Numer. Math. 50, 613-632 (1987)

Numerische Mathematik
© Springer-Verlag 1987

Error Analysis of the Björck-Pereyra Algorithms for Solving Vandermonde Systems

Nicholas J. Higham

Department of Mathematics, University of Manchester, Manchester M13 9PL, UK

Summary. A forward error analysis is presented for the Björck-Pereyra algorithms used for solving Vandermonde systems of equations. This analysis applies to the case where the points defining the Vandermonde matrix are nonnegative and are arranged in increasing order. It is shown that for a particular class of Vandermonde problems the error bound obtained depends on the dimension n and on the machine precision only, being independent of the condition number of the coefficient matrix. By comparing appropriate condition numbers for the Vandermonde problem with the forward error bounds it is shown that the Björck-Pereyra algorithms introduce no more uncertainty into the numerical solution than is caused simply by storing the right-hand side vector on the computer. A technique for computing "running" a posteriori error bounds is derived. Several numerical experiments are presented, and it is observed that the ordering of the points can greatly affect the solution accuracy.

Subject Classifications: AMS(MOS): 65G05, 65F05; CR: G1.3.

1. Introduction

A Vandermonde matrix may be defined in terms of a set of distinct scalars $\alpha_0, \alpha_1, \ldots, \alpha_n \in \mathbb{C}$ by

$$V = V(\alpha_0, \alpha_1, \ldots, \alpha_n) = \begin{bmatrix} 1 & 1 \ldots & 1 \\ \alpha_0 & \alpha_1 & \alpha_n \\ \vdots & \vdots & \vdots \\ \alpha_0^n & \alpha_1^n \ldots & \alpha_n^n \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n+1)}. \tag{1.1}$$

The associated linear systems

$$\text{Primal:} \quad Vx = b, \tag{1.2}$$

$$\text{Dual:} \quad V^T a = f, \tag{1.3}$$

arise in a variety of applications. Some examples are polynomial interpolation, numerical differentiation [15], approximation of linear functionals [3, 4, 14] rational Chebyshev approximation [2, 8], and polynomial root finding [19]. In some of these applications many Vandermonde systems may have to be solved, so an efficient method is desirable for their solution.

The standard method for solving dense systems of linear equations, Gaussian elimination, requires $O(n^3)$ arithmetic operations and $O(n^2)$ elements of storage when applied to (1.2) or (1.3). Several authors have attempted to exploit the structure of the Vandermonde matrix in order to derive more efficient solution methods. Lyness and Moler [15] derive via Neville interpolation an algorithm for solving the dual system (1.3); the algorithm has an operation count of approximately $\frac{1}{3}n^3(M+2A)$, where A denotes an addition or subtraction and M a multiplication or division, and it requires $\frac{1}{2}n^2$ elements of storage. Ballester and Pereyra [3] show how the primal system (1.2) can be solved in $\frac{1}{2}(3n+2)(n+1)(M+A)$ and $\frac{1}{2}n^2$ elements of storage by using recursion formulae that enable the LU factors of V to be computed in $O(n^2)$ operations (see also [9]). Still more efficient algorithms are derived by Björck and Pereyra [5], for both the dual and the primal problems; these algorithms have an operation count of $\frac{1}{2}n(n+1)(2M+3A)$ and require no storage over and above that needed for the points α_i and for the right-hand side, which is transformed into the solution vector. More recently, Tang and Golub [18] have derived a block decomposition method that is well suited to parallel computation and to the solution of interpolation problems with complex conjugate interpolation points where the coefficients of the interpolating polynomial are real. For the block size 1, the method of [18] is similar, but not equivalent, to the Björck-Pereyra algorithms.

To the author's knowledge no rounding error analyses have been published for the algorithms cited above. Such analyses are desirable in order to reveal potential instabilities in the algorithms and to enable the user to estimate the accuracy of computed solutions. These considerations are especially pertinent when solving Vandermonde systems, for it is widely appreciated that Vandermonde matrices tend to have large condition numbers $\kappa(V)$ [11, 12], where $\kappa(A)=\|A\|\,\|A^{-1}\|$ is the condition number of A with respect to inversion. Indeed, large values for $\kappa(V)$, and inaccurate computed solutions to (1.2) and (1.3), have been observed in practical problems [2, 8]. It is known, however, that accurate solution of a Vandermonde system is not necessarily precluded when $\kappa(V)$ is large. Björck and Pereyra [5] present a test problem for which their algorithm gives a very accurate computed solution, even though the associated Vandermonde matrix has a large condition number. They state in [5] (see also [13, p. 123]):

"It seems as if at least some problems connected with Vandermonde systems, which traditionally have been considered too ill-conditioned to be attacked, actually can be solved with good precision".

In this paper we present an a priori rounding error analysis of the Björck-Pereyra algorithms for the case where the points α_i are real, nonnegative and arranged in increasing order. This analysis, in Sect. 2, provides an element-wise bound for the vector of relative errors in the computed solution. The implications of the bound are explored in Sect. 3. In particular it is shown that for a particular class of Vandermonde problems the bound depends only on n and on the machine precision, being independent of the condition number $\kappa(V)$. Our analysis therefore provides an explanation for the phenomenon, observed in [5], of highly accurate computed solutions.

In Sect. 4 condition numbers are derived for systems (1.2) and (1.3), appropriate to the situation where the data ($\alpha_0, \alpha_1, \ldots, \alpha_n$ and the right-hand side vector) are subject to small relative perturbations. By comparison with the error analysis it is shown that if the points α_i are nonnegative and in increasing order, then the Björck-Pereyra algorithms introduce no more uncertainty into the numerical solution than is caused simply by storing the right-hand side vector on the computer.

The practical computation of error bounds is considered in Sect. 5. Here, a "running" error bound technique is described that is applicable to any distribution of the points. Numerical experiments are presented in Sect. 6 in order to illustrate the error analysis and to investigate several interesting questions raised in the preceding sections.

To conclude the introduction, we state the algorithms to be analysed. For a derivation of the algorithms, see [5] or [13 Sect. 5.6], and for Algol 60 procedures see [5]. Our notation follows that of [5] except that we adopt the convention that an element '$z_j^{(k)}$' (j'th element at the k'th stage) not formally defined in an algorithm takes the value '$z_j^{(k-1)}$', that is, undefined elements retain their value from the previous stage.

Algorithm 1. Primal ($Vx=b$).

Stage I: $d_j^{(0)}=b_j \; (j=0,\ldots,n)$

For $k=0$ to $n-1$

$$\lfloor d_j^{(k+1)}=d_j^{(k)}-\alpha_k d_{j-1}^{(k)} \quad (j=n,n-1,\ldots,k+1) \tag{1.4}$$

Stage II: $x_j^{(n)}=d_j^{(n)} \; (j=0,\ldots,n)$

For $k=n-1$ to 0 step -1

$$x_j^{(k+\frac{1}{2})}=\frac{x_j^{(k+1)}}{\alpha_j-\alpha_{j-k-1}} \qquad (j=k+1,\ldots,n) \tag{1.5a}$$

$$x_j^{(k)}=x_j^{(k+\frac{1}{2})}-x_{j+1}^{(k+\frac{1}{2})} \qquad (j=k,\ldots,n-1) \tag{1.5b}$$

The solution is $x_j=x_j^{(0)} \; (j=0,\ldots,n)$.

Algorithm 2. Dual ($V^T a=f$)

Stage I: $c_j^{(0)}=f_j \; (j=0,\ldots,n)$

For $k=0$ to $n-1$

$$\left\lfloor c_j^{(k+1)}=\frac{c_j^{(k)}-c_{j-1}^{(k)}}{\alpha_j-\alpha_{j-k-1}} \right. (j=n,n-1,\ldots,k+1) \tag{1.6}$$

Stage II: $a_j^{(n)}=c_j^{(n)} \; (j=0,\ldots,n)$

$$\text{For } k=n-1 \text{ to } 0 \text{ step } -1$$

$$\lfloor a_j^{(k)}=a_j^{(k+1)}-\alpha_k a_{j+1}^{(k+1)} (j=k,\dots,n-1). \tag{1.7}$$

The solution is $a_j=a_j^{(0)}$ $(j=0,\dots,n)$.

Note that stage I of Algorithm 2 is the standard method for evaluating divided differences $(c_k^{(k)}=f[\alpha_0,\dots,\alpha_k])$. It is hoped that the analysis presented here will, in particular, be of help in understanding the propagation of rounding errors in divided difference schemes.

The reader may find it helpful to visualise the Björck-Pereyra algorithms in terms of a "flow diagram", as illustrated in Table 1.1 for the dual algorithm. An ' × ' denotes a component that has (potentially) changed from the previous step.

Table 1.1. Dual

j	$f=c^{(0)}$	$c^{(1)}$	…	$c^{(n-1)}$	$c^{(n)}$	$a^{(n-1)}$	$a^{(n-2)}$	…	$a^{(1)}$	$a^{(0)}=a$
0	×									×
1	×	×							×	×
2	×	×	.					.	×	×
.	.	.	. .					.	.	.
.	.	.	.	.			.		.	.
.	.	.		.	.		×		.	.
$n-1$	×	×	…	×	.	×	×	…	×	×
n	×	×	…	×	×					
	Stage I					Stage II				

2. Error Analysis

In this section we perform an a priori rounding error analysis for Algorithms 1 and 2. The analysis is a forward error analysis, rather than the backward type more commonly applied to linear equation solvers; we therefore bound directly the error in a computed solution rather than show that the computed solution is the exact solution of a perturbed problem.

We will assume that the points α_i in (1.1) are real numbers satisfying

$$0 \leqq \alpha_0 < \alpha_1 < \dots < \alpha_n. \tag{2.1}$$

This assumption assures the validity of certain sign properties upon which our analysis relies. The requirement that the points are in increasing order can always be satisfied by a simple re-ordering of the columns of the primal system, or the rows of the dual. However, it is important to realise that Algorithms 1 and 2 are not invariant under such permutations: the rounding errors committed, and the way in which they propagate, depends on the ordering of the points. For example, in stage I of the dual algorithm, $c_j^{(1)}$ $(j=1,\dots,n)$ depends on the ordering π, where $\pi=(\pi_i)$ is a permutation of the integers $0, 1, \dots, n$, through

$$c_j^{(1)} = \frac{f_{\pi_j} - f_{\pi_{j-1}}}{\alpha_{\pi_j} - \alpha_{\pi_{j-1}}}.$$

We comment further on assumption (2.1) in Sect. 3.

To begin, we present two preliminary lemmas which display some important sign properties of the Vandermonde systems, and of Algorithms 1 and 2. The modulus of a vector or matrix, respectively, is defined by $|x| = (|x_i|)$, $|A| = (|a_{ij}|)$.

Lemma 2.1. *Let the points α_i satisfy* (2.1). *If $(-1)^i b_i \geqq 0$ and $(-1)^i f_i \geqq 0$, for all i, then*

$$|V^{-1} b| = |V^{-1}|\,|b|,$$
$$|V^{-T} f| = |V^{-T}|\,|f|.$$

Proof. Let $V^{-1} = (w_{ij})$ $0 \leqq i, j \leqq n$. It is well known [20] that

$$w_{ij} = (-1)^{n-j} \sigma_{n-j}(\alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n) \left(\prod_{\substack{k=0 \\ k \neq i}}^{n} (\alpha_i - \alpha_k) \right)^{-1}, \tag{2.2}$$

where $\sigma_k(y_1, \dots, y_n)$ denotes the sum of all distinct products of k of the arguments $y_1, \dots, y_n$ (that is, σ_k is the k'th elementary symmetric function). From (2.1) and (2.2), if $w_{ij} \neq 0$,

$$\operatorname{sign}(w_{ij}) = (-1)^{n-j} \times 1 \times (-1)^{n-i} = (-1)^{i+j}.$$

Thus, in the inner product $(V^{-1} b)_i = \sum_{j=0}^{n} w_{ij} b_j$, every nonzero term has the same sign, $(-1)^{i+j} \times (-1)^j = (-1)^i$, so that $|(V^{-1} b)_i| = \sum_{i=0}^{n} |w_{ij}|\,|b_i|$, as required. Similarly for $V^{-T} f$. □

Thus, if the components of b and f alternate in sign, cancellation through subtraction cannot occur in forming the products $V^{-1} b$ and $V^{-T} f$, the solutions to the primal and the dual Vandermonde systems respectively. In our error analysis we will exploit the fact that a similar "no cancellation" property holds for solution via Algorithms 1 and 2.

Lemma 2.2. *Let the points α_i satisfy* (2.1) *and let the vectors b and f satisfy $(-1)^i b_i \geqq 0$, $(-1)^i f_i \geqq 0$, for all i.*

Then, in Algorithm 1,

$$(-1)^j x_j^{(k+\frac{1}{2})}, \quad \left.\begin{matrix} (-1)^j d_j^{(k)} \geqq 0 \\ (-1)^j x_j^{(k)} \geqq 0 \end{matrix}\right\} k = 0, \dots, n, \; j = 0, \dots, n,$$

and in Algorithm 2,

$$\left.\begin{matrix} (-1)^j c_j^{(k)} \geqq 0 \\ (-1)^j a_j^{(k)} \geqq 0 \end{matrix}\right\} k = 0, \dots, n, \; j = 0, \dots, n.$$

Hence all additions/subtractions in Algorithms 1 *and* 2 *are in fact additions of numbers with the same sign.*

Proof. A straightforward induction using (1.4)-(1.7) and (2.1). □

We are now ready to perform the rounding error analysis. We assume that the floating point arithmetic satisfies [6, p. 9]

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\varepsilon), \qquad |\varepsilon| \leqq u, \tag{2.3}$$

where op $= +, -, *$ or $/$, and u is the unit roundoff.

Theorem 2.3. *Suppose Algorithms* 1 *and* 2 *are carried out in floating-point arithmetic with unit roundoff u, where the data α_i, b_i, f_i $(i=0,\dots,n)$ are floating-point numbers. Assume the points α_i satisfy* (2.1). *Then, provided no overflows are encountered, the algorithms run to completion, and the computed solutions $\hat{x}$, $\hat{a}$ satisfy*

$$|x-\hat{x}| \leqq 5nu|V^{-1}||b| + O(u^2),$$
$$|a-\hat{a}| \leqq 5nu|V^{-T}||f| + O(u^2).$$

Proof. We will prove the result for the dual algorithm only; the proof for the primal is very similar, but slightly longer due to the form of (1.5). To simplify the presentation we will omit index ranges from equations and inequalities.

First, note that in the absence of overflow, Algorithm 2 must run to completion, because assumptions (2.1) and (2.3) guarantee that $fl(\alpha_i-\alpha_j) \neq 0$ for all $i \neq j$, ensuring the successful evaluation of (1.6).

Let $\hat{c}_j^{(k)}$ and $\hat{a}_j^{(k)}$ denote the computed intermediate quantities in Algorithm 2, and define

$$\delta_j^{(k)} = \hat{c}_j^{(k)} - c_j^{(k)}, \tag{2.4}$$
$$\mu_j^{(k)} = \hat{a}_j^{(k)} - a_j^{(k)}. \tag{2.5}$$

Our approach is to find recurrence relations satisfied by the errors $\delta_j^{(k)}$, $\mu_j^{(k)}$ and thence to majorise $|\delta_j^{(k)}|$, $|\mu_j^{(k)}|$ in terms of the solution to a related Vandermonde system.

First, we obtain recurrences for $\delta_j^{(k)}$ and $\mu_j^{(k)}$. From (1.6) and (2.3),

$$\begin{aligned}\hat{c}_j^{(k+1)} &= fl\left(\frac{fl(\hat{c}_j^{(k)}-\hat{c}_{j-1}^{(k)})}{fl(\alpha_j-\alpha_{j-k-1})}\right)\\ &= \left(\frac{\hat{c}_j^{(k)}-\hat{c}_{j-1}^{(k)}}{\alpha_j-\alpha_{j-k-1}}\right)(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3), \qquad |\varepsilon_i| \leqq u, \quad i=1,2,3,\\ &= \left(\frac{\hat{c}_j^{(k)}-\hat{c}_{j-1}^{(k)}}{\alpha_j-\alpha_{j-k-1}}\right)(1+\eta), \qquad |\eta| \leqq 3u+O(u^2).\end{aligned} \tag{2.6}$$

Substituting for $\hat{c}_j^{(k)}$ from (2.4), and using (1.6), we find

$$\delta_j^{(k+1)} = \frac{\delta_j^{(k)}-\delta_{j-1}^{(k)}}{\alpha_j-\alpha_{j-k-1}} + \eta\left(c_j^{(k+1)} + \frac{\delta_j^{(k)}-\delta_{j-1}^{(k)}}{\alpha_j-\alpha_{j-k-1}}\right).$$

Taking moduli gives

$$|\delta_j^{(k+1)}| \leqq (1+3u)\frac{|\delta_j^{(k)}|+|\delta_{j-1}^{(k)}|}{\alpha_j-\alpha_{j-k-1}}+3u\,|c_j^{(k+1)}|+O(u^2). \tag{2.7}$$

Similarly, from (1.7) and (2.3), using (2.5),

$$\begin{aligned}\hat{a}_j^{(k)} &= fl(\hat{a}_j^{(k+1)}-fl(\alpha_k\,\hat{a}_{j+1}^{(k+1)}))\\ &= (\hat{a}_j^{(k+1)}-\alpha_k\,\hat{a}_{j+1}^{(k+1)}(1+\varepsilon_1))(1+\varepsilon_2), \qquad |\varepsilon_i|\leqq u, \quad i=1,2,\\ &= (a_j^{(k+1)}+\mu_j^{(k+1)}-\alpha_k(a_{j+1}^{(k+1)}+\mu_{j+1}^{(k+1)})(1+\varepsilon_1))(1+\varepsilon_2)\\ &= (a_j^{(k)}+\mu_j^{(k+1)}-\alpha_k\,\mu_{j+1}^{(k+1)})(1+\varepsilon_2)\\ &\quad -\alpha_k(a_{j+1}^{(k+1)}+\mu_{j+1}^{(k+1)})\,\varepsilon_1+O(u^2).\end{aligned} \tag{2.8}$$

Thus,

$$\begin{aligned}\mu_j^{(k)} &= \mu_j^{(k+1)}-\alpha_k\,\mu_{j+1}^{(k+1)}+\varepsilon_2(a_j^{(k)}+\mu_j^{(k+1)}-\alpha_k\,\mu_{j+1}^{(k+1)})\\ &\quad -\alpha_k(a_{j+1}^{(k+1)}+\mu_{j+1}^{(k+1)})\,\varepsilon_1+O(u^2),\end{aligned}$$

which gives, on taking moduli,

$$\begin{aligned}|\mu_j^{(k)}| \leqq |\mu_j^{(k+1)}|&+\alpha_k\,|\mu_{j+1}^{(k+1)}|+u\{|a_j^{(k)}|+\alpha_k\,|a_{j+1}^{(k+1)}|\\ &+|\mu_j^{(k+1)}|+2\alpha_k\,|\mu_{j+1}^{(k+1)}|\}+O(u^2),\end{aligned}$$

and this weakens to

$$|\mu_j^{(k)}| \leqq (1+2u)(|\mu_j^{(k+1)}|+\alpha_k\,|\mu_{j+1}^{(k+1)}|)+u(|a_j^{(k)}|+\alpha_k\,|a_{j+1}^{(k+1)}|)+O(u^2). \tag{2.9}$$

Now define

$$g_i=(-1)^i\,|f_i|, \qquad i=0,\dots,n, \tag{2.10}$$

and let $p_j^{(k)}, q_j^{(k)}$ denote the values from stage I and stage II respectively of Algorithm 2 with g as right-hand side. By Lemma 2.2 there is no cancellation in the evaluation of Algorithm 2 for g.

Hence,

$$|p_j^{(k+1)}| = \frac{|p_j^{(k)}|+|p_{j-1}^{(k)}|}{\alpha_j-\alpha_{j-k-1}}, \tag{2.11}$$

$$|q_j^{(k)}| = |q_j^{(k+1)}|+\alpha_k\,|q_{j+1}^{(k+1)}|. \tag{2.12}$$

Using (2.10)-(2.12) and (1.6), (1.7) it is easy to show by induction that for all j and k,

$$|c_j^{(k)}| \leqq |p_j^{(k)}|, \tag{2.13}$$

$$|a_j^{(k)}| \leqq |q_j^{(k)}|. \tag{2.14}$$

Relations (2.11)-(2.14) are the key to the result.

We claim that the errors $\delta_j^{(k)}$ and $\mu_j^{(k)}$ may be bounded as follows:

$$|\delta_j^{(k)}| \leqq 3k\,u\,|p_j^{(k)}|+O(u^2), \tag{2.15}$$

$$|\mu_j^{(k)}| \leqq (5n-2k)\,u\,|q_j^{(k)}|+O(u^2). \tag{2.16}$$

These inequalities are proved by induction on k. By definition, $\delta_j^{(k)} \equiv 0$; therefore, using (2.7) and (2.13),

$$|\delta_j^{(1)}| \leqq 3u\,|c_j^{(1)}| + O(u^2) \leqq 3u\,|p_j^{(1)}| + O(u^2).$$

Assume (2.15) holds for k. Then from (2.7), using (2.13) and (2.11),

$$\begin{aligned} |\delta_j^{(k+1)}| &\leqq (1+3u)\,3k\,u\left(\frac{|p_j^{(k)}| + |p_{j-1}^{(k)}|}{\alpha_j - \alpha_{j-k-1}}\right) + 3u\,|p_j^{(k+1)}| + O(u^2) \\ &= ((1+3u)\,3k+3)\,u\,|p_j^{(k+1)}| + O(u^2) \\ &= 3(k+1)\,u\,|p_j^{(k+1)}| + O(u^2). \end{aligned}$$

This proves (2.15). The bound (2.16) is the same as (2.15) for $k=n$. Assuming (2.16) is true for $k+1$, we have from (2.9), (2.14) and (2.12),

$$\begin{aligned} |\mu_j^{(k)}| &\leqq (1+2u)(5n-2(k+1))\,u(|q_j^{(k+1)}| + \alpha_k\,|q_{j+1}^{(k+1)}|) \\ &\quad + u(|q_j^{(k)}| + \alpha_k|q_{j+1}^{(k+1)}|) + O(u^2) \\ &\leqq (1+2u)(5n-2k-2)\,u\,|q_j^{(k)}| + 2u\,|q_j^{(k)}| + O(u^2) \\ &= (5n-2k)\,u\,|q_j^{(k)}| + O(u^2), \end{aligned}$$

completing the proof of (2.16).

Finally, we take $k=0$ in (2.16) to obtain

$$|a_j - \hat{a}_j| = |\mu_j^{(0)}| \leqq 5nu\,|q_j^{(0)}| + O(u^2),$$

and we observe from (2.10) and Lemma 2.1 that

$$\begin{aligned} |q^{(0)}| &= |V^{-T}g| = |V^{-T}|\,|g| \\ &= |V^{-T}|\,|f|. \quad \square \end{aligned}$$

Theorem 2.3 bounds componentwise the absolute error in the computed solution. The following corollary bounds a norm-wise measure of the relative error, and will be used for comparison purposes in Sects. 3 and 4.

Corollary 2.4. *Under the hypotheses of Theorem* 2.3,

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leqq 5nu\,\frac{\|\,|V^{-1}|\,|b|\,\|_\infty}{\|x\|_\infty} + O(u^2),$$

$$\frac{\|a-\hat{a}\|_\infty}{\|a\|_\infty} \leqq 5nu\,\frac{\|\,|V^{-T}|\,|f|\,\|_\infty}{\|a\|_\infty} + O(u^2).$$

3. Implications of the Error Analysis

Now we discuss in detail the implications of Theorem 2.3, with particular reference to the primal algorithm (analogous comments apply to the dual). Throughout this section it is assumed that condition (2.1) is satisfied.

First, we show why Theorem 2.3 may, from one viewpoint, be regarded as very satisfactory. Consider the "ideal" situation where V^{-1} is known exactly, with $fl(V^{-1}) \equiv V^{-1}$ and $fl(b) \equiv b$, and suppose the solution to the primal system is computed as $\bar{x} = fl(V^{-1} \times b)$. Then, using the standard rounding error analysis for inner products [13, p. 36], the error in $\bar{x}$ can be bounded by

$$|x-\bar{x}| \leqq nu|V^{-1}||b| + O(u^2);$$

this is the same bound as in Theorem 2.3, except for the unimportant constant factor 5.

Consider now the case where, in Algorithm 1,

$$(-1)^i b_i \geqq 0, \qquad i=0,\ldots,n. \tag{3.1}$$

From Lemma 2.1, $|V^{-1}||b| = |V^{-1}b| = |x|$, whence Theorem 2.3 yields

$$|x-\hat{x}| \leqq 5nu|x| + O(u^2). \tag{3.2}$$

Thus, to first order in u, the relative error in computed nonzero components of x is bounded by a quantity *independent of the condition number* $k(V)$, and the bound is about as small as could possibly be expected! We have therefore identified a class, (3.1), of Vandermonde systems for which the Björck-Pereyra algorithm is guaranteed to provide high accuracy.

An interesting application of (3.2) is to the computation of V^{-1} via Algorithm 1 with $b=e_i$, $0 \leqq i \leqq n$, where the identity matrix $I_{n+1} = [e_0, e_1, \ldots, e_n]$. Denoting the computed V^{-1} by $\hat{X} = [\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_n]$ we have

$$|\hat{X} - V^{-1}| \leqq 5nu|V^{-1}| + O(u^2), \tag{3.3}$$

which shows that, contrary to what one might expect, V^{-1} can be computed with high relative accuracy, however ill-conditioned the matrix V. It should be mentioned that this approach to computing V^{-1} requires $O(n^3)$ operations, whereas Traub [20] gives a method requiring only $O(n^2)$ operations. However, [20] does not contain a rounding error analysis, and it can be shown that Traub's $O(n^2)$ algorithm must involve subtraction of like-signed numbers, suggesting that a result of the form (3.3) will not hold.

In Corollary 2.4 the key quantity in the bound for the primal algorithm is

$$\theta(V,b) = \frac{\||V^{-1}||b|\|_\infty}{\|x\|_\infty}.$$

Note that

$$1 \leqq \theta(V,b) \leqq \gamma(V),$$

where

$$\gamma(V) = \||V^{-1}||V|\|_\infty \leqq k_\infty(V). \tag{3.4}$$

To gain insight into $\theta(V,b)$ it is helpful to consider V and $|b|$ fixed, with different distributions of $\operatorname{sign}(b_i)$. In general, as these signs vary, the solution x varies in norm. For the distribution of alternating signs we have seen that $\|x\|_\infty$ attains the maximal value $\||V^{-1}||b|\|_\infty$, so that $\theta(V,b)=1$. However, there may be sign distributions for which $\theta(V,b) \gg 1$. For example, for $n=1$, with

$0 \leqq \alpha_0 < \alpha_1 \approx \alpha_0$ and $b = [1, \alpha_0]^T$, $x = [1, 0]^T$ and $\theta(V, b) = (\alpha_1 + \alpha_0)/(\alpha_1 - \alpha_0)$. Note that $\theta(V, b)$ can be large only when there is severe cancellation by subtraction in the product $x = V^{-1} b$.

Loosely, we can say that $\theta(V, b)$ is large or small if the system $Vx = b$ has respectively a "small solution" or a "large solution".

We stress that the results of Sect. 2 are valid for the case $0 \leqq \alpha_0 < \ldots < \alpha_n$ only. For some alternative orderings of one-signed points, our results hold with suitable modification (for example, $0 \geqq \alpha_0 > \ldots > \alpha_n$). However, for distributions containing both positive and negative points, suitable analogues of the results on which our analysis is based, Lemmas 2.1 and 2.2, do not hold. In particular, for a given $|b|$ there does not always exist a sign distribution $S = \text{diag}(\pm 1)$ such that no genuine subtraction of like-signed numbers occurs in Algorithm 1 for the right-hand side $S|b|$. An example is given by the case $n = 1$, $\alpha_0 < 0$, $\alpha_1 > 0$, $f_0 \neq 0$, $f_1 \neq 0$, in the dual algorithm; see Table 2.1.

Table 2.1. Dual

j	$f = c^{(0)}$	$c^{(1)} = a^{(1)}$	$a^{(0)}$
0	f_0	-	$f_0 - \alpha_0 \left(\frac{f_1 - f_0}{\alpha_1 - \alpha_0}\right)$
1	f_1	$\frac{f_1 - f_0}{\alpha_1 - \alpha_0}$	-

To avoid subtraction in forming $c_1^{(1)}$, we require that $f_1 f_0 \leqq 0$; but then, $(f_1 - f_0)/(\alpha_1 - \alpha_0)$ has opposite sign to f_0 and subtraction occurs in forming $a_0^{(0)}$. Interchanging α_0 and α_1 does not help. For a case such as this, the technique used to prove Theorem 2.3 is not applicable, because a suitable majorising Vandermonde system (corresponding to the right-hand side g in (2.10)) cannot be constructed.

Finally, we use Corollary 2.4 to compare the Björck-Pereyra primal algorithm with Gaussian elimination applied to (1.2), for the case where the points α_i satisfy (2.1). Let $\tilde{x}$ denote the computed solution from Gaussian elimination. To obtain an a priori bound for the error $\|x - \tilde{x}\|$, it is necessary to appeal to a backward error analysis. A particularly strong backward error result, given in [7], can be employed if we assume that pivoting is not used. For V is *totally positive* when the points α_i satisfy (2.1), that is, all the minors of V are nonnegative [10, p. 99]. Applying the results of [7] for Gaussian elimination without pivoting on a totally positive, nonsingular matrix, we have

$$(V + E)\tilde{x} = b, \qquad |E| \leqq \varepsilon |V|, \qquad \varepsilon = 4nu + O(u^2), \tag{3.5}$$

provided u is sufficiently small. (Here we have taken into account the errors which may be incurred in forming V from the α_i). A sharp bound for $\|x - \tilde{x}\|_\infty$ can be obtained by applying the perturbation theory of [17, Theorem 2.4] to (3.6); this yields

$$\frac{\|x - \tilde{x}\|_\infty}{\|x\|_\infty} \leqq 4nu \frac{\| |V^{-1}| |V| |x| \|_\infty}{\|x\|_\infty} + O(u^2). \tag{3.6}$$

Ignoring the unimportant constant factors 4 and 5, we see that the forward error bound (3.6) for Gaussian elimination is larger than the bound in Corollary 2.4 for Algorithm 1 by the factor $\Omega(V,b)$, where (see (3.4))

$$| \leqq \Omega(V,b)| = \frac{\| |V^{-1}| |V| |x| \|_\infty}{\| |V^{-1}| |b| \|_\infty} \leqq \gamma(V) \leqq k_\infty(V).$$

This comparison between bounds on the respective forward errors suggests, though it does not prove, that the computed solution from Algorithm 1 will usually be at least as accurate as that from Gaussian elimination, and may, in cases where $\Omega(V,b)$ is relatively large, be far more accurate.

We remark that the assumption of no pivoting is "fair" to Gaussian elimination in the sense that if any form of pivoting is used, it does not seem possible to obtain an a priori backward error result as strong as (3.5).

4. Condition Numbers

Theorem 2.3 gives a bound for the difference between the true and the computed solutions to the machine problem: the problem defined by the machine numbers $\hat{\alpha}_i, \hat{b}_i, \hat{f}_i$. The machine numbers may be subject to uncertainty, for example through rounding errors in converting to floating point (e.g., $\hat{\alpha}_i = fl(1/(1+i))$) or through rounding and truncation errors in computing the data (e.g., $\hat{f}_i = fl(\exp(\alpha_i))$), so the machine problem may differ from the problem whose solution is required. It is therefore desirable to understand the sensitivity of Vandermonde systems to perturbations in the data, that is, the conditioning of problems (1.2) and (1.3). By combining knowledge of the conditioning with the error analysis of Sect. 2, one can obtain bounds for the "true" error. A further, and equally important reason for investigating the conditioning is to determine whether Algorithms 1 and 2 introduce any more uncertainty into the computed solution than can be attributed to an inexact machine problem.

It is clear that the standard perturbation results for a linear system $Ax=b$, and the associated condition numbers, are applicable to Vandermonde systems; indeed much interesting work has been focussed in this direction (see, for example, [11, 12]). However, insofar as these results apply to general linear systems they do not take account of the structure of the Vandermonde matrix and so may be pessimistic. Note, for example, the rigid conditions to be imposed on a perturbation δV if $V+\delta V$ is to be a Vandermonde matrix.

In this section we derive condition numbers for the dual and the primal Vandermonde systems. No restrictions are placed on the points α_i. We use the same metrics in which to measure perturbations in the data and in the solution, and hence the same definition of condition number, as Skeel [17, Sect. 2]. Thus a perturbation $\delta\xi$ in the vector ξ of data (which consists here of the points α_i together with the right-hand side b or f) is measured by the smallest ε such that

$$|\delta\xi| \leqq \varepsilon |\xi|,$$

and the corresponding perturbation δx in the solution is measured norm-wise by

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty}.$$

The associated condition number is defined by

$$\lim_{\varepsilon \to 0} \sup_{\delta\xi: |\delta\xi| \leq \varepsilon|\xi|} \frac{\|\delta x\|_\infty}{\varepsilon \|x\|_\infty}.$$

We consider first perturbations in the right-hand side alone. In this case the special structure of V confers no advantage.

Lemma 4.1 [17]. Let $Ax=b$ *and* $A(x+\delta x)=b+\delta b$, *where* $|\delta b| \leq \varepsilon |b|$. *Then*

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \leq \varepsilon \frac{\| |A^{-1}| |b| \|_\infty}{\|x\|_\infty},$$

with equality for suitable choice of δb.

Proof. The bound follows from $|\delta x| = |A^{-1} \delta b| \leq \varepsilon |A^{-1}| |b|$. Equality is attained for $(\delta b)_j = \operatorname{sign}((A^{-1})_{kj}) \varepsilon |b_j|$, where $\| |A^{-1}| |b| \|_\infty = (|A^{-1}| |b|)_k$. □

We conclude that condition numbers for (1.2) and (1.3) with respect to relative perturbations in the right-hand sides are, respectively,

$$P_1 = \frac{\| |V^{-1}| |b| \|_\infty}{\|x\|_\infty},$$

$$D_1 = \frac{\| |V^{-T}| |f| \|_\infty}{\|a\|_\infty}.$$

Since these are the same quantities that appear in the bounds of Corollary 2.4, it follows that for the distribution (2.1), ignoring the second-order terms and the factors $5n$ in Corollary 2.4, *Algorithms* 1 *and* 2 *introduce no more uncertainty into the numerical solution than was already present if the machine right-hand side vector were subject to relative errors of the order of the unit roundoff.*

Next, we consider perturbations in the points α_i. For the primal system, $V = V(\alpha_0, \ldots, \alpha_n)$, let

$$Vx=b; \tag{4.1}$$

$$\bar{\alpha}_j = \alpha_j(1+\varepsilon_j), \quad |\varepsilon_j| \leq \varepsilon, \quad j=0, \ldots, n;$$

$$V(\bar{\alpha}_0, \ldots, \bar{\alpha}_n) \equiv V + \delta V = V + W + 0(\varepsilon^2);$$

$$(V+\delta V)(x+\delta x) = b. \tag{4.2}$$

It is easy to check that we can take

$$W=\begin{bmatrix} 0 & \cdots & 0 \\ \varepsilon_0\alpha_0 & & \varepsilon_n\alpha_n \\ 2\varepsilon_0\alpha_0^2 & & 2\varepsilon_n\alpha_n^2 \\ \vdots & & \vdots \\ n\varepsilon_0\alpha_0^n & \cdots & n\varepsilon_n\alpha_n^n \end{bmatrix}=H_n V\mathrm{diag}(\varepsilon_i),$$

where

$$H_n=\mathrm{diag}(0,1,2,\ldots,\mathrm{n}).$$

From (4.1) and (4.2),

$$\delta x=-(V^{-1}\delta Vx+V^{-1}\delta V\delta x),$$

which, provided that $\|V^{-1}\delta V\|<1$, implies

$$\frac{\|V^{-1}\delta Vx\|}{1+\|V^{-1}\delta V\|}\leqq\|\delta x\|\leqq\frac{\|V^{-1}\delta Vx\|}{1-\|V^{-1}\delta V\|}. \tag{4.3}$$

Now

$$\|V^{-1}\delta V\|=O(\varepsilon),$$

and

$$\|V^{-1}\delta Vx\|=\|V^{-1}Wx\|+O(\varepsilon^2),$$

so

$$\frac{\|V^{-1}\delta Vx\|}{1\pm\|V^{-1}\delta V\|}=\|V^{-1}Wx\|+O(\varepsilon^2).$$

Substituting in (4.3), we obtain

$$\frac{\|\delta x\|}{\|x\|}=\frac{\|V^{-1}Wx\|}{\|x\|}+O(\varepsilon^2).$$

Therefore, the required condition number is an attainable bound for $\|V^{-1}Wx\|_\infty/(\varepsilon\|x\|_\infty)$ that is independent of the individual ε_j. This bound can be computed with the aid of the following lemma.

Lemma 4.2. *Let $A\in\mathbb{R}^{n\times n}$, $z\in\mathbb{R}^n$ and $|\varepsilon_j|\leqq\varepsilon$, $j=0,\ldots,n$. Then*

$$\|A\,\mathrm{diag}(\varepsilon_j)\,z\|_\infty\leqq\varepsilon\,\||A|\,|z|\|_\infty,$$

with equality for suitable choice of ε_j, $j=0,\ldots,n$.

Proof. The bound follows from $|A\,\mathrm{diag}(\varepsilon_i)\,z|\leqq|A|\,\varepsilon I\,|z|=\varepsilon|A|\,|z|$. Equality is attained for $\varepsilon_j=\mathrm{sign}(a_{kj}z_j)\,\varepsilon$, where

$$\||A|\,|z|\|_\infty=(|A|\,|z|)_k.\quad\square$$

Applying the lemma to the expression

$$\|V^{-1}Wx\|_\infty=\|V^{-1}H_n V\mathrm{diag}(\varepsilon_i)\,x\|_\infty$$

we deduce the condition number for the primal system with respect to relative perturbations in the α_i,

$$P_2=\frac{\||V^{-1}H_n V|\,|x|\|_\infty}{\|x\|_\infty}.$$

In similar fashion one can show that the condition number for the dual problem with respect to relative perturbations in the α_i is given by an attainable upper bound for $\|V^{-T}W^T a\|_\infty/(\varepsilon\,\|a\|_\infty)$. Applying Lemma 4.2 once again, we find the condition number

$$D_2 = \frac{\| |V^{-T}| |V^T H_n a| \|_\infty}{\|a\|_\infty}.$$

The following bounds on the condition numbers are easily derived. Recall from (3.4) that $\gamma(V) = \| |V^{-1}| |V| \|_\infty \leqq \kappa_\infty(V)$. We have

$$1 \leqq P_1 \leqq \gamma(V), \qquad 0 \leqq P_2 \leqq n\,\gamma(V),$$
$$1 \leqq D_1 \leqq \gamma(V^T), \qquad 0 \leqq D_2 \leqq n\,\gamma(V^T).$$

The lower bounds are attainable for suitable choice of the Vandermonde system; this follows from Sect. 3 for P_1 and D_1, and for P_2 and D_2 is shown by the examples

$$P_2 = 0 \text{ for } x = e_0 = [1, 0, \ldots, 0]^T, \qquad \alpha_0 = 0 \ (P_1 = 1),$$
$$D_2 = 0 \text{ for } a = e_0 \qquad (D_1 = \|V^{-T}\|_\infty).$$

The upper bounds for P_1 and D_1 are attainable, but those for P_2 and D_2 are not.

Unfortunately, there does not appear to be any straightforward relationship between P_1 and P_2, or between D_1 and D_2, so it is difficult to compare theoretically the effect of perturbations in the points α_i with the effect of perturbations in the right-hand side.

5. Running Error Bounds

Practical experience with Algorithms 1 and 2 [5] shows that the computed solutions obtained on different problems can vary greatly in accuracy, and the analysis of Sect. 3 indicates that wide variation is possible even for fixed V with different right-hand sides. Clearly, estimates of the error in a computed solution are desirable.

Suppose the points α_i satisfy (2.1) and consider Algorithm 1. Theorem 2.3 is applicable and provides a bound for the error $|x - \hat{x}|$ in terms of $\phi = |V^{-1}|\,|b|$. From Lemma 2.1 it follows that ϕ may be evaluated efficiently as the modulus $\phi = |y|$ of the solution to the additional Vandermonde system

$$Vy = c; \qquad c_i = (-1)^i\,|b_i|, \quad i = 0, \ldots, n.$$

However, the main purpose of the a priori bonds of Theorem 2.3 is to give insight into the numerical behaviour of Algorithms 1 and 2, and one would not expect the bounds always to be accurate, because of the many inequalities used in their derivation.

We now present an alternative, a posteriori technique for bounding the error, which is applicable to any distribution of the points. The technique is essentially the same as that used in [1, 16] for bounding the error in evaluating a polynomial by Horner's scheme. The idea is to compute "running" bounds for the errors $\delta_j^{(k)}$, $\mu_j^{(k)}$ in the proof of Theorem 2.3, using versions of Eqs. (2.7) and (2.9) that contain the computed quantities $\hat{c}_j^{(k)}$, $\hat{a}_j^{(k)}$, rather than their exact (and hence unknown) counterparts $c_j^{(k)}$, $a_j^{(k)}$. The majorisation in the second part of the proof of Theorem 2.3 is not carried out, so, to first order, the running bounds are no larger than those provided by Theorem 2.3.

The running bounds are computed concurrently with the main solution process, since they make use of the intermediate computed quantities $\hat{c}_j^{(k)}$, $\hat{a}_j^{(k)}$ or $\hat{d}_j^{(k)}$, $\hat{x}_j^{(k)}$.

The bounds are defined by the following recurrences, whose straightforward derivation (cf. [1, 16]) is omitted.

For Algorithm 1

Stage I:

$$\Delta_j^{(k+1)} = \Delta_j^{(k)} + |\alpha_k|\,\Delta_{j-1}^{(k)} + |\hat{d}_j^{(k+1)}| + |\alpha_k\,\hat{d}_{j-1}^{(k)}| \tag{5.1}$$

Stage II:

$$M_j^{(k+\frac{1}{2})} = \frac{M_j^{(k+1)}}{|\alpha_j - \alpha_{j-k-1}|} + 2\,|\hat{x}_j^{(k+\frac{1}{2})}| \tag{5.2a}$$

$$M_j^{(k)} = M_j^{(k+\frac{1}{2})} + M_{j+1}^{(k+\frac{1}{2})} + |\hat{x}_j^{(k)}| \tag{5.2b}$$

For Algorithm 2

Stage I:

$$\Delta_j^{(k+1)} = \frac{\Delta_j^{(k)} + \Delta_{j-1}^{(k)}}{|\alpha_j - \alpha_{j-k-1}|} + 3\,|\hat{c}_j^{(k+1)}| \tag{5.3}$$

Stage II:

$$M_j^{(k)} = M_j^{(k+1)} + |\alpha_k|\,M_{j+1}^{(k+1)} + |\hat{a}_j^{(k)}| + |\alpha_k\,\hat{a}_{j+1}^{(k+1)}|. \tag{5.4}$$

The index ranges are understood to be those defined in the respective algorithms, and in both cases initial values are defined by

$$\left.\begin{array}{l} \Delta_j^{(0)} = 0 \\ M_j^{(n)} = \Delta_j^{(n)} \end{array}\right\} j = 0, \ldots, n.$$

The final error bounds are, for $z = x$ (Algorithm 1) or $z = a$ (Algorithm 2).

$$|z - \hat{z}| \leqq u\,|M^{(0)}| + O(u^2). \tag{5.5}$$

The running bounds are slightly more expensive to evaluate than the bounds based on Theorem 2.3, due to the terms involving $\hat{c}_j^{(k)}$, $\hat{a}_j^{(k)}$, $\hat{d}_j^{(k)}\,\hat{x}_j^{(k)}$ in (5.1)-(5.4).

Some numerical experiments to compare the running bounds with the a priori bounds of Theorem 2.3 are reported in the next section.

6. Numerical Experiments

Several numerical experiments have been carried out using a Fortran 77 program on a CDC computer with unit roundoff $u = 2^{-48} \approx 3.55 \times 10^{-15}$. The following questions were investigated.

1) How much smaller are the running error bounds than the a priori bounds?

2) Are the running error bounds realistic *estimates* of the error?

3) Is the ordering in (2.1) demonstrably superior to other orderings?

4) Are the answers to questions 2) and 3) valid for general points α_i, or just for nonnegative points?

Numerical results for the first question may be summarised as follows. We ran a wide variety of dual and primal test problems for points satisfying (2.1), with $5 \leqq n \leqq 30$. In every case the a priori bound of Theorem 2.3 was larger in every component than the running bound (5.5) (as expected - see Sect. 5), usually by approximately the same factor for each component. The largest ratio of components was 1.76×10^4, with ratios of orders 10^2 and 10^3 occurring frequently.

To investigate the remaining questions the following approach was adopted. Each test problem was formed and solved, and the running error bound computed, in single-precision arithmetic. The single-precision numbers defining the machine problem were converted to double-precision and the problem then solved entirely in double-precision arithmetic. The difference between the computed single- and double-precision solutions was used to form an approximation e to the relative error,

$$e_i = \left| \frac{\hat{x}_i(dp) - \hat{x}_i(sp)}{\hat{x}_i(dp)} \right|, \quad i = 0, \ldots, n;$$

for all problems of interest (that is, those with $\hat{x}(sp)$ having some correct digits) e can be expected to give the relative error correct to single precision.

Six test problems are reported in detail. The first two are taken from [5], with the points, which are in decreasing order in [5], rearranged in increasing order.

$$\text{Primal:} \quad \alpha_i = \frac{1}{n-i+3}, \qquad b_i = \frac{1}{2^i}; \tag{6.1}$$

$$\text{Dual:} \quad \alpha_i = \frac{1}{n-i+2}, \qquad f_i = T_n(\alpha_i), \tag{6.2}$$

where $T_n(x)$ is the Chebyshev polynomial of degree n. Problem (6.2) is to recover the coefficients of a Chebyshev polynomial from function values at the points α_i. The remaining problems are

$$\text{Primal:} \quad \alpha_{n-i} = \tfrac{1}{2}\left(1 + \cos\left(\frac{(i+\frac{1}{2})\pi}{n+1}\right)\right), \qquad b = e_n; \tag{6.3}$$

$$\text{Dual:} \quad \alpha_i = \frac{i}{n}, \qquad f_i = \frac{1}{1+25\alpha_i^2}; \tag{6.4}$$

$$\text{Primal:} \quad \alpha_{n-i} = \cos\left(\frac{(i+\frac{1}{2})\pi}{n+1}\right), \qquad b = e_n; \tag{6.5}$$

$$\text{Dual:} \quad \alpha_i = -1 + \frac{2i}{n}, \qquad b_i = \frac{1}{2^i}. \tag{6.6}$$

Table 6.1. Problem 6.1

n	$\lvert x_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i max	e_i/u max	M_i/e_i max	e_i/u max
5	4.6E1	3.6E3	1.1E2	2.6	1.8E2	6.0
10	6.3E2	1.6E9	2.9E2	1.0E1	1.8E4	3.8E4
15	7.0E3	8.6E15	9.5E2	1.8E1	3.6E7	1.2E10
20	7.0E4	2.5E23	3.5E4	1.7E1	1.5E14	9.7E14
25	6.6E5	2.5E31	7.0E4	2.1E1	5.2E11	1.2E26
30	5.9E6	6.9E39	3.4E4	2.4E1	4.1E24	1.6E29

Table 6.2. Problem 6.2

n	$\lvert a_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i max	e_i/u max	M_i/e_i max	e_i/u max
5	6.2E-13	2.0E1	1.0E3	1.1E12	5.2E1	1.9E13
10	1.4E-8	5.1E2	2.5E1	1.7E13	3.4E1	1.8E15
15	6.9E-7	4.8E7	3.3E1	8.9E11	1.1E3	2.9E14
20	5.2E-2	4.4E16	9.2E1	1.2E12	1.1E6	7.0E13
25	5.6E-3	4.6E24	7.6	2.8E15	1.2E6	1.1E18
30	4.5	2.6E35	2.3E1	2.3E13	2.8E6	4.6E15

The points in (6.3) and (6.5) are the Chebyshev interpolation points [6, p. 243], for the intervals $[0, 1]$ and $[-1, 1]$ respectively, and in both cases the problem is to compute the last column of V^{-1}.

In each test, the problem was solved twice, first with the points in the given, increasing order, and then with a random ordering (depending on n, but the same for each problem). The results are given in Tables 6.1–6.6, in which M_i denotes the i'th component of the running error bound. Note that, ideally, $M_i/e_i \equiv 1$ (exact bound) and $e_i/u \equiv 1$ (minimal relative error).

We make the following comments and observations on Tables 6.1–6.6.

(a) The tendency of Vandermonde matrices to be extremely ill-conditioned (in the sense of a large standard condition number $\kappa(V)$) is evident from the large-normed solutions, which are produced from right-hand sides with elements of order 1.

(b) For the increasing order, Tables (6.1) and (6.3) illustrate well the phenomenon of highly accurate computed solutions. For problem (6.3) the high accuracy is predicted by the a priori bound of Theorem 2.3 (see (3.1), (3.2)). However, in problem (6.1) the a priori bound was between 10 and 20 times larger than the running bound, which itself was a moderately pessimistic estimate of the error, as can be seen from Table 6.1.

(c) The quality of the running bounds as estimates of the error is somewhat variable. In problems (6.2), (6.3), (6.4) and (6.5), for the increasing ordering, the running bound provides a sharp estimate, but in problems (6.1) and (6.6) it is

Table 6.3. Problem 6.3

n	$\lvert x_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i	e_i/u	M_i/e_i	e_i/u
			max	max	max	max
n	4.4E1	1.6E2	1.3E1	4.3	4.0E1	5.8
10	1.4E4	9.5E4	6.3	8.5	2.3E4	2.0E1
15	6.6E6	6.7E7	1.1E1	1.1E1	7.3E5	4.7E2
20	3.9E9	5.2E10	8.5	1.8E1	1.6E7	1.2E4
25	2.6E12	4.3E13	6.1	2.3E1	2.6E12	1.1E4
30	1.9E15	3.7E16	1.2E1	2.2E1	1.5E11	4.9E4

Table 6.4. Problem 6.4

n	$\lvert a_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i	e_i/u	M_i/e_i	e_i/u
			max	max	max	max
5	1.0	1.6E1	7.5	3.2E1	2.7E1	7.8E1
10	9.2E-1	2.8E3	1.1E3	4.8E1	1.3E3	2.1E3
15	2.0E-1	1.3E6	4.3	1.9E4	4.8E3	5.2E5
20	2.5E-2	3.6E8	5.2	1.0E6	2.4E5	3.6E7
25	2.8E-3	7.4E10	1.4	3.0E9	3.0E8	2.3E10
30	2.8E-3	1.1E13	1.1E1	4.4E10	3.6E9	3.3E12

Table 6.5. Problem 6.5

n	$\lvert x_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i	e_i/u	M_i/e_i	e_i/u
			max	max	max	max
5	1.4	5.2	6.3	5.8	4.4E1	5.8
10	1.3E1	9.3E1	6.1	9.7	4.6E3	3.9E1
15	2.0E3	2.0E3	6.5	1.1E1	3.3E6	1.3E3
20	3.7E3	5.0E4	8.4	1.6E1	2.7E6	1.4E4
25	7.8E4	1.3E6	8.5	2.5E1	1.7E11	6.3E3
30	1.7E6	3.5E7	9.2	2.2E1	9.1E11	2.0E6

Table 6.6. Problem 6.6

n	$\lvert a_i\rvert$		Increasing order		Random order	
	min	max	M_i/e_i	e_i/u	M_i/e_i	e_i/u
			max	max	max	max
5	2.5E-2	3.1	1.5E2	7.4	9.7E2	2.7E2
10	2.6E-3	2.2	1.5E3	5.1E3	1.1E4	1.7E5
15	3.1E-4	1.7	4.5E7	1.4E3	1.6E7	8.4E7
20	3.9E-5	1.5	1.2E6	3.0E7	1.0E6	6.2E11
25	5.1E-6	1.4	4.6E6	7.7E10	5.6E10	6.5E14
30	1.2E-6	1.2	8.6E7	3.5E13	4.8E13	2.1E17

quite pessimistic. For the random ordering the running bound is completely unreliable as an error estimate.

(d) The maximum componentwise relative error is, in every case, larger for the random ordering of points than for the increasing ordering. The ordering clearly can have a profound effect on the relative error: in problem (6.1), for $n = 20, 25, 30$, the increasing ordering yields approximately 14 correct significant digits, the random ordering none.

7. Conclusions

Our investigation has provided several new insights into the numerical behaviour of the Björck-Pereyra algorithms for solving Vandermonde systems. The rounding error analysis in Sect. 2 gives theoretical support to the observation of Björck and Pereyra that their algorithms sometimes produce surprisingly accurate solutions to ill-conditioned systems. The analysis requires that the points α_i be nonnegative and be arranged in increasing order. We identified certain sign properties that are essential to the analysis (Lemmas 2.1 and 2.2), but which do not hold for general orderings of the points, or for distributions containing both positive and negative points.

The numerical results of Sect. 6 demonstrate clearly that the ordering of the points can have a profound influence on the accuracy of the computed solution (see, in particular, Table 6.1). The combined theoretical and numerical evidence presented here leads us to conclude that the increasing order is a sound choice, though we suspect that the best ordering (that is one that minimises the error in the computed solution, or an a priori bound on the error) may be problem-dependent. We have been unable to discern numerically any major difference in the behaviour of the Björck-Pereyra algorithms for the cases of nonnegative points and general points α_i.

In summary, we make the following recommendations for use of the Björck-Pereyra algorithms.

(1) Order the points α_i in increasing order.

(2) If error estimates are required, compute concurrently with the main solution process the running error bounds of Sect. 5. These bounds can overestimate the error by many orders of magnitude, but they may be useful for verifying that correct digits have been obtained.

(3) If the machine data α_i, and b or f, is contaminated by small relative errors, use the condition numbers of Sect. 4 to estimate the corresponding additional uncertainty in the computed solution.

Acknowledgements. I am pleased to thank Ian Gladwell for stimulating discussions on this work and for his comments on the manuscript. I also thank Len Freeman for his comments on the manuscript and for pointing out reference [1]. The helpful suggestions of the referee are much appreciated.

References

1. Adams, D.A.: A stopping criterion for polynomial root finding. Commun. ACM **10**, 655–658 (1967)

2. Almacany, M., Dunham, C.B., Williams, J.: Discrete Chebyshev approximation by interpolating rationals. IMA J. Numer. Anal. **4**, 467-477 (1984)
3. Ballester, C., Pereyra, V.: On the construction of discrete approximations to linear differential expressions. Math. Comput. **21**, 297-302 (1967)
4. Björck, Å., Elfving, T.: Algorithms for confluent Vandermonde systems. Numer. Math. **21**, 130-137 (1973)
5. Björck, Å., Pereyra, V.: Solution of Vandermonde systems of equations. Math. Comput. **24**, 893-903 (1970)
6. Conte, S.D., de Boor, C.: Elementary numerical analysis (Third edition). New York-Tokyo: McGraw-Hill 1980
7. de Boor, C., Pinkus, A.: Backward error analysis for totally positive linear systems. Numer. Math. **27**, 485-490 (1977)
8. Dunham, C.B.: Choice of basis for Chebyshev approximation. ACM Trans. Math. Software **8**, 21-25 (1982)
9. Freeman, T.L.: Solution of Vandermonde systems of equations: an alternative view. Numerical Analysis Report No. 45, University of Manchester, England, 1980
10. Gantmacher, F.R.: The Theory of Matrices, vol. Two. New York: Chelsea 1959
11. Gautschi, W.: On inverses of Vandermonde and confluent Vandermonde matrices. Numer. Math. **4**, 117-123 (1962)
12. Gautschi, W.: Optimally conditioned Vandermonde matrices. Numer. Math. **24**, 1-12 (1975)
13. Golub, G.H., Van Loan, C.F.: Matrix Computations. Baltimore, MD: Johns Hopkins University Press 1983
14. Gustafson, S.-Å.: Control and estimation of computational errors in the evaluation of interpolation formulae and quadrature rules. Math. Comput. **24**, 847-854 (1970)
15. Lyness, J.N., Moler, C.B.: Van der Monde systems and numerical differentiation. Numer. Math. **8**, 458-464 (1966)
16. Peters, G., Wilkinson, J.H.: Practical problems arising in the solution of polynomial equations. J. Inst. Math. Appl. **8**, 16-35 (1971)
17. Skeel, R.D.: Scaling for numerical stability in Gaussian elimination. J. Assoc. Comput. Mach. **26**, 494-526 (1979)
18. Tang, W.P., Golub, G.H.: The block decomposition of a Vandermonde matrix and its applications. BIT **21**, 505-517 (1981)
19. Trapp, G.E., Squire, W.: Solving nonlinear Vandermonde systems. Comput. J. **18**, 373-374 (1975)
20. Traub, J.F.: Associated polynomials and uniform methods for the solution of linear problems. SIAM Rev. **8**, 277-301 (1966)

Received March 3, 1986 / November 17, 1986