary representation

$$k = \sum_{j=0}^{\infty} k_j 2^j, \qquad k_j \in \{0, 1\}. \tag{1.8a}$$

Then c_k is given by

$$c_k := \sum_{j=0}^{\infty} k_j 2^{-j-1}. \tag{1.8b}$$

Table 1.1 shows some values of c_k and $\arg(z_k)$ defined by (1.7) and (1.8).

TABLE 1.1
The van der Corput sequence

k	0	1	2	3	4	5	6	7
c_k	0	$\frac{4}{8}$	$\frac{2}{8}$	$\frac{6}{8}$	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{3}{8}$	$\frac{7}{8}$
$2\pi c_k$	0	π	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	$\frac{\pi}{4}$	$\frac{5\pi}{4}$	$\frac{3\pi}{4}$	$\frac{7\pi}{4}$

Properties of the van der Corput sequence are discussed by, e.g., Hlawka [15, p. 93], and properties of the nodes (1.7) are considered in §3, as well as in [6, 7]. In §3 we use the van der Corput sequence to allocate nodes z_k on ∂E_ρ as follows. If $0 \le \rho < 1$, then we let $\alpha \in \mathbb{R}$ be an arbitrary but fixed constant and define

$$z_k := \exp(2\pi i(c_k + \alpha)) + \rho \exp(-2\pi i(c_k + \alpha)) \in \partial E_\rho, \qquad k = 0, 1, 2, \ldots. \tag{1.9}$$

If instead $\rho = 1$, then we define nodes on $[-2, 2]$ by

$$\begin{cases} z_0 := -2, \\ z_k := 2\cos(\pi c_{k-1}), \qquad k = 1, 2, 3, \ldots. \end{cases} \tag{1.10}$$

Example 1.3. Let $l \ge 0$ be an arbitrary integer. Assume first that $\rho = 0$. Then the set of nodes $\{z_k\}_{k=0}^{2^l-1}$ defined by (1.9) is a set of equidistant points on the unit circle. More generally, for any fixed $\rho \in [0, 1)$, the set of nodes $\{z_k\}_{k=0}^{2^l-1}$ is a set of so-called Fejér points on ∂E_ρ. Fejér points are defined in, e.g., [17, Chapter 1] and [6, 7]. □

Example 1.4. Let $l \ge 0$ be an arbitrary integer. Then the set of nodes $\{z_k\}_{k=0}^{2^l}$ defined by (1.10) is the set of extreme points of the Chebyshev polynomial $p_{2^l}(x) := 2\cos(2^l \arccos(x/2))$ for the interval $[-2, 2]$. □

In §3 we present bounds for the propagated error due to errors in the right-hand side for the CV solvers when the matrices V_ρ are defined by the nodes

(1.9) or (1.10). These bounds grow slower than exponentially with n. If we would use $c_k := k/n$ in (1.9) and (1.10), then the error in the computed solution would grow exponentially with n. This is illustrated by computed examples in §4.

We finally remark that for a *fixed* value of n, and the nodes (1.9) or (1.10) the solution of (1.4) can be computed by the fast Fourier transform method in $O(n \log n)$ arithmetic operations (see, e.g., Ellacott [4] for a discussion on the use of nodes (1.9)). However, it is difficult to make this approach efficient in a progressive algorithm.

2. Progressive algorithms for CV systems

This section describes progressive algorithms for the solution of linear systems (1.4) and (1.5), and introduces notation to be used in the analysis of §3. Our derivation of the algorithms follows closely the derivation by Björck and Pereyra [1] of progressive algorithms for the solution of 'ordinary' Vandermonde systems.

Let $(z_k, f_k) \in \mathbb{C}^2$, $0 \le k \le n$, be given data with pairwise distinct nodes z_k. We wish to compute the coefficients $a_j^{(n)}$ of the polynomial

$$q_n(z) = \sum_{j=0}^{n} a_j^{(n)} p_j(z), \tag{2.1}$$

which is uniquely determined by $q_n(z_k) = f_k$, $0 \le k \le n$. Following [1], we first express q_n in Newton form

$$q_n(z) =: \sum_{j=0}^{n} c_{n-j}^{(n)} \prod_{k=0}^{j-1} (z - z_k) = q_{n-1}(z) + c_0^{(n)} \prod_{k=0}^{n-1} (z - z_k), \tag{2.2}$$

where empty products are understood to have value one. Assume that the coefficients $a_j^{(n-1)}$ of q_{n-1} are already known, and write the product on the right in formula (2.2) as a linear combination of the polynomials p_j, i.e.,

$$\sum_{j=0}^{n} b_j^{(n)} p_j(z) := \prod_{k=0}^{n-1} (z - z_k). \tag{2.3}$$

In order to determine the coefficients $b_j^{(n)}$, we write (2.3) in the form

$$\sum_{j=0}^{n} b_j^{(n)} p_j(z) = (z - z_{n-1}) \prod_{j=0}^{n-2} (z - z_j) = (z - z_{n-1}) \sum_{j=0}^{n-1} b_j^{(n-1)} p_j(z), \tag{2.4}$$

and assume that the coefficients $b_j^{(n-1)}$ are already known. The $b_j^{(n)}$ can now be determined by substituting (1.2) and $z = w + \rho w^{-1}$ into (2.4), and comparing coefficients of equal nonnegative powers of w on the left and right. We obtain in this manner, for $n \geq 2$,

$$\begin{cases} b_0^{(n)} = -z_{n-1} b_0^{(n-1)} + 2\rho b_1^{(n-1)}, \\ b_j^{(n)} = b_{j-1}^{(n-1)} - z_{n-1} b_j^{(n-1)} + \rho b_{j+1}^{(n-1)}, & 1 \leq j \leq n-2, \\ b_{n-1}^{(n)} = b_{n-2}^{(n-1)} - z_{n-1} b_{n-1}^{(n-1)}, \\ b_n^{(n)} = 1. \end{cases}$$

Finally, substituting (2.1) and (2.3) into (2.2) and comparing coefficients of the p_j yields expressions for the $a_j^{(n)}$ in terms of the coefficients $a_j^{(n-1)}$:

$$\begin{cases} a_0^{(0)} = f_0, \\ a_j^{(n)} = a_j^{(n-1)} + c_0^{(n)} b_j^{(n)}, & j = 0, 1, \dots, n-1, \\ a_n^{(n)} = c_0^{(n)} b_n^{(n)}. \end{cases}$$

Combining the above formulas gives rise to the following algorithm.

Algorithm 1. Progressive algorithm for the solution of dual CV systems.

Data: ρ; (z_n, f_n), $n = 0, 1, 2, \dots$

$c_0^{(0)} := f_0$; $a_0^{(0)} := f_0$;

for $n := 1, 2, 3, \dots$ **until** no more nodes **do**

: compute divided difference $c_0^{(n)}$:

$c_n^{(n)} := f_n$;

for $j := n-1, n-2, \dots, 0$ **do**

$c_j^{(n)} := (c_{j+1}^{(n)} - c_j^{(n-1)})/(z_n - z_j)$;

: compute coefficients $b_j^{(n)}$:

if $n = 1$ **then**

$b_0^{(1)} := -z_0$; $b_1^{(1)} := 1$;

else

$b_0^{(n)} := 2\rho b_1^{(n-1)} - z_{n-1} b_0^{(n-1)}$;

for $j := 1, 2, \dots, n-2$ **do**

$b_j^{(n)} := b_{j-1}^{(n-1)} - z_{n-1} b_j^{(n-1)} + \rho b_{j+1}^{(n-1)}$;

$b_{n-1}^{(n)} := b_{n-2}^{(n-1)} - z_{n-1} b_{n-1}^{(n-1)}$; $b_n^{(n)} := 1$;

: compute coefficients $a_j^{(n)}$:

for $j := 0, 1, \dots, n-1$ **do**

$a_j^{(n)} := a_j^{(n-1)} + c_0^{(n)} b_j^{(n)}$;

$a_n^{(n)} := c_0^{(n)}$;

We remark that the nodes in Algorithm 1 are arbitrary pairwise distinct nodes. The value of ρ determines the polynomial basis. Two FORTRAN subroutines for Algorithm 1 are listed in [16]: one for complex nodes and $0 \leq \rho < 1$, using complex arithmetic, and one for the important special case of real nodes and $\rho = 1$, using real arithmetic only. The codes are available from the authors. The subroutines require $O(n)$ storage locations in order to compute the coefficients

$\{a_j^{(n)}\}_{j=0}^n$ of (2.1). The operation count for computing these coefficients by the code for real nodes and $\rho = 1$ is $\frac{3}{2}n^2 + O(n)$ multiplications or divisions and $\frac{5}{2}n^2 + O(n)$ additions or subtractions. If we compute the coefficients $\{a_j^{(n)}\}_{j=0}^n$ by the code for complex nodes and $0 \le \rho < 1$, and convert complex arithmetic operations into real ones, then $10n^2 + O(n)$ real multiplications or divisions are required. This operation count is based on the observation that one complex multiplication takes three real multiplications, and one complex division takes six real multiplications or divisions.

We now turn to the derivation of a progressive algorithm for the solution of primal CV systems (1.5). Following the approach used in [1] for the derivation of Vandermonde solvers, we first make a matrix interpretation of Algorithm 1. The matrices introduced will be used in the error analysis of §3.

Let m be an arbitrary integer larger than or equal to n. For $0 \le k < m$, define the matrices

$$L_k := \left(\begin{array}{c|ccccc} I_{k+1} & & & \mathbf{0} & & \\ \hline & 1 & 0 & \dots & & 0 \\ & -1 & 1 & & & \\ \mathbf{0} & \vdots & \ddots & \ddots & & \vdots \\ & & & -1 & 1 & 0 \\ & 0 & & & -1 & 1 \end{array}\right) \in \mathbb{R}^{(m+1)\times(m+1)}$$

and

$$D_k := \operatorname{diag}[1, 1, \dots, 1, (z_{k+1} - z_0)^{-1}, (z_{k+2} - z_1)^{-1}, \dots, (z_m - z_{m-k-1})^{-1}] \in \mathbb{C}^{(m+1)\times(m+1)},$$

where I_j denotes the identity matrix of order j. Introduce the coordinate vectors $\mathbf{e}_j$, i.e., $\mathbf{e}_j$ is the $(j+1)$st column of I_{m+1}. Then the divided difference $c_0^{(n)}$ in formula (2.2) can be written as

$$c_0^{(n)} = \mathbf{e}_n^{\mathrm{T}} D_{n-1} L_{n-1} D_{n-2} L_{n-2} \cdots D_0 L_0 \mathbf{f}, \qquad 1 \le n \le m. \tag{2.5}$$

Introduce the tridiagonal matrices

$$W_j := \begin{pmatrix} -z_j & 2\rho & 0 & \dots & & & 0 \\ 1 & -z_j & \rho & \ddots & & & \vdots \\ 0 & 1 & -z_j & \rho & & & \\ \vdots & & 1 & -z_j & \ddots & & \\ & & & \ddots & \ddots & & 0 \\ & & & & 1 & -z_j & \rho \\ 0 & \dots & & & 0 & 1 & -z_j \end{pmatrix} \in \mathbb{C}^{(m+1)\times(m+1)},$$

$0 \le j \le m$, and let

$$\tilde{\mathbf{b}}^{(n)} := (b_0^{(n)}, b_1^{(n)}, \dots, b_n^{(n)}, 0, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{C}^{m+1}, \qquad 0 \le n \le m,$$

$$\tilde{\mathbf{a}}^{(n)} := (a_0^{(n)}, a_1^{(n)}, \dots, a_n^{(n)}, 0, 0, \dots, 0)^{\mathrm{T}} \in \mathbb{C}^{m+1}, \qquad 0 \le n \le m,$$

where the $b_j^{(n)}$ and $a_j^{(n)}$ are defined by (2.3) and (2.1), respectively. Then

$$\tilde{\mathbf{b}}^{(n)} = W_{n-1}W_{n-2}\cdots W_0\mathbf{e}_0, \qquad 1 \le n \le m, \tag{2.6}$$

$$\tilde{\mathbf{a}}^{(n)} = \tilde{\mathbf{a}}^{(n-1)} + \tilde{\mathbf{b}}^{(n)}c_0^{(n)} = \tilde{\mathbf{a}}^{(n-1)} + S_n\mathbf{f}, \qquad 0 \le n \le m, \tag{2.7}$$

where

$$\begin{cases} S_n := W_{n-1}W_{n-2}\cdots W_0\mathbf{e}_0\mathbf{e}_n^{\mathrm{T}}D_{n-1}L_{n-1}D_{n-2}L_{n-2}\cdots D_0L_0, \\ \qquad\qquad\qquad\qquad 1 \le n \le m, \\ S_0 := \mathbf{e}_0\mathbf{e}_0^{\mathrm{T}}, \end{cases} \tag{2.8}$$

and $\tilde{\mathbf{a}}^{(-1)} := \mathbf{0}$. Hence,

$$\tilde{\mathbf{a}}^{(n)} = \sum_{j=0}^{n} S_j\mathbf{f}, \qquad 1 \le n \le m. \tag{2.9}$$

Let $\widetilde{V}$ denote the CV matrix of order $m+1$ defined by the node set $\{z_j\}_{j=0}^m$ and $\rho \in [0, 1]$. It follows from (1.4) and (2.9) that

$$\widetilde{V}^{-\mathrm{T}} = \sum_{j=0}^{m} S_j, \tag{2.10}$$

and therefore

$$\widetilde{V}^{-1} = \sum_{j=0}^{m} S_j^{\mathrm{T}}. \tag{2.11}$$

From (2.11), and the fact that the W_j commute, we obtain the following algorithm for the solution of (1.5):

Algorithm 2. Progressive algorithm for the solution of primal CV systems.

Data: ρ; (z_n, f_n), $n = 0, 1, 2, \ldots$

$f_0^{(0)} := f_0$; $a_0^{(0)} := f_0$; $u_0^{(0)} := 1$;

for $n := 1, 2, 3, \ldots$ **until** no more nodes **do**

: compute $f_n^{(n)} := \mathbf{e}_0^{\mathrm{T}}W_0^{\mathrm{T}}W_1^{\mathrm{T}}\cdots W_{n-1}^{\mathrm{T}}\mathbf{f}$:

$f_0^{(n)} := f_n$;

for $k := 0, 1, \ldots, n-2$ **do**

$f_{k+1}^{(n)} := f_k^{(n)} - z_k f_k^{(n-1)} + \rho f_k^{(n-2)}$;

if $k = n-2$ **then** $f_{k+1}^{(n)} := f_{k+1}^{(n)} + \rho f_k^{(n-2)}$;

$f_n^{(n)} := f_{n-1}^{(n)} - z_{n-1}f_{n-1}^{(n-1)}$;

: compute $\mathbf{u}^{(n)} := (1/w_0^{(n)}, 1/w_1^{(n)}, \ldots, 1/w_n^{(n)})^{\mathrm{T}}$, where $\mathbf{w}^{(n)} = (w_0^{(n)}, w_1^{(n)}, \ldots, w_n^{(n)})^{\mathrm{T}} := L_0^{\mathrm{T}}D_0L_1^{\mathrm{T}}D_1\cdots L_{n-1}^{\mathrm{T}}D_{n-1}\mathbf{e}_n$:

$u_n^{(n)} := 1$;

for $k := 0, 1, \ldots, n-1$ **do**

$u_k^{(n)} := (z_k - z_n)u_k^{(n-1)}$;

$u_n^{(n)} := (z_n - z_k)u_n^{(n)}$;

: compute $\mathbf{a}^{(n)} := \mathbf{a}^{(n-1)} + \mathbf{w}^{(n)} f_n^{(n)}$:
for $k := 0, 1, \ldots, n-1$ **do**
$\quad a_k^{(n)} := a_k^{(n-1)} + f_n^{(n)} / u_k^{(n)}$;
$a_n^{(n)} := f_n^{(n)} / u_n^{(n)}$;

Two FORTRAN subroutines for Algorithm 2 are listed in [16]: one for complex nodes and $0 \le \rho < 1$, and one for real nodes and $\rho = 1$. The codes are available from the authors. Similarly as for Algorithm 1, the nodes for Algorithm 2 are assumed to be pairwise distinct but otherwise arbitrary, and ρ determines the polynomial basis (1.2).

3. Condition number bounds

In this section we assume that the nodes z_k are given by (1.9) or (1.10). We derive bounds for the rate of growth with n of the condition numbers (1.6) of the CV matrices V_ρ of order $n+1$. Also, we present bounds for propagated errors due to errors in the right-hand side vectors in (1.4) and (1.5). These bounds are derived by bounding the quantities computed by Algorithms 1 and 2; i.e., in order to bound V_ρ^{-1}, we bound the mapping from the right-hand side vector in (1.4) to the divided differences $c_{n-j}^{(n)}$ in (2.2), and the mapping from the divided differences to the solution vector $\mathbf{a}$. Our analysis extends previous results in [7] on bounds for the condition number for the Newton interpolation formula. This application will be discussed in Remark 3.1 below.

Introduce the mappings $M_1 : \mathbb{C}^{n+1} \to \mathbb{C}^{n+1}$ and $M_2 : \mathbb{C}^{n+1} \to \mathbb{C}^{n+1}$ defined by

$$M_1 \mathbf{f} := \mathbf{c} = (c_0^{(n)}, c_1^{(n)}, \ldots, c_n^{(n)})^{\mathrm{T}}, \tag{3.1}$$

$$M_2 \mathbf{c} := \mathbf{a} = (a_0^{(n)}, a_1^{(n)}, \ldots, a_n^{(n)})^{\mathrm{T}}, \tag{3.2}$$

where the $c_j^{(n)}$ are the divided differences of the Newton form (2.2), and $\mathbf{a}$ solves (1.4). By using the orthogonality of the p_j with respect to one of the inner products

$$\text{(3.3a)}\quad \langle \Psi_1, \Psi_2 \rangle_\rho := \frac{1}{2\pi} \int_{E_\rho} \overline{\Psi_1(z)} \Psi_2(z) |z^2 - 4\rho|^{-1/2} |dz|, \qquad 0 \le \rho < 1,$$

$$\text{(3.3b)}\quad \langle \Psi_1, \Psi_2 \rangle_\rho := \frac{1}{\pi} \int_{-2}^{2} \Psi_1(x) \Psi_2(x) |x^2 - 4|^{-1/2} dx, \qquad \rho = 1,$$

we can bound the mapping M_2 in a fairly straightforward manner. The derivation of a bound for M_1 requires more work and will be discussed first. Most of the proofs are just outlined; details can be found in [16].

Equip $\mathbb{C}^{n+1}$ with the uniform norm,

$$\|\mathbf{v}\|_\infty := \max_{0 \le j \le n} |v_j|, \qquad \mathbf{v} = (v_0, v_1, \ldots, v_n)^{\mathrm{T}},$$

and let $\|M_j\|_\infty$ denote the induced operator norm. In order to bound $\|M_1\|_\infty$, we first note that the divided differences $c_j^{(n)}$ can be written as

$$(3.4) \qquad c_{n-j}^{(n)} = \sum_{k=0}^{j} \frac{f_k}{\prod_{l=0,\, l\neq k}^{j}(z_k - z_l)}, \qquad 0 \le j \le n,$$

(see, e.g., Davis [2, §2.6]). A lower bound for the products in (3.4) was derived in [7] for z_k given by (1.9) and $0 \le \rho < 1$. This bound is used in the proof of the following theorem.

Theorem 3.1. *Let the nodes z_k, $0 \le k \le n$, be defined by (1.9) for an arbitrary constant $\alpha \in \mathbb{R}$, and let $0 \le \rho < 1$. Then there are nonnegative constants d_1, d_2 depending on ρ, but independent of n, such that*

$$(3.5) \qquad \|M_1\|_\infty \le d_1(n+1)^{d_2}, \qquad n \ge 1.$$

Proof. By (3.4),

$$(3.6) \qquad \|M_1\|_\infty = \max_{\|\mathbf{f}\|_\infty = 1} \|\mathbf{c}\|_\infty \le \max_{0\le j\le n} \sum_{k=0}^{j} \sum_{\substack{l=0\\ l\neq k}}^{j} |z_k - z_l|^{-1}.$$

The right-hand side of (3.6) can be bounded by applying Lemma 2.5 of [7], and (3.5) follows. □

In a sequence of lemmas we now present some auxiliary results that we use to bound $\|M_1\|_\infty$ for $\rho = 1$ and the nodes (1.10):

Lemma 3.1. *Let the nodes z_k be defined by (1.10). Then*

$$(3.7) \qquad \prod_{\substack{j=0\\ j\neq k}}^{2^l} |z_k - z_j| \ge 2^{l+1}, \qquad 0 \le k \le 2^l,$$

for any integer $l \ge 0$.

Proof. Let $U_n(x) := \sin((n+1)\theta)/\sin\theta$, where $x = \cos\theta$, denote Chebyshev polynomials of the second kind. Then

$$(3.8) \qquad \prod_{j=0}^{2^l}(z - z_j) = (z^2 - 4)U_{2^l-1}\left(\frac{z}{2}\right), \qquad -2 \le z \le 2.$$

By evaluating, and then estimating, the derivative of (3.8) at $z = z_k$ we obtain (3.7). □

Let the nodes z_k be defined by (1.10). Then Lemma 3.1 yields

$$(3.9) \qquad \prod_{\substack{j=0\\ j\neq k}}^{n} |z_k - z_j|^{-1} \le 2^{-l-1} \prod_{j=n+1}^{2^l} |z_k - z_j|, \qquad 0 \le k \le n,$$

where as usual the empty product is defined to have value one. The lemma below bounds the right-hand side of (3.9).

Lemma 3.2. *Let the nodes* z_j *be defined by* (1.10) *and assume that* $1 \leq n < 2^l$. *Then*

$$\prod_{m=n+1}^{2^l} |z_k - z_m| \leq 2^{l(l+1)}. \tag{3.10}$$

Proof. The bound is obtained by partitioning the product (3.10) into subproducts, each of which contains factors $|z_k - z_m|$ with nodes z_m that are distributed like the first 2^l points determined by (1.10) for some integer $l \geq 0$. Such a partitioning is described by [6, Lemma 2.4]. Each subproduct can be bounded, and (3.10) is obtained. □

We are now in a position to show a result for the nodes (1.10) analogous to Theorem 3.1.

Theorem 3.2. *Let the nodes* z_k, $0 \leq k \leq n$, *be defined by* (1.10), *and let* $\rho = 1$. *Then*

$$\|M_1\|_\infty \leq 2n^{3+\log_2 n}, \qquad n \geq 1. \tag{3.11}$$

Proof. From (3.6) we obtain

$$\begin{aligned} \|M_1\|_\infty &\leq \max_{0\leq j\leq n} \sum_{k=0}^{j} \sum_{\substack{m=0\\ m\neq k}}^{j} |z_k - z_m|^{-1} \\ &= \max\left\{1, \max_{1\leq j\leq n} \sum_{k=0}^{j} \sum_{\substack{m=0\\ m\neq k}}^{j} |z_k - z_m|^{-1}\right\}. \end{aligned} \tag{3.12}$$

Let $l \geq 0$ be the unique integer such that $n < 2^l \leq 2n$. An application of (3.7) and (3.10) yields

$$\begin{aligned} \max_{1\leq j\leq n} \sum_{k=0}^{j} \prod_{\substack{m=0\\ m\neq k}}^{j} |z_k - z_m|^{-1} &= \max_{1\leq j\leq n} \sum_{k=0}^{j} \frac{\prod_{m=j+1}^{2^l} |z_k - z_m|}{\prod_{m=0,\, m\neq k}^{2^l} |z_k - z_m|} \\ &\leq \max_{1\leq j\leq n} 2^{-l-1} \sum_{k=0}^{j} \prod_{m=j+1}^{2^l} |z_k - z_m|^{-1} \leq \max_{1\leq j\leq n} 2^{-l-1} \sum_{k=0}^{j} 2^{l(l+1)} \\ &= \frac{1}{2}(n+1)2^{l\cdot l} \leq \frac{1}{2}(n+1)(2n)^l \leq (n+1)n^{2+\log_2 n}, \qquad n \geq 1. \end{aligned}$$

Substitution of this inequality into (3.12) shows (3.11). □

Remark 3.1. In [7] the stability of the Newton interpolation formula is discussed for interpolation at nodes on a smooth Jordan curve. The nodes considered are Fejér points ordered by the van der Corput sequence, such as the nodes (1.9). A mapping T is defined that maps the vector $\mathbf{f} := (f_0, f_1, \ldots, f_n)^{\mathrm{T}}$ to the polynomial $q_n \in \Pi_n$ in Newton form (cf. (2.2)). The range and domain of T

are equipped with the uniform norm, and it is shown in [7, Theorem 2.6] that $\lim_{n\to\infty} \operatorname{cond}(T)^{1/n} = 1$, where $\operatorname{cond}(T)$ denotes the condition number of T. By using Theorem 3.2, this equality can also be shown when interpolation is carried out at the nodes (1.10). This result follows by substituting (3.11) into the proof of [7, Theorem 2.6]. □

We now derive a bound for $\|M_2\|_\infty$. This is achieved by first bounding the products $\prod_{k=0}^{j-1}(z - z_k)$ in the Newton form (2.2), and then using the orthogonality of the p_j with respect to one of the inner products (3.3).

Lemma 3.3. *Let* $0 \le \rho < 1$ *and let the nodes* z_k, $0 \le k < r$, *be defined by* (1.9). *Then*

$$\prod_{k=0}^{r-1} |z - z_k| \le 4r^2, \qquad r \ge 1, \quad z \in \partial E_\rho. \tag{3.13}$$

If, instead, the nodes z_k, $0 \le k \le r$, *are defined by* (1.10), *then*

$$\prod_{k=0}^{r} |z - z_k| \le 4r^{\log_2 2r}, \qquad r \ge 1, \quad z \in [-2, 2]. \tag{3.14}$$

Proof. The product (3.13) is partitioned into subproducts, each of which contains 2^l nodes z_k that are distributed roughly like the first 2^l nodes (1.9). Such a partitioning is described by [6, Lemma 2.3]. These subproducts can be bounded, and (3.13) is obtained. The proof of (3.14) is analogous. □

We are now in a position to bound the mapping M_2. The bounds show that the norm of M_2 grows fairly slowly with n.

Theorem 3.3. *Let the nodes* z_k, $0 \le k \le n$, *be defined by* (1.9), *and let* $0 \le \rho < 1$. *Then*

$$\|M_2\|_\infty \le 8n^3, \qquad n \ge 2. \tag{3.15}$$

If, instead, the nodes z_k, $0 \le k \le n$, *are given by* (1.10), *and* $\rho = 1$, *then*

$$\|M_2\|_\infty \le 10n^{2+\log_2 n}, \qquad n \ge 2. \tag{3.16}$$

Proof. Let the values of ρ in the definition (1.2) of the polynomials p_j and in the inner product $\langle \cdot, \cdot \rangle_\rho$ given by (3.3) be identical. Then

$$\langle p_j, p_k \rangle_\rho = 0, \quad k \ne j, \qquad 1 \le \langle p_j, p_j \rangle_\rho \le 2, \quad j \ge 0. \tag{3.17}$$

We obtain from (2.1) and the orthogonality of the p_j that

$$a_j^{(n)} = \langle q_n, p_j \rangle_\rho / \langle p_j, p_j \rangle_\rho, \qquad 0 \le j \le n. \tag{3.18}$$

Now (3.2), (3.17), and (3.18) show that

$$\|M_2\|_\infty = \max_{\|\mathbf{c}\|_\infty = 1} \|\mathbf{a}\|_\infty \le \max_{\|\mathbf{c}\|_\infty = 1} \max_{0 \le j \le n} |\langle q_n, p_j \rangle_\rho|. \tag{3.19}$$

Substituting

$$|\langle q_n, p_j \rangle_\rho| \le \max_{z\in\partial E_\rho} |q_n(z)| \max_{z\in\partial E_\rho} |p_j(z)|, \qquad \max_{z\in\partial E_\rho} |p_j(z)| \le 2, \tag{3.20}$$

and (2.2) into (3.19) yields

$$\|M_2\|_\infty \le 2 \max_{\|\mathbf{c}\|=1} \max_{z\in\partial E_\rho} |q_n(z)| \le 2 \max_{z\in\partial E_\rho} \sum_{j=0}^{n} \prod_{k=0}^{j-1} |z - z_k|. \tag{3.21}$$

The right-hand side of (3.21) can be bounded, using Lemma 3.3, and the theorem follows. □

We note that formulas similar to (3.15)–(3.16) are valid also if the Chebyshev polynomials p_j are replaced by polynomials that belong to some other family of orthogonal polynomials, such as Legendre polynomials. The proof of Theorem 3.3 only requires that an inequality of the form (3.17) is valid.

Theorem 3.4. *Let* $V_\rho \in \mathbb{C}^{(n+1)\times(n+1)}$. *Assume that* $0 \le \rho < 0$, *and let the nodes* z_k *be given by* (1.9). *Then there are constants* c *and* d *depending on* ρ, *but independent of* n, *such that*

$$\kappa_\infty(V_\rho^{\mathrm{T}}) \le cn^d, \qquad n \ge 1.$$

If, instead, $\rho = 1$, *and the nodes* z_k *are given by* (1.10), *then*

$$\kappa_\infty(V_1^{\mathrm{T}}) \le 40n^{5+2\log_2 n}, \qquad n \ge 1.$$

Proof. From the second inequality in (3.20) it follows that $\|V_\rho^{\mathrm{T}}\|_\infty \le 2(n+1)$. The factorization $V_\rho^{-\mathrm{T}} = M_2 M_1$ and Theorems 3.1–3.3 yield bounds for $V_\rho^{-\mathrm{T}}$, and the theorem follows. □

Theorem 3.4 shows that the condition number grows slower than exponentially with n for nodes (1.9) and (1.10). The bounds in Theorems 3.1–3.4 are not sharp, however, and the numerical experiments of §4 display a quite modest growth of $\kappa_\infty(V_\rho^{\mathrm{T}})$ with n. Bounds for $\kappa_\infty(V_\rho) = \|V_\rho\|_\infty \|V_\rho^{-1}\|_\infty$ can be obtained by Theorem 3.4 and the observation that for any matrix $A \in \mathbb{C}^{(n+1)\times(n+1)}$,

$$\|A^{\mathrm{T}}\|_\infty = \|A\|_1 \le (n+1)\|A\|_\infty. \tag{3.22}$$

We turn to the propagation of errors in the right-hand side vectors in (1.4) and (1.5) by Algorithms 1 and 2. A comparison of (2.5)–(2.8) with (3.1)–(3.2) yields, for $n \ge 1$,

$$\begin{aligned} \mathbf{e}_n^{\mathrm{T}} D_{n-1} L_{n-1} D_{n-2} L_{n-2} \cdots D_0 L_0 &= \mathbf{e}_0^{\mathrm{T}} M_1, \\ W_{n-1} W_{n-2} \cdots W_0 \mathbf{e}_0 &= M_2 \mathbf{e}_0, \\ S_n &= M_2 \mathbf{e}_0 \mathbf{e}_0^{\mathrm{T}} M_1. \end{aligned} \tag{3.23}$$

The following theorem shows that the propagated errors in the solution vectors of (1.4) and (1.5), due to perturbations in the right-hand side vectors **f** and **g**, grow slower than exponentially with n. We remark that for many distributions and orderings of nodes z_k, the propagated error does, indeed, grow exponentially with n (see the numerical examples of §4).

Theorem 3.5. *Let $0 \le \rho < 0$, and let the nodes z_k be given by* (1.9). *Then there are constants c_1 and d_1 independent of n such that*

$$\max \left\{ \sum_{j=0}^{n} \|S_j\|_\infty , \sum_{j=0}^{n} \|S_j^{\mathrm{T}}\|_\infty \right\} \le c_1 n^{d_1} .$$

If, instead, $\rho = 1$, and the nodes z_k are given by (1.10), *then there are constants c_2 and d_2 independent of n such that*

$$\max \left\{ \sum_{j=0}^{n} \|S_j\|_\infty , \sum_{j=0}^{n} \|S_j^{\mathrm{T}}\|_\infty \right\} \le c_2 n^{d_2 \log_2(n)} .$$

Proof. The proof follows from $\|S_0\|_\infty = 1$, bounds for $\|M_1\|_\infty$ and $\|M_2\|_\infty$, and (3.22)–(3.23). □

4. NUMERICAL EXAMPLES

The computed examples of this section illustrate the results of §3. All examples have been computed on an IBM 3090VF computer. Throughout this section the parameter α in (1.9) is set to zero.

Example 4.1. This example shows $\kappa_2(V_\rho)$ $(= \kappa_2(V_\rho^{\mathrm{T}}))$ as a function of n for different values of ρ, where $V_\rho \in \mathbb{C}^{(n+1)\times(n+1)}$. The condition numbers $\kappa_2(V_\rho)$ have been computed in double precision arithmetic, i.e., with 15 significant digits, using the subroutine ZSVDC of LINPACK [3]. Figure 4.1.1 illustrates the oscillating behavior of $n \to \kappa_2(V_0)$, where V_0 is defined by the nodes (1.9). The condition number $\kappa_2(V_0)$ is smallest when n is such that the set of nodes $\{z_k\}_{k=0}^{n}$ can be written as the union of only a few disjoint sets of equidistant nodes. For instance, if $n = 2^l - 1$ for some integer $l \ge 0$, then the z_k are the nth roots of unity and V_0 is orthogonal, i.e., $\kappa_2(V_0) = 1$.

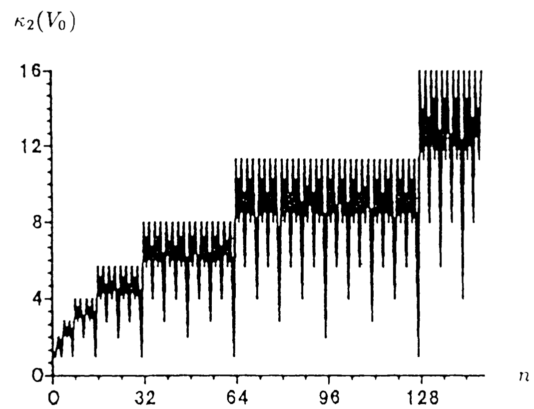

FIGURE 4.1.1
Condition number $\kappa_2(V_0)$ as a function of n

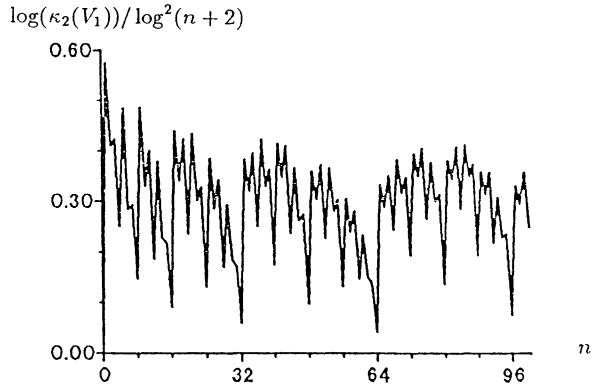

FIGURE 4.1.2
Growth of $\kappa_2(V_1)$ *with* n

In Figure 4.1.2 we have chosen $\rho = 1$ and the nodes (1.10). This figure suggests that $\kappa_2(V_1) < n^{0.45\log(n)}$ for large n, where log denotes the natural logarithm.

The following examples illustrate the propagation of roundoff errors in Algorithms 1 and 2. Because of the small amplification of roundoff errors when the nodes (1.9) and (1.10) are used, we are able to solve fairly large CV systems (1.4) and (1.5) in single precision arithmetic, i.e., with only six significant digits.

Example 4.2. In this example we solve dual CV systems (1.4) by Algorithm 1 in single precision arithmetic. Let $\mathbf{x} \in \mathbb{C}^{n+1}$ denote the exact solution of (1.4), and let $\mathbf{x}^*$ denote the computed solution. We determine the residual error

$$\mathbf{r} := V_\rho^{\mathrm{T}}\mathbf{x}^* - \mathbf{f} \tag{4.1}$$

by accumulating sums in double precision arithmetic. The norm $\|\mathbf{r}\|_\infty$ is a good estimate for the norm of the error in the solution $\|\mathbf{x}^* - \mathbf{x}\|_\infty$ because V^{T} is quite well-conditioned (see Example 4.1).

Figure 4.2.1 shows $\|\mathbf{r}\|_\infty$ when $\rho = 0$ and the nodes are defined by (1.9). The real and imaginary parts of the right-hand side $\mathbf{f} \in \mathbb{C}^{n+1}$ are uniformly distributed elements in $[0, 1]$, computed by the random number generator SURAND of the ESSL program library [5]. The figure shows a slow growth of $\|\mathbf{r}\|_\infty$ with n.

The computations for Figures 4.2.1 and 4.2.2 differ only in the ordering of the nodes. The matrix $V_0^{\mathrm{T}} \in \mathbb{C}^{(n+1)\times(n+1)}$ used for Figure 4.2.2 is defined by the nodes

$$z_k = \exp(2\pi ik/(n+1)), \qquad 0 \le k \le n, \tag{4.2}$$

for every $n \ge 1$. The nodes (4.2) make V_0^{T} orthogonal for every n, but yield severe amplification of roundoff errors, as shown by Figure 4.2.2. The rapid

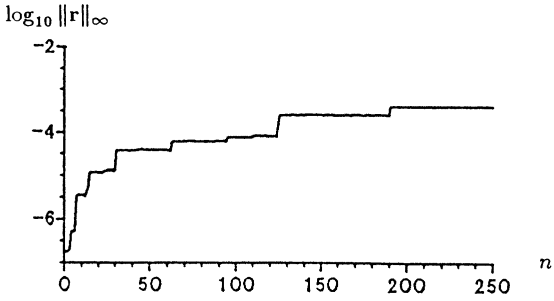

FIGURE 4.2.1
Growth of $\|\mathbf{r}\|_\infty$ *with* n *for* $\rho = 0$ *and nodes* (1.9)

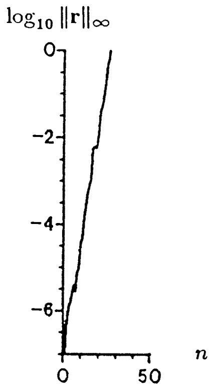

FIGURE 4.2.2
Growth of $\|\mathbf{r}\|_\infty$ *with* n *for* $\rho = 0$ *and nodes* (4.2)

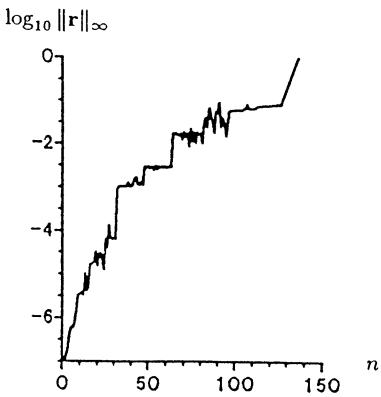

FIGURE 4.2.3
Growth of $\|\mathbf{r}\|_\infty$ *with* n *for* $\rho = 1$ *and nodes* (1.10)

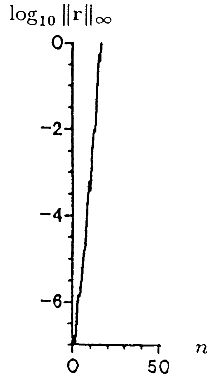

FIGURE 4.2.4
Growth of $\|\mathbf{r}\|_\infty$ *with* n *for* $\rho = 1$ *and nodes* (4.3)

growth of the propagated error is a result of the ordering of the nodes (4.2), which is unsuitable because it causes many products in the denominators in (3.4) to have tiny magnitude.

Experiments suggest that the graphs of Figures 4.2.1–4.2.2 are quite insensitive to the choice of right-hand side vector $\mathbf{f}$, as well as to the choice of $0 \le \rho < 1$, if ρ is not very close to 1.

In Figures 4.2.3 and 4.2.4 we set $\rho = 1$ and present graphs analogous to those of Figures 4.2.1–4.2.2. Figure 4.2.3 shows $\|\mathbf{r}\|_\infty$ when the nodes z_j are given by (1.10) and $\mathbf{f} = [f_j]_{j=0}^n$ has elements uniformly distributed in $[0, 1]$. The computations for Figure 4.2.4 differ from those for Figure 4.2.3 only in the ordering of the nodes. For Figure 4.2.4 we select for every $n \ge 1$ the nodes

$$z_k := 2\cos(\pi k/n), \qquad 0 \le k \le n. \tag{4.3}$$

Example 4.3. We consider the solution of primal CV systems (1.5) by Algorithm 2 using single precision arithmetic. Let $\mathbf{x}^* \in \mathbb{C}^{n+1}$ denote the computed solution, and define the residual error

$$\mathbf{r}' := V_\rho \mathbf{x}^* - \mathbf{g}. \tag{4.4}$$

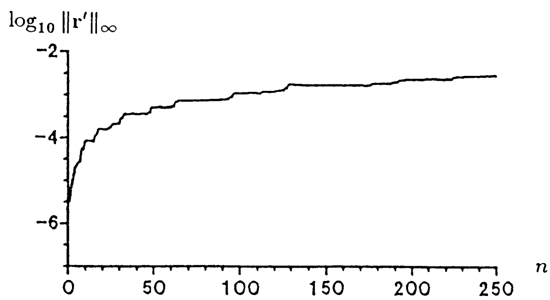

FIGURE 4.3.1
Growth of $\|\mathbf{r}'\|_\infty$ *with* n *for* $\rho = 0.8$ *and nodes* (1.9)

Figure 4.3.1 shows $\|\mathbf{r}'\|_\infty$ for $\rho = 0.8$ and the nodes (1.9). The right-hand side $\mathbf{g} = [g_j]_{j=0}^n$ is given by $g_j := \exp(z_j)$.

Numerous numerical experiments indicate that the residual error (4.4) often is somewhat larger than the error (4.1) for identical matrices V_ρ and right-hand sides. Further computed examples can be found in [16].

5. Conclusions

Fast progressive algorithms are derived for the solution of CV systems, and in §§3 and 4 these algorithms are demonstrated to be fairly insensitive to perturbations for suitably distributed and ordered nodes.

Acknowledgment

We would like to thank Åke Björck and Arnold Schönhage for helpful discussions.

Addendum. Figure 4.1.1 inspired A. Córdova, W. Gautschi, and S. Ruscheweyh to completely describe the spectrum and eigenvectors of $V_0 V_0^H$ in the paper *Vandermonde matrices on the circle: spectral properties and conditioning*, Numer. Math. **57** (1990), 577–591. In particular, they show that $\kappa_2(V_0) = O(n^{1/2})$ for $V_0 \in \mathbb{C}^{(n+1)\times(n+1)}$ defined by the nodes (1.9) with $\rho = 0$. A survey of condition number bounds for Vandermonde matrices can be found in the paper *How (un)stable are Vandermonde systems?* by W. Gautschi, in Asymptotic and Computational Analysis (R. Wong, ed.), Lecture Notes in Pure and Appl. Math., vol. 124, Dekker, New York, 1990, pp. 193–210.

Bibliography

1. Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp. **24** (1970), 893–903.
2. P. J. Davis, *Interpolation and approximation*, Dover, New York, 1975.
3. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *Linpack users' guide*, SIAM, Philadelphia, PA, 1979.
4. S. W. Ellacott, *Computation of Faber series with application to numerical polynomial approximation in the complex plane*, Math. Comp. **40** (1984), 575–587.
5. *Engineering and scientific subroutine library*, IBM, 1987.
6. B. Fischer and L. Reichel, *A stable Richardson iteration method for complex linear systems*, Numer. Math. **54** (1988), 225–242.
7. ——, *Newton interpolation in Fejér and Chebyshev points*, Math. Comp. **53** (1989), 265–278.
8. W. Gautschi and G. Inglese, *Lower bounds for the condition number of Vandermonde matrices*, Numer. Math. **52** (1988), 241–250.
9. W. Gautschi, *The condition of Vandermonde-like matrices involving orthogonal polynomials*, Linear Algebra Appl. **52/53** (1983), 293–300.
10. ——, *The condition of polynomials in power form*, Math. Comp. **33** (1979), 343–352.
11. ——, *Optimally conditioned Vandermonde matrices*, Numer. Math. **24** (1975), 1–12.

12. G. H. Golub and C. F. Van Loan, *Matrix computations*, 2nd ed., Johns Hopkins Univ. Press, Baltimore, MD, 1989.

13. N. J. Higham, *Fast solution of Vandermonde-like systems involving orthogonal polynomials*, IMA J. Numer. Anal. **8** (1988), 473–486.

14. ——, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl. **11** (1990), 23–41.

15. E. Hlawka, *The theory of uniform distribution*, AB Academic Publishers, Berkhamsted, 1984.

16. L. Reichel and G. Opfer, *Chebyshev-Vandermonde systems*, BSC Report 88/48, Bergen Scientific Centre, Bergen, Norway, 1988.

17. V. I. Smirnov and N. A. Lebedev, *Functions of a complex variable*, Iliffe Books, London, 1968.

18. W. P. Tang and G. H. Golub, *The block decompositions of a Vandermonde matrix and its applications*, BIT **21** (1981), 505–517.

Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506

Institute of Applied Mathematics, University of Hamburg, Bundesstrasse 55, D-2000 Hamburg 13, Germany