A Fast Parallel Björck-Pereyra-type Algorithm for Solving Cauchy Linear Equations *

Tibor Boros
ArrayComm, Inc.
3141 *Zanker Road*
San Jose, CA 95134, *USA*

Thomas Kailath
Information Systems Laboratory
Department of Electrical Engineering
Stanford University, Stanford CA 94305-4055, *USA*

and

Vadim Olshevsky†
Department of Mathematics and Statistics
Georgia State University
Atlanta, GA 30303, *USA*

To Hans Schneider on the occasion of his 70th birthday

Abstract

We propose a new fast $O(n^2)$ parallel algorithm for solving Cauchy systems of linear equations. We perform an a priori rounding error analysis and obtain componentwise bounds for the forward, backward and residual errors. These bounds indicate that for the class of totally positive Cauchy matrices the new algorithm is forward and backward stable, producing a remarkably high relative accuracy. In particular, Hilbert linear systems, often considered to be too ill-conditioned to be attacked, can be rapidly solved with high precision.

The results indicate a close resemblance between the numerical properties of Cauchy matrices and the much-studied Vandermonde matrices. In fact, our proposed Cauchy solver is an analog of the well-known Björck-Pereyra algorithm for Vandermonde systems. As a by-product we obtain favorably backward error bounds for the original Björck-Pereyra algorithm.

Several computed examples illustrate a connection of high relative accuracy to the concepts of effective well-conditioning and total positivity.

Key words: Cauchy matrix, Hilbert matrix, Vandermonde matrix, Fast algorithms, Björck-Pereyra algorithm, Forward error analysis, Backward error analysis, Ordering of interpolation points, Total positivity, Effective well-conditioning, Skeel condition number, Chan-Foulser condition number.

AMS subject classification: 65F05, 65L20, 15A09, 15A23

*This work was supported in part by NSF contract CCR 9732355 and ARO contract DAAH04-96-1-0176.
†*contact author*: web: www.cs.gsu.edu/~matvro email: volshevsky@cs.gsu.edu

1 Introduction

1.1. Cauchy matrices. Cauchy matrices of the form

$$C(x_{1:n},\ y_{1:n}) = \begin{bmatrix} \frac{1}{x_1-y_1} & \cdots & \frac{1}{x_1-y_n} \\ \vdots & \ddots & \vdots \\ \frac{1}{x_n-y_1} & \cdots & \frac{1}{x_n-y_n} \end{bmatrix} \tag{1.1}$$

are encountered in many important applications, including particle simulation [Ro85], the computation of conformal mappings [T86], multiple evaluation of the Riemann zeta function [OS86], numerical solution of singular integral equations [R90a], construction of soliton-like solutions of some nonlinear equation of KdV type [ST90], the pole placement problem [HLM95], various rational interpolation problems [BGR90], [GO94], [KO97], [OP98], in preconditioning [KO96]. in the decoding of Reed-Solomon, Goppa and algebraic-geometric codes [RS94], [MS77], [OS99]. In [BKO94] (see also [GKO95], [KO95] for a more general result) we showed that Vandermonde and Chebyshev-Vandermonde matrices can be efficiently transformed to Cauchy matrices by using Fast Fourier, Cosine or Sine transforms. Therefore computational results obtained for Cauchy matrices can be used to solve the corresponding problems for Vandermonde and Chebyshev-Vandermonde matrices.

Matrices of the form (1.1) are classical. The problem of solving the associated linear system with the special right-hand side $\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ was considered in the year 1837 by Binet [B37], who was motivated by a geometric application. Four years later Cauchy [C41] solved the problem for a general right-hand side, arriving at his well-known formula

$$\det C(x_{1:n},\ y_{1:n}) = \frac{\prod_{j>k}(x_j - x_k)\ \prod_{j<k}(y_j - y_k)}{\prod_{j,k}(x_j - y_k)} \qquad 1 \le j,k \le n. \tag{1.2}$$

This formula provides an explicit expression, which can be found in several sources, see, *e.g.*, [M30],

$$a_j = \frac{f(y_j)}{g'(y_j)}\left(\sum_{i=1}^{n} \frac{f_i}{x_j - y_i}\right)\frac{g(x_i)}{f'(x_i)} \qquad \text{with} \qquad f(x) = \prod_{i=1}^{n}(x - x_i), \quad g(x) = \prod_{i=1}^{n}(x - y_i) \tag{1.3}$$

for the entries of the solution $a = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^T$ for a Cauchy linear system with right-hand side $f = \begin{bmatrix} f_1 & f_2 & \cdots & f_n \end{bmatrix}^T$. Note that for the special right-hand side $\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$, formula (1.3) was given in [B37]. At Cauchy's time determinants were the main tool to express the solution of a linear system, so an immediate reformulation of (1.3) yields the formula

$$C(x_{1:n},\ y_{1:n})^{-1} = \left[\frac{\prod_{k=1}^{n}(x_k - y_i)}{\prod_{\substack{k=1 \\ k\neq i}}^{n}(y_k - y_i)} \cdot \frac{1}{x_j - y_i} \cdot \frac{\prod_{k=1}^{n}(x_j - y_k)}{\prod_{\substack{k=1 \\ k\neq i}}^{n}(x_j - x_k)} \right]_{1\le i,j\le n}, \tag{1.4}$$

which could also be attributed to Cauchy, although nowadays (1.4) is usually associated with the names of Schechter [S59] and Gastinel [Gas60]. The expression (1.4) allows one to solve the linear system $C(x_{1:n},\ y_{1:n}) \cdot a = f$ by forming $a = C(x_{1:n},\ y_{1:n})^{-1} \cdot f$ in only $9n^2$ float point operations (flops), which is less than $O(n^3)$ flops of standard structure-ignoring methods.

1.2. An analogy with Vandermonde matrices. Many algebraic properties of Cauchy matrices are similar to those of Vandermonde matrices,

$$V(x_{1:n}) = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix}. \tag{1.5}$$

For example, there is a classical closed form expression for the determinant of $V(x_{1:n})$. Furthermore, analogous to (1.4), an explicit inversion formula for Vandermonde matrices, see, e.g., [K32], can be used

to compute the entries of $V^{-1}(x_{1:n})$ in only $6n^2$ arithmetic operations, leading to the well-known Parker-Forney-Traub algorithm [P64], [F66], [T66]. There are several other similar algebraic properties, in particular, both $C(x_{1:n}, y_{1:n})$ and $V(x_{1:n})$ have displacement structure, see, e.g., [KS95], [GKO95], [O97] and the references therein.

1.3. The Björck-Pereyra algorithm. As we shall see, along with many interesting algebraic properties, $C(x_{1:n}, y_{1:n})$ and $V(x_{1:n})$ have many often favorable *numerical properties*, which, however, have been much more studied and understood for Vandermonde matrices [BP70], [TG81], [CF88], [Hig87], [Hig88], [Hig90], [RO91], [CR92], [V93], [GK93], as compared to the analysis of numerical issues related to Cauchy matrices [GK90], [GK93].

In particular, most of the above listed papers were devoted to the analysis and extensions of the fast $O(n^2)$ Björck-Pereyra algorithm for Vandermonde systems [BP70], [GVL89]. This algorithm is now a well-known example where the exploitation of the structure allows one not only to speed-up computations, but also to achieve, for special right-hand sides, more accuracy in the computed solution than when using standard (non-fast) numerically stable methods. The first indication of this interesting phenomena can be found in [BP70], where it was observed that *"some problems, connected with Vandermonde systems, which traditionally have been considered to be too ill-conditioned to be attacked, actually can be solved with good precision"*. This fact attracted much attention and motivated a number of papers appearing in the last decade. Specifically, many efforts were devoted to the extension of the Björck-Pereyra algorithm to more general classes of matrices. Thus, W.Tang and G.Golub [TG81] devised a closely related algorithm for block Vandermonde matrices, N.J.Higham extended in [Hig88] this algorithm to the so-called Vandermonde-like matrices involving orthogonal polynomials[1], and L.Reichel and G.Opfer [RO91] specified a progressive (i.e., allowing updating) Björck-Pereyra-type algorithm for Chebyshev-Vandermonde matrices.

Another challenging problem was to give a theoretical support for the favorable numerical properties of the Björck-Pereyra algorithm by performing an *a priori* rounding error analysis; a class of Vandermonde systems with a nice error bound was identified in [Hig87]. Here we may recall Wilkinson's [W71] advocation not to give *" ... much importance to the precise error bounds obtained by a priori error analysis,"* even they are often impractically large, they can *"expose the potential instabilities, if any, of an algorithm, so that hopefully from the insight thus obtained one might be led to improved algorithms."* In contrast to many such impractical (though still useful) bounds, the one of [Hig87] is surprisingly favorable, predicting that the Björck-Pereyra algorithm can produce all 7 possible-in-single-precision digits, even in situations where Gaussian elimination with complete pivoting will fail to produce as much as one correct digit. This occurrence motivated a further extension of the forward error bounds of [Hig87] to three-term Vandermonde matrices, see, *e.g.*, [Hig88] and [Hig90].

1.4. Main results and contents. However, all the attempts at generalization were so far limited to Vandermonde-related structures. In the present paper we demonstrate that all the above important results can be carried over to certain other classes of structured matrices, in particular to Cauchy matrices.

After a brief review of known facts in Sec. 2, we give a detailed description of the main results. An analogue of the Björck-Pereyra algorithm for Cauchy matrices is presented in Sec.3. Then in Sec. 4 we perform not only a forward, but also a backward error analysis for the new algorithm. In Sec. 5 we identify an important class of *totally positive* Cauchy systems for which our algorithm is shown to be forward and backward stable, guaranteeing a remarkably high accuracy. In section 6 we prove that the new algorithm is *effectively stable* (see the main text below for the definition of this notion), and illustrate this point by several computed examples. Some remarks and practical recommendations are offered in the concluding section.

The results of this paper and of [BKO94] were available since 1994 as ISL reports at Stanford University, and they were reported at several conferences. They have influenced a recent interest to connections between accuracy and total positivity, cf., e.g., with [CR96], [CR97], [D97], [DG97] and [MP98].

[1] We call them three-term Vandermonde matrices, because we use the postfix *"like"* in a different meaning, which it has in the context of displacement structure theory.

2 Related facts and main results

2.1 Vandermonde matrices

The Björck-Pereyra algorithm. Björck and Pereyra derived in [BP70] (see also [GVL89]) a fast algorithm (the BP algorithm) for solving Vandermonde linear systems, and observed that it corresponds to the decomposition of the inverse of a Vandermonde matrix $V(x_{1:n})$ into the product

$$V^{-1}(x_{1:n}) = U_1 \ \ldots \ U_{n-1} \ L_{n-1} \ \ldots \ L_1 \ , \tag{2.6}$$

of diagonal and bidiagonal factors of the form

$$U_k = \begin{bmatrix} 1 & -\gamma_1^{(k)} & & \\ & 1 & \ddots & \\ & & \ddots & -\gamma_n^{(k)} \\ & & & 1 \end{bmatrix} \begin{bmatrix} \delta_0^{(k)} & & & \\ & \delta_1^{(k)} & & \\ & & \ddots & \\ & & & \delta_n^{(k)} \end{bmatrix}, \qquad L_k = \begin{bmatrix} 1 & & & \\ -\alpha_1^{(k)} & 1 & & \\ & \ddots & \ddots & \\ & & -\alpha_n^{(k)} & 1 \end{bmatrix},,$$

with some $\gamma_i^{(k)}, \delta_i^{(k)}, \alpha_i^{(k)}$. The BP algorithm solves the corresponding linear system by multiplying the representation in (2.6) by the right-hand side, achieving a favorable efficiency of only $\frac{5}{2}n^2$ flops.

Leja Ordering. It is important to realize that the BP algorithm is not invariant to permutations of the points defining the Vandermonde matrix; different configurations of $\{x_{1:n}\}$ yield different decompositions for $V(x_{1:n})^{-1}$, though all of the form (2.6). Therefore the way roundoff errors propagate depends on the ordering of the points. It was empirically observed in [Hig90] that the accuracy of the BP algorithm is often improved by the ordering :

$$\begin{aligned} |x_1| &= \max_{1\le j\le n} |x_j|, \\ \textstyle\prod_{i=1}^{k-1} |x_k - x_i| &= \max_{k\le j\le n} \textstyle\prod_{i=1}^{k-1} |x_j - x_i| \qquad \text{for} \qquad 2 \le k \le n \ , \end{aligned} \tag{2.7}$$

The ordering (2.7) mimics the row interchange that would be produced by applying Gaussian elimination with partial pivoting. We note that (2.7) was called the *Leja ordering* in [R90b], because of its relation to to the Leja's work on interpolation. See, *e.g.*, [W35], p.170-173, where they were called the *Fekete points*.

Monotonic ordering. Forward stability. Although Leja ordering often improves the numerical performance of the BP algorithm, nonetheless there exists a specific class of Vandermonde systems for which it is not optimal. Thus, it was shown in [Hig87] (see also [Hig96]) that if all the points $\{x_{1:n}\}$ are positive and monotonically ordered[2]:

$$0 \le x_1 \le x_2 \le x_3 \le \ldots \le x_n \ , \tag{2.8}$$

then the error in the solution $\widehat{a}$ of a Vandermonde system $V(x_{1:n})a = f$ computed by the BP algorithm can be nicely bounded:

$$|a - \widehat{a}| \le 5nu|V(x_{1:n})^{-1}||f| + \mathcal{O}(u^2), \tag{2.9}$$

where n denotes the size of the exact solution a, u stands for the machine precision, and the operations of comparison, and of taking the absolute value of a vector or a matrix, are understood in a componentwise sense.

Furthermore, it was observed in [Hig87] that if the components of the right-hand side $f = \begin{bmatrix} f_i \end{bmatrix}_{1\le i\le n}$ are sign-interchanging, i.e., $(-1)^i \cdot f_i \ge 0$, then (2.9) reduces to the following pleasing bound,

$$|a - \widehat{a}| \le 5nu|a| + \mathcal{O}(u^2), \tag{2.10}$$

which does not depend on the (exponentially growing with the size [GI88], [Ty94]) condition number of the Vandermonde matrix. Recalling that *a priori* error analysis is usually very pessimistic (compared to

[2] The BP algorithm was derived in [BP70] using a polynomial language, and its first part, multiplying the factors $L_{n-1}...L_1$ by a right-hand side f, is in fact the "good old" Newton divided differences table. We would like to thank W.Kahan, who drew our attention to the note [KF63], where it was noted that monotonic ordering of the nodes is crucial in order for a rounding error analysis of the divided difference table to result in a favorably small forward bound.

actual practical results), the bound in (2.10), which is about as small as could possibly be expected, seems to be a striking exception to the general rule. Indeed, it identifies a class of Vandermonde systems for which the BP algorithm is guaranteed to provide a very high accuracy. Figure 1 illustrates this point, showing that for positive monotonically ordered points and a sign-interchanging right-hand side, the BP algorithm does produce about 7 correct digits from about the 7 possible in single precision. Notice also the failure of numerically stable Gaussian elimination with complete pivoting, which is not surprising given the extreme ill-conditioning of the coefficient matrix, as shown in Table 2.

Monotonic ordering. Backward stability. The paper [Hig87] contains only a forward error bound for the BP algorithm; an attempt to also obtain results on backward stability was made in [Hig90]. The backward bounds of [Hig90], corollary 4.2, were obtained under certain additional (beyond (2.8)) assumptions, with the remark that, generally speaking, those simplifying assumptions did not hold. Specifically, it was observed in [Hig87] that each of the $n-1$ steps of the BP algorithm (corresponding to the first $n-1$ factors $L_{n-1}...L_1$, i.e. to the divided differences table) introduces a small backward error.

Table 2. Conditioning of Vandermonde matrix with $x_i = i^2/n^2$.

n	5	10	15	20	25	30	35	40
$\kappa_\infty(V(x_{1:n}))$	1e+03	3e+07	2e+12	9e+16	3e+18	7e+18	8e+18	8e+19

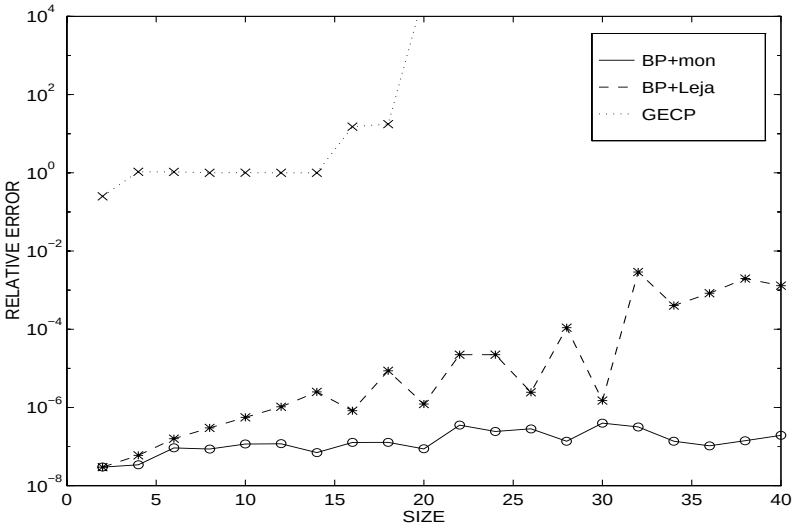


Figure 1: *Vandermonde linear system with* $x_i = i^2/n^2$, *and* $f_i = (-1)^i$, $(i = 1, \ldots, n)$. *The graphs display the relative error* $\|a - \widehat{a}\|_\infty / \|a\|_\infty$ *as a function of* n, *for the following three algorithms :* **BP+mon:** *BP algorithm with monotonically ordered nodes,* **BP+Leja:** *BP algorithm with Leja ordering,* **GECP:** *Gaussian elimination with complete pivoting.*

It was further noted that if the same were true for the second $n-1$ steps, corresponding to $U_{n-1}...U_1$, then the overall procedure would be backward stable. However, since this assumption is not formulated in terms of the input data, and hence it is actually not a form of an *a priori* error analysis. Therefore, no particular class of problems can be associated with the bounds formulated in [Hig90].

However, one can show that no additional assumptions are necessary, and that when (2.8) holds, the BP algorithm is backward stable, computing the solution $\widehat{a}$ of $V(x_{1:n})a = f$, with the following favorable bounds for the corresponding backward and residual errors :

$$|V(x_{1:n}) - \widehat{V}| \leq c_n u V(x_{1:n}) + \mathcal{O}(u^2), \qquad \text{where} \qquad \widehat{V}\widehat{a} = f, \tag{2.11}$$

$$|f - V(x_{1:n})\widehat{a}| \leq c_n u V(x_{1:n})|\widehat{a}| + \mathcal{O}(u^2), \tag{2.12}$$

with $c_n = 12n^2$. These new bounds can be deduced with exactly the same arguments used to obtain their counterparts for Cauchy matrices in Theorem 4.3 in the main text below.

2.2 Main results

The Björck-Pereyra-type algorithm for Cauchy matrices. We design a new algorithm (referred below to as the BKO algorithm) for solving Cauchy linear systems, which is based on the decomposition

$$C^{-1}(x_{1:n},\, y_{1:n}) = U_1 \,\ldots\, U_{n-1}\; D\; L_{n-1} \,\ldots\, L_1\;, \tag{2.13}$$

into a product of of diagonal and bidiagonal factors (whose entries will be specified in Section 3) of the form

$$U_k = \begin{bmatrix} 1 & -\gamma_1^{(k)} & & \\ & 1 & \ddots & \\ & & \ddots & -\gamma_n^{(k)} \\ & & & 1 \end{bmatrix} \begin{bmatrix} \delta_0^{(k)} & & & \\ & \delta_1^{(k)} & & \\ & & \ddots & \\ & & & \delta_n^{(k)} \end{bmatrix}, \quad D = \begin{bmatrix} \omega_0 & & & \\ & \omega_1 & & \\ & & \ddots & \\ & & & \omega_n \end{bmatrix}, \quad L_k = \begin{bmatrix} \beta_0^{(k)} & & & \\ & \beta_1^{(k)} & & \\ & & \ddots & \\ & & & \beta_n^{(k)} \end{bmatrix} \begin{bmatrix} 1 & & & \\ -\alpha_1^{(k)} & 1 & & \\ & \ddots & \ddots & \\ & & -\alpha_n^{(k)} & 1 \end{bmatrix}. \tag{2.14}$$

The corresponding linear system is solved by multiplying the expression (2.13) by a right-hand side, which requires only $3n$ memory locations and can be done in $7n^2$ flops. A direct comparison with (2.6) shows the close analogy between the BKO algorithm for Cauchy matrices and the BP algorithm for Vandermonde matrices.

Predictive partial pivoting and Leja ordering. In [BKO94] we obtained in another context a counterpart of Leja ordering for Cauchy matrices. More precisely, it turns out that a reordering of the $\{x_{1:n}\}$ that successively maximizes the quantities

$$|d_i| = |\frac{\prod_{j=1}^{i-1}(x_i - x_j)\;\prod_{j=1}^{i-1}(y_i - y_j)}{(x_i - y_i)\;\prod_{j=1}^{i-1}(x_i - y_j)\;\prod_{j=1}^{i-1}(x_j - y_i)}|, \qquad i = 1, \ldots, n, \tag{2.15}$$

often improves the numerical performance of several algorithms for Cauchy matrices. It was shown in [BKO94] that the above reordering corresponds to the permutation of the rows of $C(x_{1:n}, y_{1:n})$ that would be obtained by applying GEPP, but it can be rapidly performed *in advance*. Therefore it was called *predictive partial pivoting*.

Totally positive Cauchy matrices with monotonic nodes. In general, predictive partial pivoting improves the numerical performance of the BKO algorithm. Nonetheless, there exists a specific class of Cauchy systems for which it is not optimal. Moreover, it turns out that an appropriate analog of the positive monotonic ordering (2.8) of [Hig90] for Vandermonde matrices is

$$y_n \le y_{n-1} \le \ldots y_1 \le x_1 \le x_2 \le \ldots \le x_n\;, \qquad x_i, y_i \in \mathbb{R} \tag{2.16}$$

Both conditions, (2.8) and (2.16), are known to imply the total positivity of $V(x_{1:n})$ and $C(x_{1:n}, y_{1:n})$, respectively (see [GK50] or [K72] for the definition of this important concept, and for several applications). For totally positive Cauchy matrices we obtained counterparts of all the remarkable error bounds (2.9), (2.10), (2.11) and (2.12) for totally positive Vandermonde matrices. For example, if the components of the right-hand side have a sign-interchanging pattern :

$$(-1)^i \cdot f_i \ge 0, \tag{2.17}$$

then

$$|a - \widehat{a}| \le (10n - 5) \cdot u \cdot |a| + \mathcal{O}(u^2). \tag{2.18}$$

Thus, to first order in u, the relative error in the nonzero components of the computed solution is indeed bounded by a *quantity independent of the condition number of* $C(x_{1:n}, y_{1:n})$, and this bound is about as small as could possibly be expected. These analytical results validate the excellent numerical behavior of the BKO algorithm shown in Figure 2, and moreover, they identify the class (2.16) of Cauchy systems for which the BKO algorithm is guaranteed to provide an attractively high accuracy.

Table 2. Conditioning of Cauchy matrix with $x_i = i^4/n^4$, $y_i = -i^4/n^4$.

n	10	20	30	40	50	60
$\kappa_\infty(C(x_{1:n}, y_{1:n}))$	2e+08	4e+15	4e+21	4e+22	4e+22	7e+22

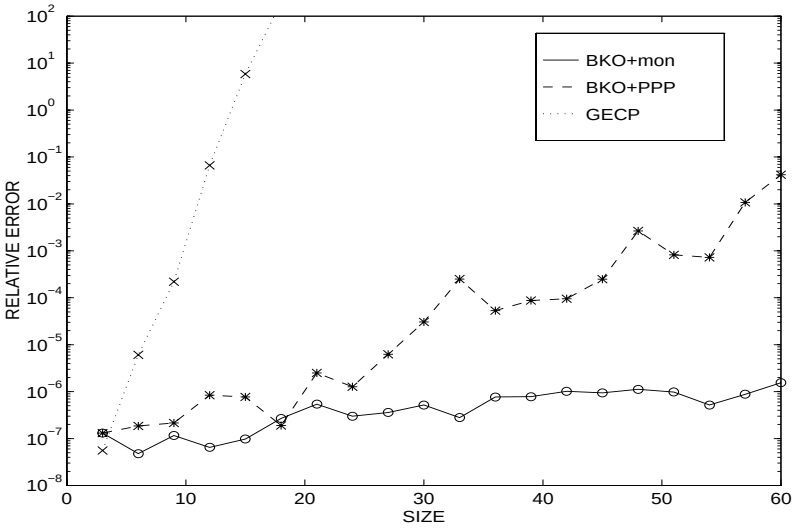


Figure 2: *Cauchy linear system with* $x_i = i^4/n^4$, $y_i = -i^4/n^4$, *and* $f_i = (-1)^i$, $(i = 1, \ldots, n)$. *The graphs display the relative error* $\|a - \widehat{a}\|_\infty / \|a\|_\infty$ *as a function of* n, *for the following three algorithms :* **BKO+mon:** *BKO algorithm with monotonically ordered nodes,* **BKO+PPP:** *BKO algorithm with predictive partial pivoting,* **GECP:** *Gaussian elimination with complete pivoting.*

More numerical examples will be given in section 5. We now turn to the derivation of the algorithm.

3 Derivation of the BKO algorithm.

Björck and Pereyra used polynomial language to obtain their factorization for $V(x_{1:n})^{-1}$. Here we derive a similar decomposition for $C(x_{1:n},\ y_{1:n})$ via purely matrix arguments.

Theorem 3.1 *The inverse of the Cauchy matrix* $C(x_{1:n},\ y_{1:n})$ *can be decomposed as*

$$C^{-1}(x_{1:n},\ y_{1:n}) = U_1\, U_2\, \ldots\, U_{n-1}\, D\, L_{n-1}\, \ldots\, L_2\, L_1, \tag{3.1}$$

where

$$L_k = \begin{bmatrix} I_k & \\ & \begin{matrix} \frac{1}{x_{k+1}-x_1} & & & \\ & \frac{1}{x_{k+2}-x_2} & & \\ & & \ddots & \\ & & & \frac{1}{x_n - x_{n-k}} \end{matrix} \end{bmatrix} \begin{bmatrix} I_{k-1} & \\ & \begin{matrix} 1 & 0 & & \\ -(x_1 - y_k) & (x_{k+1} - y_k) & & \\ & \ddots & \ddots & \\ & & -(x_{n-k} - y_k) & (x_n - y_k) \end{matrix} \end{bmatrix}, \tag{3.2}$$

$$U_k = \begin{bmatrix} I_{k-1} & \\ & \begin{matrix} 1 & -(x_k - y_1) & & \\ 0 & (x_k - y_{k+1}) & \ddots & \\ & \ddots & \ddots & -(x_k - y_{n-k}) \\ & & & (x_k - y_n) \end{matrix} \end{bmatrix} \begin{bmatrix} I_i & \\ & \begin{matrix} \frac{1}{y_1 - y_{k+1}} & & & \\ & \frac{1}{y_2 - y_{k+2}} & & \\ & & \ddots & \\ & & & \frac{1}{y_{n-k} - y_n} \end{matrix} \end{bmatrix} \tag{3.3}$$

$$D = \mathrm{diag}\{(x_1 - y_1), (x_2 - y_2), ..., (x_n - y_n)\}. \tag{3.4}$$

Proof. First, let us introduce a sequence of matrices $C^{(k)}(x_{k:n},\ y_{k:n})$, $k = 1, \ldots, n$

$$C^{(k)}(x_{k:n},\ y_{k:n}) \stackrel{\text{def}}{=} \left[\ \frac{1}{(x_i - y_j)(x_{i-1} - y_j)\ \ldots\ (x_{i-k+1} - y_j)}\ \right]_{i,j=k}^{n},$$

where $C^{(1)}(x_{k:n},\ y_{k:n}) = \left[\ \frac{1}{x_i - y_j}\ \right]_{i,j=1}^{n}$ is an ordinary Cauchy matrix. These matrices can be recursively computed from each other by applying to $C^{(k)}(x_{k:n}, y_{k:n})$ one step of a "non-pivotal" scheme of Gaussian elimination, in which the $(k,1)$ entry is used to eliminate the $(k+1,1)$ entry, namely

$$\begin{bmatrix} 1 & & & \\ -(x_1 - y_k) & (x_{k+1} - y_k) & & \\ & \ddots & \ddots & \\ & & -(x_{n-k} - y_k) & (x_n - y_k) \end{bmatrix} C^{(k)}(x_{k:n}, y_{k:n}) = \begin{bmatrix} * & * \cdots & * \\ 0 & & \\ \vdots & \mathcal{X} & \\ 0 & & \end{bmatrix}, \tag{3.5}$$

where the asterisk $*$ denotes nonrelevant entries. Indeed, $\mathcal{X}$ can be further factored as

$$\begin{aligned} \mathcal{X} &= \left[\ \frac{(x_i - y_k)}{\prod_{p=0}^{k-1}(x_{i-p} - y_j)} - \frac{(x_{i-k} - y_k)}{\prod_{p=0}^{k-1}(x_{i-1-p} - y_j)}\ \right]_{i,j=k+1}^{n} = \left[\ (x_i - x_{i-k})\ \frac{1}{\prod_{p=0}^{k}(x_{i-p} - y_j)}\ (y_k - y_j)\ \right]_{i,j=k+1}^{n} \\ &= \begin{bmatrix} x_{k+1} - x_1 & & \\ & \ddots & \\ & & x_n - x_{n-k} \end{bmatrix} C^{(k+1)}(x_{k+1:n}, y_{k+1:n}) \begin{bmatrix} y_k - y_{k+1} & & \\ & \ddots & \\ & & y_k - y_n \end{bmatrix}. \end{aligned} \tag{3.6}$$

A recursive application of the procedure (3.5), (3.6) shows that the matrix

$$L_{n-1}\, L_{n-2}\, \ldots L_1\, C(x_{1:n}, y_{1:n}) = U$$

is upper triangular.

Similar arguments applied to the Cauchy matrix $C^T(x_{1:n},y_{1:n})$=$C(y_{1:n},x_{1:n})$ imply that the matrix

$$U_{n-1}^T\, U_{n-2}^T\, \ldots U_1^T\, C^T(x_{1:n},\ y_{1:n}) = L^T,$$

is lower triangular. Here the U_k are given by (3.3). Since the L and U factors in the LU factorization of a non-singular matrix are unique up to a diagonal factor, we conclude that

$$C(x_{1:n},\ y_{1:n}) = L_1^{-1}\ \ldots\ L_{n-1}^{-1}\, D^{-1}\, U_{n-1}^{-1}\ \ldots\ U_1^{-1}$$

for some diagonal matrix D. Using the well-known Cauchy determinant formula to evaluate the successive leading minors of

$$D^{-1} = L_{n-1}\ \ldots\ L_1\, C(x_{1:n},\ y_{1:n})\ U_1\ \ldots\ U_{n-1},$$

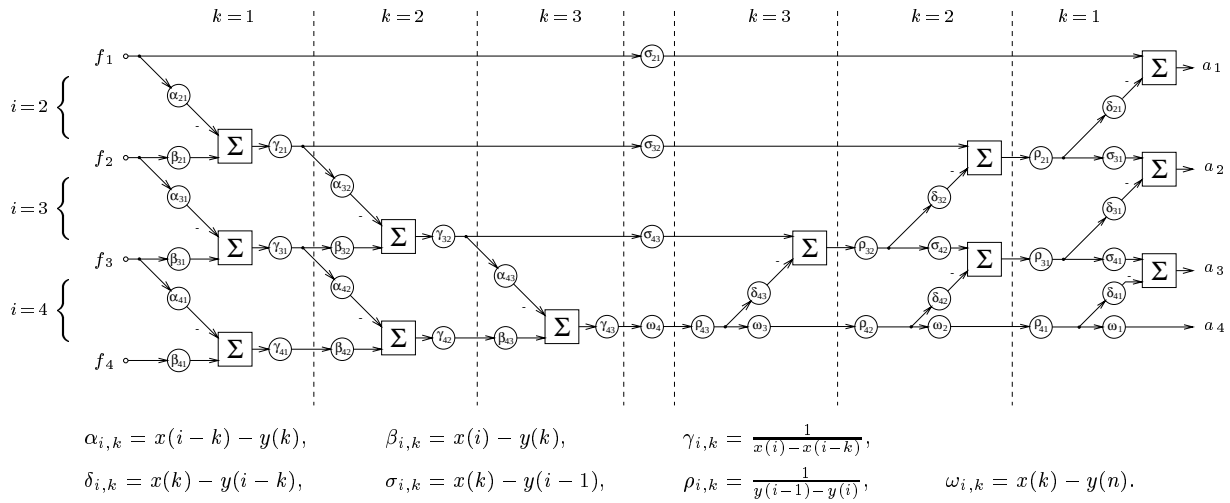
it is easy to verify that D is given by (3.4). ■

The following MATLAB program implements a fast Cauchy solver based on Theorem 3.1. In order to simplify this code we used a minor variation of (3.1), in which for $k = 1, 2, ..., n-1$ we interchange with each other the (k,k) entries of D and U_k (i.e. $(x_k - y_k)$ and 1, respectively).

Algorithm 3.2 [3] *Complexity: $7n^2$ flops.*

```
function a=BKO(n,x,y,f)
        a=f;
        for k = 1:(n-1)
            for i = n:-1:(k+1)
                a(i) = (a(i)*(x(i)-y(k)) - a(i-1)*(x(i-k)-y(k))) / (x(i)-x(i-k));
            end
        end
```

[3] Algorithm 3.2 has lower complexity and better error bounds than its earlier variant called Cauchy-III in [BKO94], which has been in circulation since 1994.



$\alpha_{i,k} = x(i-k) - y(k)$, $\beta_{i,k} = x(i) - y(k)$, $\gamma_{i,k} = \frac{1}{x(i)-x(i-k)}$,

$\delta_{i,k} = x(k) - y(i-k)$, $\sigma_{i,k} = x(k) - y(i-1)$, $\rho_{i,k} = \frac{1}{y(i-1)-y(i)}$, $\omega_{i,k} = x(k) - y(n)$.

Figure 3: Parallel implementation of the BKO algorithm.

```
        a(n)=a(n)*(x(n)-y(n));
        for k = (n-1):-1:1
            for i = k+1:n
                a(i) = a(i) / (y(i-k) - y(i));
                a(i-1) = a(i-1)*(x(k) - y(i-1)) - a(i)*(x(k) - y(i-k)) ;
            end
            a(n) = a(n) *(x(k) - y(n));
        end
return
```

Each of the two steps of the above algorithm consists of two nested loops. The computations in each inner loop are independent of each other, so that they can be performed on different processors. Therefore the parallel complexity of the BKO algorithm is $14n$ parallel flops with n processors (see, *e.g.*, Figure 3).

4 Rounding error analysis

In this section we perform an *a priori* forward and backward rounding error analysis for the BKO algorithm, adopting the standard model of floating point arithmetic

$$fl\{x \text{ op } y\} = (x \text{ op } y)(1+\epsilon)\,, \qquad |\epsilon| \le u\,, \tag{4.1}$$

where op $= +, -, *,$ or $/$, and u denotes the unit round-off. Our aim here is to obtain analogs of the nice error bounds for the BP algorithm for Vandermonde matrices.

General case. We first make no assumptions on the Cauchy linear system to be solved, and present general error bounds. However we then use these bounds to identify an important class of Cauchy linear systems for which the BKO algorithm is guaranteed to produce a very high accuracy. The analysis of the BKO algorithm, consisting of the $2n-1$ matrix-vector multiplications in (cf 3.1)

$$a = U_1 \cdots U_{n-1} D L_{n-1} \cdots L_1 f$$

is based on the observations, summarized in the next statement.

Lemma 4.1 *When Algorithm* 3.2 *is carried out in floating-point arithmetic with unit round-off u, the computed solution $\widehat{a}$ satisfies*

$$\widehat{a} = (U_1 + \Delta U_1) \cdots (U_{n-1} + \Delta U_{n-1})(D + \Delta D)(L_{n-1} + \Delta L_{n-1}) \cdots (L_1 + \Delta L_1) f \tag{4.2}$$

so that the following properties hold.

(a) *The matrices ΔU_k and ΔL_k are bidiagonal upper and lower triangular matrices, resp., satisfying*

$$|\Delta U_k| \le ((1+u)^5 - 1)|U_k|, \qquad |\Delta L_k| \le ((1+u)^5 - 1)|L_k|. \tag{4.3}$$

(b) *The perturbations of the inverse matrices are also nicely bounded:*

$$\begin{aligned} |(U_k + \Delta U_k)^{-1} - U_k^{-1}| &\le ((\frac{1+u}{1-u})^{10n} - 1)|U_k^{-1}|, \\ |(L_k + \Delta L_k)^{-1} - L_k^{-1}| &\le ((\frac{1+u}{1-u})^{10n} - 1)|L_k^{-1}|. \end{aligned} \tag{4.4}$$

Proof. Denote by B a single Jordan block of size n with eigenvalue 1, and let $*$ denote for the Hadamard (i.e. componentwise) product. Then (cf. the approach taken in in [S71]) the model (4.1) applied to each step of the algorithm implies that the matrices in (4.2) satisfy

$$\begin{aligned} L_k + \Delta L_K = L_k * \delta L_k \qquad &\text{with} \qquad (1-u)^5 B^T \le \delta L_k \le (1+u)^5 B^T, \\ U_k + \Delta U_K = U_k * \delta U_k \qquad &\text{with} \qquad (1-u)^5 B \le \delta U_k \le (1+u)^5 B, \end{aligned}$$

which implies (4.3).

The bounds in (4.4) are deduced from the observation that the matrices U_k and $U_k + \Delta U_k$, due to their bidiagonal structure, can be easily factored as

$$U_k = D_L B D_R, \qquad U_k + \Delta U_k = (D_L + \Delta D_L) B (D_R + \Delta D_R), \tag{4.5}$$

where all D's are diagonal matrices, satisfying

$$D_L + \Delta D_L = D_L * \delta D_L, \qquad D_R + \Delta D_R = D_R * \delta D_R,$$

with

$$(\frac{1-u}{1+u})^{5n} I \le \delta D_L \le (\frac{1+u}{1-u})^{5n} I, \qquad (\frac{1-u}{1+u})^{5n} I \le \delta D_R \le (\frac{1+u}{1-u})^{5n} I,$$

and with the same inequalities for δD_L^{-1} and δD_R^{-1}. From this it follows that

$$|(D_L + \Delta D_L)^{-1} - D_L^{-1}| \le (\frac{1+u}{1-u})^{5n} - 1)|D_L^{-1}|, \qquad |(D_R + \Delta D_R)^{-1} - D_R^{-1}| \le (\frac{1+u}{1-u})^{5n} - 1)|D_R^{-1}|.$$

The result in (4.4) now follows from (4.5) and the following easily verified fact : let $|\Delta X_k| \le \delta |X_k|$ for $k = 1, 2, ..., m$, then

$$|\prod_{k=1}^{m}(X_k + \Delta X_k) - \prod_{k=1}^{m} X_k| \le ((1+\delta)^{m-s} - 1) \prod_{k=1}^{m} |X_k|, \tag{4.6}$$

where s is the number of zero matrices among ΔX_k. ■

The above lemma implies bounds for forward, backward and residual errors associated with the computed solution. These bounds are formulated next and they immediately follow from (4.2), (4.3), (4.4) and (4.6).

Theorem 4.2 *Suppose that Algorithm 3.2 is carried out in floating-point arithmetic with unit roundoff u, and that no overflows were encountered during computation. Then the following three bounds hold for the computed solution $\widehat{a}$.*

(a) *The forward bound,*

$$|a - \widehat{a}| \le 5(2n+1) \cdot u \cdot |U_1|\,|U_2| \cdots |U_{n-1}|\,|D|\,|L_{n-1}| \cdots |L_2|\,|L_1|\,|f| + \mathcal{O}(u^2)\,. \tag{4.7}$$

(b) *The backward bound,*

$$|C(x_{1:n}, y_{1:n}) - \widehat{C}| \leq c_n \cdot u \cdot |L_1^{-1}| \cdots |L_{n-1}^{-1}|\,|D^{-1}|\,|U_{n-1}^{-1}| \cdots |U_1^{-1}| + \mathcal{O}(u^2)\ , \tag{4.8}$$

where $c_n = 20n(2n-1)$, and the matrix $\widehat{C}$ is such that the computed solution $\widehat{a}$ solves a nearby system $\widehat{C}\widehat{a} = f$.

(c) *The residual bound,*

$$|f - C(x_{1:n}, y_{1:n})\widehat{a}| \leq c_n \cdot u \cdot |L_1^{-1}| \cdots |L_{n-1}^{-1}|\,|D^{-1}|\,|U_{n-1}^{-1}| \cdots |U_1^{-1}|\,|\widehat{a}| + \mathcal{O}(u^2)\ . \tag{4.9}$$

The product of modulii of the $2n-1$ matrices on the right-hand sides of (4.7), (4.8), (4.9) can be rather large, which indicates potential instabilities of the algorithm 3.2 in a general situation. However, recall that the BP algorithm for Vandermonde matrices in general also can produce poor numerical results, being, it is very accurate for Vandermonde systems obeying (2.8). A natural question therefore is to look for an analog of (2.8) for Cauchy matrices and the BKO algorithm. As shown in the next section such an analog does exist.

Totally positive Cauchy matrices. Here we assume that two sets $\{y_k\}$ and $\{x_k\}$, defining a Cauchy matrix, can be separated from each other. Then the monotonic ordering

$$y_n \leq y_{n-1} \leq \ldots y_0 \leq x_0 \leq x_1 \leq \ldots \leq x_n\ , \qquad x_i, y_i \in \mathbb{R} \tag{4.10}$$

turns out to be an appropriate analog of (2.8). From the well-known formula (1.2) for the determinant of a Cauchy matrix it is easy to see that condition (4.10) is equivalent to the total positivity of $C(x_{1:n}, y_{1:n})$. Recall that totally positive matrices A are matrices all of whose minors are positive, i.e.

$$\text{for} \quad \text{all} \quad r, \quad i_1 < \ldots < i_r, \quad j_1 < \ldots < j_r, \quad \det A \begin{pmatrix} i_1, & \ldots, & i_r \\ j_1, & \ldots, & j_r \end{pmatrix} \geq 0.$$

see, *e.g.*, [GK50], [K72]. Interestingly, the counterpart condition (2.8) is also known to imply the total positivity for $V(x_{1:n})$, thus displaying one more similarity between these two matrix structures. More comments on total positivity can be found in Section 5 below.

Now observe that (4.10) means that the entries of all factors (3.2), (3.3), (3.4) in (3.1) are nonnegative. Therefore Theorem 4.2 immediately translates into the following statement.

Theorem 4.3 *Suppose that condition* (4.10) *holds (so that $C(x_{1:n}, y_{1:n})$ is totally positive), and Algorithm* 3.2 *is carried out in floating-point arithmetic with unit round-off u, and that no overflows were encountered during computation. Then the following bounds hold for the computed solution $\widehat{a}$.*

(a) *The forward bound,*

$$|a - \widehat{a}| \leq 5(2n+1) \cdot u \cdot |C(x_{1:n}, y_{1:n})^{-1}||f| + \mathcal{O}(u^2)\ . \tag{4.11}$$

(b) *The backward bound,*

$$|C(x_{1:n}, y_{1:n}) - \widehat{C}| \leq c_n \cdot u \cdot C(x_{1:n}, y_{1:n}) + \mathcal{O}(u^2)\ , \tag{4.12}$$

where $c_n = 20n(2n-1)$, and the matrix $\widehat{C}$ is such that the computed solution $\widehat{a}$ solves a nearby system $\widehat{C}\widehat{a} = f$.

(c) *The residual bound,*

$$|f - C(x_{1:n}, y_{1:n})\widehat{a}| \leq c_n \cdot u \cdot C(x_{1:n}, y_{1:n})|\widehat{a}| + \mathcal{O}(u^2)\ . \tag{4.13}$$

We next justify that the bounds in the latter theorem are very satisfactory.

5 Implications for stability

Forward stability. Consider an *"ideal"* situation, in which **(i)** the points $\{x_i\}$ and $\{y_i\}$, defining the Cauchy matrix can be exactly represented on the computer **(ii)** the computation is performed in exact arithmetic, i.e. there are no roundoff errors in solving the system $C(x_{1:n}, y_{1:n}) \cdot a = f$. In this case the accuracy of the computations may be limited only because of the roundoff errors that occur when storing the components of the right-hand side vector f on the computer. Let us therefore discuss the sensitivity of the solution of $A \cdot a = f$ to a small componentwise perturbation in the right-hand side f alone. The appropriate condition number associated with this map $M_A : f \to a$, formally defined as

$$\theta(A, f) = \lim_{\varepsilon \to 0} \sup_{|\triangle f| \leq \varepsilon |f|} \frac{\|\triangle a\|_\infty}{\|a\|_\infty \cdot \varepsilon}, \tag{5.1}$$

was shown in [S71] to be given by $\theta(A, f) = \frac{\| \, |A^{-1}| \cdot |f| \, \|_\infty}{\|a\|_\infty}$. Thus (4.11) can be rewritten in norm-wise fashion as follows

$$\frac{\|a - \widehat{a}\|_\infty}{u \|a\|_\infty} \leq 5(2n+1)\, \theta(C(x_{1:n}, y_{1:n}), f) + \mathcal{O}(u) \ , \tag{5.2}$$

which means that *the BKO algorithm introduces no more uncertainty into the numerical solution than was already present in the machine right-hand side vector.* A similar fact was observed in [Hig87] (see also [Hig96]) for the BP algorithm[4]. Finally, (5.2) and

$$1 \leq \theta(A, f) \leq \kappa_\infty(A), \qquad \text{where} \qquad \kappa_\infty(A) = \|A^{-1}\|_\infty \cdot \|A\|_\infty \tag{5.3}$$

suggest an explanation for the phenomena of the higher accuracy of BKO compared to standard numerically stable algorithms, *e.g.*, Gaussian elimination with complete pivoting, whose numerical behavior is predicted by the regular condition number $\kappa_\infty(A)$.

Backward stability. In order to interpret the backward error bound in theorem 4.3 recall the result of [GK50] (see also [K72]) that for a totally positive matrix all the entries of the L and U factors are positive. de Boor and Pincus [DBP77] used this fact to show that if during the elimination the entries of the computed factors $\widehat{L}, \widehat{U}$ remain positive, then the backward error is pleasantly small :

$$(A + \Delta A)\widehat{a} = f, \qquad \text{with} \qquad |\Delta A| \leq 3\gamma_n A \qquad \text{where} \qquad \gamma_n = \frac{nu}{1 - nu}. \tag{5.4}$$

which led them to the recommendation not to pivot with totally positive matrices. The similarity of (4.12) and (5.4) suggests that the backward stability of the BKO algorithm for totally positive Cauchy matrices is related to that of Gaussian elimination without pivoting on $C(x_{1:n}, y_{1:n})$. A similar conclusion can be made for the BP algorithm for Vandermonde matrices.

Note however, that there are situations in which the results obtained for the BKO algorithm are more favorable than those of [DBP77]. Indeed, the analysis in [DBP77] is limited to the case where the computed factors $\widehat{L}$ and $\widehat{U}$ have only nonnegative entries. Since the condition number of the Hilbert matrices is known to grow exponentially with the size, already for small n we have $k_2(H) > \frac{1}{q_n u}$. Then in accordance with [W68] the matrix H will likely lose during computation not only its total positivity, but also the weaker property of being positive definite. Correspondingly, the single precision LAPACK routine **sposv** for Cholesky factorization, when applied to the Hilbert matrix exits with error flag already for $n = 9$, warning that $\widehat{L}$ and $\widehat{U}$ became negative, so the pleasing backward bound (5.4) is no longer valid for Gaussian elimination. In contrast, the BKO algorithm was obtained by exploiting the Cauchy structure, resulting in that the factors $L_{n-1} \ldots L_1$ and $U_1 \ldots U_{n-1}$ have positive entries by construction. Therefore the bound (4.12) is valid while there is no overflows during the computation. Indeed, Figure 4 shows that the BKO algorithm works with remarkably accuracy until $n = 25$, for larger systems one usually cannot store the solution in single precision.

[4] For the first part of the BP algorithm, i.e. the divided differences table this fact was noted earlier by W.Kahan, who mentioned in [KF63] that if $\{x_k\}$ are in the monotonic order, "*then the effect of roundoff upon any n-th divided difference is no more than would be caused by perturbing each $f(x_i)$ by n units at most in its last significant place.*"

Residual error. It is well-known that the residual errors corresponding to the solution $\widehat{a}$ and to the triangular factors $\widehat{L}$ and $\widehat{U}$ computed by Gaussian elimination, admit the following componentwise bounds,

$$|f - A\widehat{a}| \leq 2\gamma_n |\widehat{L}||\widehat{U}||\widehat{a}|, \tag{5.5}$$

see, *e.g.*, [SB80], and

$$|A - \widehat{L}\widehat{U}| \leq \gamma_n |\widehat{L}||\widehat{U}|, \tag{5.6}$$

see, *e.g.*, [DBP77]. Both bounds involve the matrix $|\widehat{L}||\widehat{U}|$, which can be much larger that $|\widehat{L}\widehat{U}|$, depending upon the sign pattern of the entries. An "*ideal*" situation would be if the sign pattern of the entries of $\widehat{L}$ and $\widehat{U}$ would exclude any cancellation. Let us therefore again recall that for a totally positive matrix A, the entries of its L and U factors are nonnegative. If this fact holds also for the entries of the computed factors $\widehat{L}$ and $\widehat{U}$, then (5.6) implies $|\widehat{L}||\widehat{U}| \leq \frac{1}{1-\gamma_n}|A|$, so that (5.5) implies that the residual is nicely bounded :

$$|f - A\widehat{a}| \leq 2\frac{\gamma_n}{1-\gamma_n} A|\widehat{a}|. \tag{5.7}$$

The similarity of (4.13) and (5.7) suggests that the size of the residual error for the BKO algorithm applied to totally positive Cauchy matrices is related to that of Gaussian elimination without pivoting on $C(x_{1:n}, y_{1:n})$. A similar conclusion can be made for the BP algorithm for Vandermonde matrices. But again, the favorable bounds for Gaussian elimination are limited to the case where the computed factors $\widehat{L}$ and $\widehat{U}$ remain positive, whereas the residual bound (4.13) holds for the BKO algorithm, provided there are no overflows.

Avoidance of pivoting. Recall the recommendation of [DBP77] to avoid pivoting with totally positive matrices when using standard Gaussian elimination, see, *e.g.* the bounds (5.4) and (5.7). Interestingly, in some instances this occurrence matches our recommendation on how to use ordering of the nodes to solve Cauchy systems in a stable manner. Indeed, the pleasing bound (4.12) suggests to use the BKO algorithm combined with the ordering (4.10) (i.e. to not destroy the total positivity of $C(x_{1:n}, y_{1:n})$ by permuting its row and columns, or equivalently, the nodes $\{x_{1:n}\}$ and $\{y_{1:n}\}$). However if one cannot separate two point sets $\{x_{1:n}\}$ and $\{y_{1:n}\}$, so that they cannot be reordered so that (4.10) holds, then we recommend to use predictive partial pivoting, see section 2 for the definition and [BKO94] for details. Moreover, the predictive partial pivoting technique enhances even better the numerical performance of another algorithm for solving Cauchy systems designed in [BKO94]. A similar recommendation can be given for the use of of different orderings with the fast algorithms for solving Vandermonde systems.

Sign-interchanging right-hand side. Consider now the case where along with condition (4.10), we know that the right-hand side of $C(x_{1:n}, y_{1:n})a = f$ has the following sign-interchanging pattern :

$$(-1)^i \cdot f_i \geq 0. \tag{5.8}$$

Since $C(x_{1:n}, y_{1:n})$ is totally positive, its inverse also has sign-oscillating property, so that $|C(x_{1:n}, y_{1:n})||f| = |C(x_{1:n}, y_{1:n})f|$. Therefore (4.11) implies the attractive bound,

$$|a - \widehat{a}| \leq 5(2n+1) \cdot u \cdot |a| + \mathcal{O}(u^2). \tag{5.9}$$

Thus, to first order in u, the relative error in the nonzero components of the computed solution is indeed bounded by a *quantity independent of the condition number of* $C(x_{1:n}, y_{1:n})$, and this bound is about as small as could possibly be expected. These analytical results identify the class (4.10), (5.8) of Cauchy systems for which the BKO algorithm is guaranteed to provide an extremely high forward accuracy. Interestingly, this class includes Hilbert linear systems, which are traditionally considered to be too ill-conditioned to be attacked. Table 3 and Figure 4 show that also with this "difficult" matrix, the BKO algorithm behaves as predicted by (5.9), providing about 7 correct digits from about 7 possible in single precision. We had to stop our experiment for $n = 25$, because for the larger sizes we simply could not store the solution in single precision. Notice also the failure of numerically stable Gaussian elimination with complete pivoting, consistent with the extreme ill-conditioning of the coefficient matrix, as shown in Table 3.

Table 3. Conditioning of Hilbert matrix.

n	5	10	15	20	25
$\kappa_\infty(C(x_{1:n}, y_{1:n}))$	5e+05	2e+13	2e+17	8e+17	1e+19

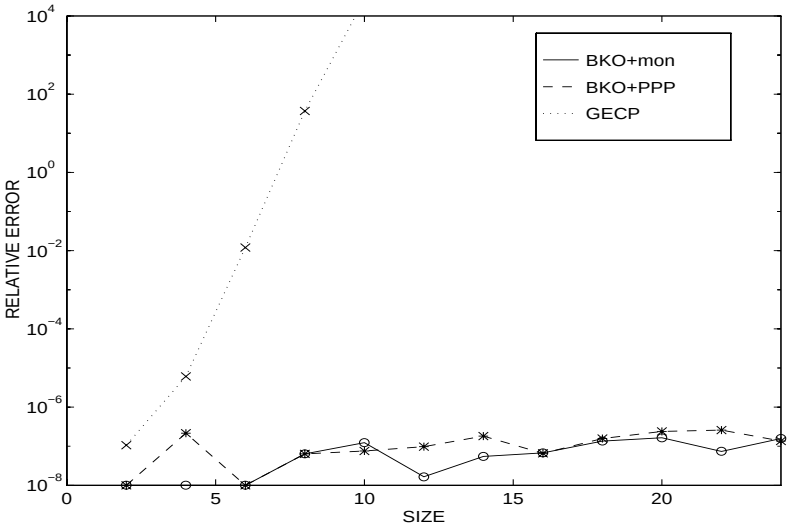


Figure 4: *Hilbert linear system with* $x_i = i$, $y_i = -j+1$, *and* $f_i = (-1)^i$, $(i = 1, \ldots, n)$. *The graphs display the relative error* $\|a - \widehat{a}\|_\infty / \|a\|_\infty$ *as a function of* n, *for the following three algorithms :* **BKO+mon:** *BKO algorithm with monotonically ordered nodes,* **BKO+PPP:** *BKO algorithm with predictive partial pivoting,* **GECP:** *Gaussian elimination with complete pivoting.*

In the next section we use the obtained results to illustrate numerically a connection of the high accuracy of the BKO algorithm with two concepts well-known in numerical linear algebra and matrix theory.

6 Numerical illustrations for effective well-conditioning and total positivity

Connection with effective well-conditioning. In [CF88] Chan and Foulser, motivated by the high accuracy of the BP algorithm for sign-oscillating right-hand side, and by some other examples, introduced the concept of *effective well-conditioning*. The latter reflects the fact that the sensitivity to perturbations of a solution depends upon the direction of the right-hand side vector, which suggests the potential existence of algorithms that can exploit the effective well-conditioning to produce higher accuracy for special right-hand sides. Note, however, that in general the numerical performance of a certain algorithm and the effective well-conditioning of a system to be solved have nothing to do with each other, and many algorithms are insensitive to the direction of the right-hand side vector. For example, for Gaussian elimination this point is illustrated by the numerical examples below. Here we use the approach of [CF88] to show that our new Cauchy solver is indeed an example of an algorithm that is guaranteed to accurately solve effectively well-conditioned Cauchy systems with totally positive coefficient matrices. We also observe that the BP algorithm applied to totally positive Vandermonde matrices has the same property. Before stating the result, let us introduce the necessary notations.

Recall that $\theta(A, f)$ in (5.1) measures, in the infinity-norm, the sensitivity of the solution of $Aa = f$ to small componentwise bounded perturbations in the right hand-side. The 2-norm condition number

associated with the same map is given by

$$\kappa_2(A,f) = \lim_{\varepsilon \to 0} \sup_{\|\triangle f\|_2 \le \varepsilon \|f\|_2} \frac{\|\triangle a\|_2}{\|a\|_2 \cdot \varepsilon}, \tag{6.1}$$

i.e. it assumes that both the perturbation in f and the variation in a are measured in the 2-norm. Chan and Foulser derived an upper bound for $\kappa_2(A,f)$ in terms of the direction of the right-hand side vector relative to the left singular vectors of A. To be more precise, let $A = U \cdot \Sigma \cdot V^T$ be the singular value decomposition, where the columns of $U = \left[\begin{array}{cccc} u_1 & u_2 & \cdots & u_n \end{array} \right]$ are the left singular vectors, the columns of $V = \left[\begin{array}{cccc} v_1 & v_2 & \cdots & v_n \end{array} \right]$ are the right singular vectors, and the diagonal entries of $\Sigma = \mathrm{diag} \left\{ \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{array} \right\}$ are the singular values of A in decreasing order. Denote by $P_k = \left[\begin{array}{ccc} u_{n+1-k} & \cdots & u_n \end{array} \right] \cdot \left[\begin{array}{ccc} u_{n+1-k} & \cdots & u_n \end{array} \right]^T$ the projection operator onto the linear span of the smallest k left singular vectors of A. Then, in accordance with [CF88],

$$\kappa_2(A,f) \le \gamma(A,f) \le \kappa_2(A), \tag{6.2}$$

where

$$\gamma(A,f) = \min_k \frac{\sigma_{n-k+1}}{\sigma_n} \cdot \frac{\|f\|_2}{\|P_k f\|_2}, \tag{6.3}$$

and $\kappa_2(A) = \|A^{-1}\|_2 \cdot \|A\|_2$ is the 2-norm condition number. Note that the second inequality in (6.2) can easily be deduced by inspecting (6.3) for $k = n$. At the same time, if a large part of f lies in the span of the small left singular vectors of A, then $\gamma(A,f)$ can be much smaller than $\kappa_2(A)$, thus providing a better bound for $\kappa_2(A,f)$ in (6.2). Linear systems for which the *Chan-Foulser number* in (6.3) is much smaller than $\kappa_2(A)$ are called *effectively well-conditioned*. We shall refer to an algorithm as *effectively (forward) stable*, if it is guaranteed to produce high relative accuracy for effectively well-conditioned systems. Backward effective stability can be defined in the same way.

Now observe that by using the equivalence of the norms, we obtain from the first inequality in (6.2) that the quantities in (6.3) and (5.1) are related by

$$\theta(A,f) \le \sqrt{n} \cdot \gamma(A,f). \tag{6.4}$$

This and (5.2) imply that the relative error in the solution computed via the BKO algorithm is bounded by a multiple of the Chan-Foulser number :

$$\frac{\|a - \widehat{a}\|_\infty}{\|a\|_\infty} \le (19n+2)\sqrt{n} \cdot \gamma(C(x_{1:n}, y_{1:n}), f) \cdot u + \mathcal{O}(u^2). \tag{6.5}$$

Thus, (6.5) shows that *the BKO algorithm is effectively stable for totally positive Cauchy systems.* Table 5 below illustrates these results with a computed example, in which we solved sixteen linear systems with 16×16 Hilbert coefficient matrix using its seven left singular vectors u_k for the right-hand sides.

Table 5. Left singular vectors for the right-hand sides.

	Conditioning				Relative error		
i	$\kappa_2(C)$	$\theta(C,f)$	$\sqrt{n}\cdot\gamma(C,f)$	$\|\widehat{a}_{d1}\|_\infty$	BKO	Inversion formula	GEPP
1	5e+17	1e+16	4e+16	4.9e+05	1e+00	1e+00	4e+04
2	5e+17	2e+16	4e+16	5.0e+05	1e+00	1e+00	1e+04
3	5e+17	2e+16	4e+16	5.7e+05	1e+00	1e+00	6e+03
4	5e+17	2e+14	6e+15	1.4e+07	1e+00	1e+00	2e+04
5	5e+17	2e+13	5e+14	1.7e+08	1e+00	1e+00	3e+04
6	5e+17	1e+12	3e+13	3.6e+09	1e+00	1e+00	1e+05
7	5e+17	3e+11	2e+12	1.8e+10	1e+00	1e+00	8e+03
8	5e+17	4e+09	8e+10	1.2e+12	1e+00	1e+00	2e+04
9	5e+17	2e+09	3e+09	2.9e+12	6e-01	1e+00	3e+03
10	5e+17	9e+07	9e+07	6.2e+13	7e-01	8e-01	1e+05
11	5e+17	1e+05	2e+06	4.3e+16	2e-03	1e-02	3e+08
12	5e+17	3e+04	4e+04	2.1e+17	4e-04	2e-03	1e+09
13	5e+17	8e+02	5e+02	8.7e+18	2e-05	8e-05	7e+09
14	5e+17	1e+00	1e+01	9.4e+21	3e-07	2e-07	9e+12
15	5e+17	2e+00	5e+00	4.5e+21	1e-07	1e-07	3e+12
16	5e+17	3e+00	4e+00	3.2e+21	4e-08	2e-07	4e+12

The numerical data do not demonstrate any dependence of the accuracy of Gaussian elimination with complete pivoting (GECP) upon the direction of the right-hand side vector. Moreover, in all 16 cases the GECP algorithm does not produce even one correct digit in the solution from about 7 possible-in-single-precision, consistent with the large condition number of the coefficient matrix, displayed in the second column.

The same table shows that the numerical behavior of the BKO algorithm indeed depends upon the direction of the right-hand side vector, thus confirming the analytical results in (5.2), (6.5). Moreover, when the largest eight left singular vectors $u_1, ..., u_8$ are used for the right-hand side, then the quantities $\theta(C(x_{1:n}, y_{1:n}), f)$ and $\sqrt{n}\cdot\gamma(C(x_{1:n}, y_{1:n}), f)$, displayed in the 3-rd and the 4-th columns of Table 5, are are bigger than the reciprocal of the machine precision, $\frac{1}{u} \approx 10^8$. Correspondingly, the BKO algorithm performs similar to GECP. At the same time, when we used three smallest left singular vectors u_{14}, u_{15} and u_{16} for the right-hand sides, the numbers $\theta(C(x_{1:n}, y_{1:n}), f)$ and $\sqrt{n}\cdot\gamma(C(x_{1:n}, y_{1:n}), f)$ are of the order of unity and much smaller that $\kappa_2(C(x_{1:n}, y_{1:n}))$. Correspondingly, the BKO algorithm performs much better than GECP, giving now about 7 correct digits from about 7 possible-in-single-precision. The same table shows that the numerical behavior of applying the inversion formula (1.4) also depends upon the effective well-conditioning of a system to be solved.

Here we may note that (6.4), combined with the bound of [Hig87], implies that if (2.8) holds, then the solution $\widehat{a}$ of a Vandermonde system computed by the BP algorithm satisfies

$$\frac{\|a-\widehat{a}\|_\infty}{\|a\|_\infty} \leq 5n\sqrt{n}\cdot\gamma(V(x_{1:n}), f)\cdot u + \mathcal{O}(u^2). \tag{6.6}$$

The latter bound proves that *the BP algorithm is also effectively stable for totally positive Vandermonde systems.* This conclusion justifies the motivation of [CF88], and gives a theoretical support for the numerical examples in [V93]. Moreover, our experiments indicate that the numerical behavior of the Parker-Traub inversion algorithm depends upon the effective well-conditioning of a Vandermonde system to be solved.

Length of the solution vector. Observe that in the example of Table 5 the norm $\|\widehat{a}\|_\infty$ of the solution vector is larger for effectively well-conditioned systems, i.e. for which the quantity $\gamma(A, f)$ displayed in the 4-th column is small. In fact this occurrence is not quite unexpected. Indeed, let $f = \sum_{k=1}^n \alpha_k \cdot u_k$ be the decomposition of the right-hand side vector f in the basis of left singular vectors u_k. Then the solution of $A\cdot a = f$ is clearly given by the following linear combination of the right singular vectors :

$$a = \sum_{k=1}^{n} \frac{\alpha_k}{\sigma_k}\cdot v_k,$$

i.e. the larger the quantities $\|a\|_2 = \sqrt{\sum_{k=1}^n \frac{\alpha_k^2}{\sigma_k^2}}$ and $\|a\|_\infty = \max_k \frac{\alpha_k}{\sigma_k}$, the larger the part of f lies in the linear span of the small left singular vectors of A. This observation can be used to explain that the result

in (6.2) is natural. Indeed observe that since $\triangle a = A^{-1} \cdot \triangle f$ does not depend on the right-hand side f, the size of

$$\sup_{\|\triangle f\|_2 \leq \epsilon \|f\|_2} \frac{\|\triangle a\|_2}{\|a\|_2}$$

is determined only by the length of the solution a. Thus, inspecting (6.1) one sees that it is not surprising that the sensitivity of the solution to small changes in the right-hand side vector should be smaller for effectively well-conditioned systems, which have larger solution vectors.

The sign pattern of the right-hand side. Here we use the results obtained in section 3 to illustrate by computed example that many of the systems obeying (4.10) are effectively well-conditioned, so the high accuracy of the BKO algorithm should not be surprising.

To this end recall that in accordance with [GK50] the singular vectors of a totally positive matrix have the property that the number of sign oscillations in the components of any linear combination f of $u_p, u_{p+1}, ..., u_q$ is between $p-1$ and $q-1$. In particular, the last left singular vector u_n corresponding to the smallest singular value has the sign-interchanging property (5.8). Moreover, many of of the right-hand sides obeying (5.8) will be far enough from being orthogonal to u_n, and hence they will be effectively well-conditioned.

The next table displays the results of a numerical experiment, where we solved eleven 22×22 Hilbert systems using right hand sides $f^{(k)}$ $(k = 0, 1, 2, ..., 11)$, with fixed $|f_i^{(k)}| = 1$, but with different distribution of sign$(f_i^{(k)})$. More precisely we started with $f^{(0)} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$, adding at each step one more "minus", so that $f^{(k)}$ has exactly $2k$ sign oscillations in its components.

Table 6.

	Conditioning				Relative error		
i	$\kappa_2(C)$	$\theta(C,f)$	$\sqrt{n} \cdot \gamma(C,f)$	$\|\widehat{a}_{d1}\|_\infty$	BKO	Inversion formula	GEPP
0	2e+18	2e+15	6e+09	2.0e+16	3e-01	1e+00	4e+11
1	2e+18	1e+12	1e+07	2.6e+19	2e-04	1e+00	2e+12
2	2e+18	3e+08	2e+04	1.5e+23	2e-07	1e+00	3e+14
3	2e+18	5e+05	5e+02	7.2e+25	1e-07	9e-02	9e+16
4	2e+18	5e+03	9e+01	7.4e+27	9e-08	8e-04	4e+18
5	2e+18	2e+02	4e+01	2.3e+29	2e-07	3e-05	2e+20
6	2e+18	1e+01	2e+01	2.6e+30	1e-07	2e-06	3e+21
7	2e+18	3e+00	2e+01	1.2e+31	1e-07	5e-07	1e+22
8	2e+18	1e+00	9e+00	2.7e+31	2e-07	2e-07	1e+22
9	2e+18	1e+00	1e+01	3.6e+31	1e-07	1e-07	4e+22
10	2e+18	1e+00	1e+01	3.7e+31	1e-07	1e-07	4e+22
11	2e+18	1e+00	9e+00	3.7e+31	9e-08	1e-07	5e+22

Loosely, one can say that in this case the right-hand side $f^{(k)}$ migrates from the large left singular vectors towards the small ones. Correspondingly, the quantity $\gamma(C(x_{1:n}, y_{1:n}), f)$ in the 4-th column of Table 6 becomes smaller, and hence the corresponding system becomes more effectively well-conditioned. Since we are changing only the sign distribution in the components of the right-hand side, the value $\| \ |C(x_{1:n}, y_{1:n})^{-1}| \ \cdot |f^{(k)}| \ \|_\infty$ remains constant. However, the above migration of the right-hand side causes growth of the solution norm, see, *e.g.*, the 5-th column of Table 6, therefore decreasing the quantity $\theta(C(x_{1:n}, y_{1:n}), f)$ in (5.2). Thus both $\gamma(C(x_{1:n}, y_{1:n}), f)$ and $\theta(C(x_{1:n}, y_{1:n}), f)$ decay (note, however, the discrepancy with (6.4) caused by the roundoffs), and, in accordance with the bounds in (5.2) and (6.5), the BKO algorithm produces more and more accurate solutions. Table 6 also indicates that the numerical behavior of applying the inversion formula (1.4) also depends upon the direction of the right-hand side vector, though giving in this example less accuracy than the BKO algorithm.

7 Conclusions

In this paper we designed a fast low-storage BKO algorithm for Cauchy linear systems that matches the well-known Björck-Pereyra algorithm (the BP algorithm) for Vandermonde matrices. Our investigation has provided several insights into the numerical behavior of both algorithms, showing that for the important class of totally positive matrices, these algorithms are *forward and backward stable*, and moreover they are guaranteed to produce a favorably high accuracy, independently of conditioning. This extends the

prior N.J.Higham's result [Hig87] on the favorable forward accuracy of the BP algorithm for Vandermonde matrices. The obtained results were used to illustrate the relationship of the high accuracy of the BKO algorithm to the concept of effective well-conditioning. The theoretical and numerical results demonstrate clearly that the ordering of the points has a profound influence on the accuracy of the computed solution.

In summary we make the following recommendation on how to use different orderings to rapidly solve Cauchy linear systems in a stable manner. If the points satisfy the condition $x_i < y_j$ for all i, j, then use the BKO algorithm and the monotonic ordering (4.10). For other cases, where one cannot separate the two sets $\{x_{1:n}\}$ and $\{y_{1:n}\}$, predictive partial pivoting (2.15) becomes numerically preferable, as described in [BKO94].

The present contribution indicates a striking resemblance between the numerical properties of Cauchy matrices and the much-studied Vandermonde matrices. The results illustrate that the structure of a matrix often allows us not only to speed-up computations, but also to achieve much more accuracy compared to standard numerically stable algorithms.

References

[B37] J.P.M. Binet, *Observations sur des théoremes de Géométrie, énoncées page 160 de ce volume et page 222 du volume et page 222 du volume précédent*, Journ. (de Liouville) de Math, ii. (1837), . 248-252.

[BGR90] J.Ball, L.Rodman and I.Gohberg, *Interpolation of Rational Matrix Functions*, Birkhauser Verlag, Basel, 1990.

[BKO94] T.Boros, T.Kailath and V.Olshevsky, *Pivoting and Backward Stability of Fast Cauchy solvers*, preprint, 1994. www.cs.gsu.edu/~matvro

[BP70] A.Björck and V.Pereyra, *Solution of Vandermonde Systems of Equations*, Math. Comp., **24** (1970), 893-903.

[C41] A.L. Cauchy, *Mémorie sur les fonctions alternées et sur les sommes alternées*, Exercices d'analyse et de phys. math., ii (1841), 151-159.

[CF88] T.Chan and D.Foulser, *Effectively well-conditioned linear systems*, SIAM J. Sci. Stat. Computations, 9 (1988), 963 – 969.

[CR92] D.Calvetti and L.Reichel, *A Chebyshev-Vandermonde solver*, Lin. Alg. Appl., **172**(1992), 219-229.

[CR96] D.Calvetti and L.Reichel, *On the solution of Cauchy systems of equations*, Elec. Trans. Numer. Anal., 4 (1996), 125-136.

[CR97] D.Calvetti and L.Reichel, *Factorizations of Cauchy matrices*, J. Computational And Applied Mathematics (86)1 (1997) 103-123.

[D97] J.Demmel, *Accurate SVD of Structured matrices*, preprint, 1997.

[DG97] J.Demmel, Ming Gu, Stanley Eisenstat, Ivan Slapnikar, Kresemir Veselić, Zlatco Drmač, *Computing the singular value decomposition with high relative accuracy*, LAPACK working note 119, 1997

[DBP77] C. de Boor and A. Pincus, *Backward error analysis for totally positive linear systems*, Numer. Math., **27** (1977), 485-490.

[F66] G.Forney, *Concatenated codes*, The M.I.T. Press, 1966, Cambridge.

[Gas60] N.Gastinel, *Inversion d'une matrice generalisant la matrice de Hilbert*, Chiffres, **3**(1960), 149-152.

[GI88] W. Gautschi and G.Inglese, *Lower bounds for the condition number of Vandermonde matrix*, Numer. Math., **52** (1988), 241 – 250.

[GK50] F.R.Gantmacher and M.G.Krein, *Oscillatory matrices and kernels, and small vibrations of mechanical systems*, second edition, (in Russian), GITTL, Moscow, 1950. *German translation :* Oszillationsmatrizen, Oszillationskerne und kleine Schwingungen mechanischer Systeme, Berlin, Akademie Verlag, 1960.

[GK90] I.Gohberg and I.Koltracht, *On the inversion of Cauchy matrices*, In Signal processing, Scattering and Operator Theory, and Numerical methods. Proc of the MTNS-89 (M.A.Kaashoek, J.H.van Schuppen and A.C.M.Ran, eds), 381-392, Birkhäuser, Boston, MA, 1990.

[GK93] I.Gohberg and I.Koltracht, *Mixed, componentwise and structured condition numbers*, SIAM J. Matrix Anal. Appl., **14**(1993), 688–704.

[GO94] I.Gohberg and V.Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, **20, No. 1** (1994), 44-83.

[GKO95] I.Gohberg, T.Kailath and V.Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. of Comp., **64** (1995), 1557-1576.

[GVL89] G. Golub and C, Van Loan, *Matrix Computations*, second edition, John Hopkins U. P., Baltimore, 1989.

[Hig87] N.J.Higham, *Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems*, Numer. Math., 50 (1987), 613 – 632.

[Hig88] N.J.Higham, *Fast solution of Vandermonde-like systems, involving orthogonal polynomials*, IMA J. Numer. Anal., **8** (1988), 473-486.

[Hig90] N.J.Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., **11**(1) (1990), 23–41.

[Hig96] N.J.Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[HLM95] C.He, A.Laub and V.Mehrmann, *Placing plenty of poles is pretty preposterous*, preprint, 1995.

[KF63] W.Kahan and I.Farkas, *Algorithms 167 - 169*, Comm. ACM, **6** No 4, April 1963, 164-165; see also the *certification*, isid, **6** No 9, September 1963, 523.

[K32] G. Kowalewski, *Interpolation und genäherte Quadratur*, Teubner, Berlin, 1932.

[K72] S.Karlin, *Total Positivity*, vol 1, Stanford University Press, Stanford, 1972.

[KO95] T.Kailath and V.Olshevsky *Displacement structure approach to Chebyshev–Vandermonde and related matrices*, Integral Equations and Operator Theory, **22**, 1995, 65 – 92.

[KO96] T. Kailth and V. Olshevsky, *Displacement structure approach to discrete transform based preconditioners of G.Strang type and of T.Chan type, Calcolo*, **33**(1996), pp. 191-208.

[KO97] T.Kailath and V.Olshevsky, *Diagonal Pivoting for Partially Reconstructible Cauchy-like Matrices, With Applications to Toeplitz-like Linear Equations and to Boundary Rational Matrix Interpolation Problems*, *Linear Algebra and Its Applications*, **254** (1997), 251-302.

[KS95] T. Kailath and A.H.Sayed, *Displacement structure : theory and applications*, SIAM Review, **37** No.3 (1995), 297-386.

[M30] T. Muir, *A treatise on the theory of determinants*, Albany, New York, 1930.
see also Dover edition, 1960.

[MS77] F.J. McWilliams and N.J.Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, Amsterdam, 1977.

[MP98] J.J. Martnez, J.M. Peña, *Factorizations of Cauchy–Vandermonde matrices*, Linear Algebra And Its Applications (284)1-3 (1998) 229-237.

[O97] V.Olshevsky, *Pivoting for structured matrices with applications*, 52p, www.cs.gsu.edu/ ˜ matvro

[OP98] V.Olshevsky and V.Pan, *A unified superfast algorithm for boundary rational tangential interpolation problems and for inversion and factorization of dense structured matrices*, Proc. of 39th Annual Symposium on Foundations of Computer Science, FOCS'98, IEEE Computer Society, Los Alamitos, CA, 1998, 192-201.

[OS86] A.Odlyzko and A.Schönage, *Fast algorithms for multiple evaluations of the Riemann zeta function*, Technical report, AT&T Bell Labs, Murray Hill, N.J., 1986.

[OS99] V.Olshevsky and A. Schokrollahi, *A displacement approach to efficient decoding of algebraic-geometric codes*, to appear in Proc. of the International Symposium of Theory of Computing, 10 p.

[P64] F.Parker, *Inverses of Vandermonde matrices*, Amer. Math. Monthly, 71 (1964), 410 - 411.

[R90a] L.Reichel, *A matrix problem with application to rapid solution of integral equations*, SIAM J. Sci. Stat. Comput, **11** (1990), 263-280.

[R90b] L.Reichel, *Newton interpolation at Leja points*, BIT, 30(1990), 23 – 41.

[Ro85] V. Rokhlin, *Rapid solution of integral equations of classic potential theory*, J. Comput Phys. **60** (1985), 187-207.

[RO91] L. Reichel and G. Opfer, *Chebyshev-Vandermonde Systems*, Math. of Comp., **57**(1991) , 703-721.

[RS94] *Reed-Solomon codes and their applications*, (S.Wicker and V.Bhargava, eds.), IEEE Press, New York, 1994.

[S59] S. Schechter, *On inversion of certain matrices*, Mathematical Tables and Other Aids to Computation, vol. XIII, **66**, April 1959, 73 - 77.

[S71] G.Stewart, *Error analysis of the algorithm for shifting the zeros of a polynomial by synthetic division*, Math. of Comp., **19**(1971), 135-139.

[S79] R.Skeel, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach., **26** (1979), 494-526.

[SB80] J.Stoer and R.Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, 1980.

[ST90] L.Sakhnovich and I.Tydnik, *Effective solution of the sh-Gordon equation*, Dokl. Akad. Nauk Ukr. SSR, **Ser A 9** (1990), 20-25.

[T66] J. Traub, *Associated polynomials and uniform methods for the solution of linear problems*, SIAM Review, 8, No. 3 (1966), 277 – 301.

[T86] M.Trummer, *An efficient implementation of conformal mapping method using the Szegö kernel*, SIAM J. Numer. Anal., **23** (1986), 853-872.

[TG81] W.Tang and G.Golub, *The block decomposition of a Vandermonde matrix and its aplications*, BIT, **21** (1981), 505-517.

[Ty94] E.Tyrtyshnikov, *How bad are Hankel matrices*, Numer. Math., **67**, No. 2 (1994), 261 – 269.

[V93] J. M. Varah, *Errors and Perturbations in Vandermonde Systems*, IMA Journal of Numer. Anal., **13** (1993), 1-12.

[W35] J.L.Walsh, *Interpolation and Aproximation by Rational Functions in the Complex Domain*, Amer. Math. Soc., Newe York City, 1935.

[W68] J.Wilkinson, *A priori error analysis of algebraic processes*, Proc. Intern. Congr. Math. (Moskow 1966), pp. 629-639, Moskow, Mir 1968.

[W71] J.Wilkinson, *Modern error analysis*, SIAM Review, **14**(1971), 548-568.