Scientific Research

Fast and Numerically Stable Approximate Solution of Trummer's Problem

Mohammad M. Tabanjeh

Department of Mathematics and Computer Science, Virginia State University, Petersburg, USA
Email: mtabanjeh@vsu.edu

Abstract

Trummer's problem is the problem of multiplication of an $n \times n$ Cauchy matrix C by a vector. It serves as the basis for the solution of several problems in scientific computing and engineering [1]. The straightforward algorithm solves Trummer's problem in $O(n^2)$ flops. The fast algorithm solves the problem in $O(n\log^2 n)$ flops [2] but has poor numerical stability. The algorithm we discuss here in this paper is the celebrated multipoint algorithm [3] which has been studied by Pan *et al*. The algorithm approximates the solution in $O(n\log n)$ flops in terms of n but its cost estimate depends on the bound of the approximation error and also depends on the correlation between the entries of the pair of n-dimensional vectors defining the input matrix C.

Keywords

Cauchy Matrix, Mulipoint Algorithm, Structure Matrices, Displacement Operators

1. Introduction

Computations with dense structured matrices have many applications in sciences, communications and engineering. The structure enables dramatic acceleration of the computations and major decrease in memory space but sometimes leads to numerical stability problems. The best well-known classes of structured matrices are *Toeplitz*, *Hankel*, *Cauchy* and *Vandermonde* matrices.

The computations with such matrices are widely applied in the areas of algebraic coding, control, signal processing, solution of partial differential equations and algebraic computing. For example, Toeplitz matrices arise in some major signal processing computations and the problem of multiplying Vandermonde matrix by a vector is equivalent to polynomial evaluation, whereas solving a Vandermonde system is equivalent to polynomial interpolation. Moreover, Cauchy matrices appear in the study of integral equations and conformal mappings. The complexity of computations with $n \times n$ dense structured matrices dramatically decreases in compari-

son with the general $n \times n$ matrices, that is, from the order of n^2 words of storage space and n^α arithmetic operations (ops) with $2.37 < \alpha \le 3$ in the best algorithms, to $O(n)$ words of storage space and $O(n\log n)$ ops (see **Table 1** below for more details).

2. Some Basic Definitions

Throughout this paper, we use the following notations; $\mathbb{N}$ denotes the set of nonnegative integers, $\mathbb{R}^+$ is the set of positive real numbers, $\mathbf{Z}^+$ denotes the set of positive integers, and $\mathbf{R}$ denotes the set of real numbers.

Definition 2.1. A matrix $\boldsymbol{T} = \left[t_{i,j} \right]_{i,j=0}^{n-1}$ is a Toeplitz matrix if $t_{i,j} = t_{i+1,j+1}$ for every pair of its entries $t_{i,j}$ and $t_{i+1,j+1}$. A matrix $\boldsymbol{H} = \left[h_{i,j} \right]_{i,j=0}^{n-1}$ is a Hankel matrix if $h_{i,j} = h_{i-1,j+1}$ for every pair of its entries $h_{i,j}$ and $h_{i-1,j+1}$.

Definition 2.2. For a given vector $\boldsymbol{v} = \left(v_i \right)_{i=0}^{n-1}$, the matrix $\boldsymbol{V} = V(\boldsymbol{v})$ of the form $\boldsymbol{V} = \left[v_i^j \right]_{i,j=0}^{n-1}$ is called a Vandermonde matrix.

Definition 2.3. Given two vectors $\boldsymbol{s}$ and $\boldsymbol{t}$ such that $s_i \ne t_j$ for all i and j, the $n \times n$ matrix $\boldsymbol{C} = C(\boldsymbol{s},\boldsymbol{t})$ is a Cauchy (generalized Hilbert) where

$$C(\boldsymbol{s},\boldsymbol{t}) = \left[\frac{1}{s_i - t_j} \right]_{i,j=0}^{n-1}.$$

For more details regarding the four classes of structured matrices, see **Table 2** below.

Table 1. Parameter and flops count.

Matrices A size $n \times n$	Number of parameters for A	Number of flops required for Multiplication by a vector
General	n^2	$2n^2 - 2n$
Toeplitz	$2n-1$	$O(n\log n)$
Hankel	$2n-1$	$O(n\log n)$
Vandermonde	n	$O(n\log^2 n)$
Cauchy	$2n$	$O(n\log^2 n)$

Table 2. General definition of the four classes of structured matrices.

Toeplitz matrices, $\boldsymbol{T} = \left[t_{i-j} \right]_{i,j=0}^{n-1}$

$$\begin{bmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}$$

Hankel matrices, $\boldsymbol{H} = \left[h_{i+j} \right]_{i,j=0}^{n-1}$

$$\begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \iddots & h_n \\ \vdots & \iddots & \iddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix}$$

Vandermonde matrices, $\boldsymbol{V} = \left[v_i^j \right]_{i,j=0}^{n-1}$

$$\begin{bmatrix} 1 & v_0 & \cdots & v_0^{n-1} \\ 1 & v_1 & \cdots & v_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & v_{n-1} & \cdots & v_{n-1}^{n-1} \end{bmatrix}$$

Cauchy matrices, $\boldsymbol{C} = \left[\dfrac{1}{x_i - y_j} \right]_{i,j=0}^{n-1}$

$$\begin{bmatrix} \dfrac{1}{x_0 - y_0} & \cdots & \dfrac{1}{x_0 - y_{n-1}} \\ \dfrac{1}{x_1 - y_0} & \cdots & \dfrac{1}{x_1 - y_{n-1}} \\ \vdots & \ddots & \vdots \\ \dfrac{1}{x_{n-1} - y_0} & \cdots & \dfrac{1}{x_{n-1} - y_{n-1}} \end{bmatrix}$$

Remark 2.1. It is quite easy to verify that $\boldsymbol{TJ}$ and $\boldsymbol{JT}$ are Hankel matrices if $\boldsymbol{T}$ is a Toeplitz matrix, and $\boldsymbol{HJ}$ and $\boldsymbol{JH}$ are Toeplitz matrices if $\boldsymbol{H}$ is a Hankel matrix where $\boldsymbol{J}$ is the following reflection matrix,

$$\boldsymbol{J} = \begin{pmatrix} 0 & & 1 \\ & \mathinner{\mkern1mu\raise1pt\hbox{.}\mkern2mu\raise4pt\hbox{.}\mkern2mu\raise7pt\hbox{.}} & \\ 1 & & 0 \end{pmatrix}.$$

3. The Displacement Operators of Dense Structured Matrices

The concept of displacement operators and displacement rank which was introduced by T. Kailath, S. Y. Kung, and M. Morf in 1979 and studied by Pan, Bini, and other authors is one of the powerful tools for studying and dealing with matrices that have structure. The displacement rank approach, when it was initially introduced, was intended for more restricted use [4], namely, to measure how "close" to Toeplitz a given matrix is. Then the idea turned out to be even more powerful, thus it was developed, generalized and extended to other structured matrices. In this section, we consider the most general and modern interpretation of the displacement of a matrix.

The main idea is, for a given structured matrix $\boldsymbol{A}$, we need to find an operator L that transforms the matrix into a low rank matrix $L(\boldsymbol{A})$ such that one can easily recover $\boldsymbol{A}$ from its image $L(\boldsymbol{A})$ and operate with low rank matrices instead. Such operators that shift and scale the entries of the structured matrices turn out to be appropriate tools for introducing and defining the matrices of *Toeplitz-like*, *Hankel-like*, *Vandermonde-like*, and *Cauchy-like* types [5].

Definition 3.1. For any fixed field $\mathbb{F}$ such as the complex field $\mathbb{C}$ and a fixed pair $\{\boldsymbol{M}, \boldsymbol{N}\}$ of operator matrices, we define the linear displacement operators $L : \mathbb{F}^{m \times n} \to \mathbb{F}^{m \times n}$ of Sylvester type,

$$L(\boldsymbol{A}) = \nabla_{\boldsymbol{M}, \boldsymbol{N}}(\boldsymbol{A}) = \boldsymbol{MA} - \boldsymbol{AN}$$

and Stein type,

$$L(\boldsymbol{A}) = \nabla_{\boldsymbol{M}, \boldsymbol{N}}(\boldsymbol{A}) = \boldsymbol{A} - \boldsymbol{MAN}.$$

The image $L(\boldsymbol{A})$ of the operator L is called the *displacement* of the matrix $\boldsymbol{A}$. The operators of Sylvester and Stein types can be transformed easily into one another if at least one of the two associated operator matrices is non-singular. The following theorem explains this fact.

Theorem 3.1. $\nabla_{\boldsymbol{M}, \boldsymbol{N}} = \boldsymbol{M}\nabla_{\boldsymbol{M}^{-1}, \boldsymbol{N}}$ if the operator matrix $\boldsymbol{M}$ is non-singular, and $\nabla_{\boldsymbol{M}, \boldsymbol{N}} = -\nabla_{\boldsymbol{M}, \boldsymbol{N}^{-1}}$ if the operator matrix $\boldsymbol{N}$ is non-singular.

Proof:

$$\nabla_{\boldsymbol{M}, \boldsymbol{N}}(\boldsymbol{A}) = \boldsymbol{MA} - \boldsymbol{AN} = \boldsymbol{MA} - \boldsymbol{MM}^{-1}\boldsymbol{AN} = \boldsymbol{M}\left(\boldsymbol{A} - \boldsymbol{M}^{-1}\boldsymbol{AN}\right) = \boldsymbol{M}\Delta_{\boldsymbol{M}^{-1}, \boldsymbol{N}}(\boldsymbol{A}),$$

$$\nabla_{\boldsymbol{M}, \boldsymbol{N}}(\boldsymbol{A}) = \boldsymbol{MA} - \boldsymbol{AN} = \left(\boldsymbol{MAN}^{-1}\boldsymbol{N} - \boldsymbol{AN}\right) = \left(\boldsymbol{MAN}^{-1} - \boldsymbol{A}\right)\boldsymbol{N} = -\left(\boldsymbol{A} - \boldsymbol{MAN}^{-1}\right)\boldsymbol{N} = -\Delta_{\boldsymbol{M}, \boldsymbol{N}^{-1}}(\boldsymbol{A})\boldsymbol{N}.$$

The operator matrices that we will be using are the matrices $\boldsymbol{Z}_f$, $\boldsymbol{Z}_f^{\mathrm{T}}$, and $D(\boldsymbol{d})$ where $\boldsymbol{Z}_f$, is the unit f-circulant matrix,

$$\boldsymbol{Z}_f = \begin{bmatrix} 0 & \cdots & 0 & f \\ 1 & \ddots & \vdots & \vdots \\ \vdots & \cdots & & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix}$$

f is any scalar, $\boldsymbol{Z}_f^{\mathrm{T}}$ is the transpose of $\boldsymbol{Z}_f$, and $D(\boldsymbol{d})$ is a diagonal matrix with diagonal entries $d_0, \cdots, d_{n-1}$,

$$D(\boldsymbol{d}) = \mathrm{diag}\left(d_i\right)_{i=0}^{n-1} = \begin{pmatrix} d_0 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & d_{n-1} \end{pmatrix}.$$

We may use the operator matrices $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_0$ in the case of Toeplitz matrices, $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_1^{\mathrm{T}}$ in the case of Han-

kel matrices, Z_0 and $D(v)$ in the case of Vandermonde matrices, and $D(x)$ and $D(y)$ in the case of Cauchy matrices. However, there are other choices of operator matrices that can transform these matrices to low rank.

4. The Correlation of Structured Matrices to Polynomials

The product

$$V(x)p = v \tag{1}$$

represents the vector $v = (v_i)$ of the values of the polynomial $p(x) = \sum_j p_j x^j$ on a node set $\{x_i\}$.
If $x = w = (w_n^i)_{i=0}^{n-1}$ is the vector of the nth roots of unity,

$$w_n = e^{\frac{2\pi}{n}i} = \cos\left(\frac{2\pi}{n}\right) + i\sin\left(\frac{2\pi}{n}\right), \text{ where } i = \sqrt{-1}, \text{ and } w_n^n = 1,$$

then the matrix $F = \dfrac{V(w)}{\sqrt{n}}$ and multipoint evaluation turns into discrete Fourier transform which takes only

$O(n\log n)$ ops and allows numerically stable implementation according to [6]. If we express $V(x)$ in Equation (1) via Cauchy matrices we will get

$$V(x) = \frac{1}{\sqrt{n}} \operatorname{dia}\left(1 - x_i^n\right)_{i=0}^{n-1} C(x,w) \operatorname{diag}\left(w_i\right)_{i=0}^{n-1} F. \tag{2}$$

Here, $D(h) = \operatorname{diag}(h_i)_{i=0}^{n-1}$ for $h = (h_i)_{i=0}^{n-1}$ denotes $n \times n$ diagonal matrix with diagonal entries $h_0, \cdots, h_{n-1}$.

Note that the numerical stability is very important in approximation algorithm for multipoint polynomial evaluation. It relies on expressing $V(x)$ in terms of Cauchy matrices as in Equation (2). Clearly in Equation (2), the product $V(x)$ by a vector has been reduced to ones with Cauchy, which brings us to Trummer's problem, that is, the problem of multiplication of Cauchy matrix by a vector. Its solution by multipoint algorithm ([6], pp. 261-262) leads to multipoint polynomial evaluation based on Equation (2) which is fast in terms of ops and numerically stable as it was proved by Pan.

We may vary the vector x by linearly mapping it to the vector $y = ax + be$ where we can take $e = (1)_{i=0}^{n-1}$ and $0 \neq a$ and b are any scalars.

5. New Transformation of Cauchy Matrices

As we mentioned earlier, Trummer's problem is the problem of multiplication of an $n \times n$ Cauchy matrix C by a vector which is the basis for the solution of many important problems of scientific computing and engineering. The straightforward algorithm solves Trummer's problem in $O(n^2)$ flops. The fast algorithm solves the problem in $O(n\log^2 n)$ flops but has poor numerical stability.

The algorithm we presenting in this paper approximates the solution in $O(n\log n)$ flops in terms of n. However, its cost estimate depends on the bound of the approximation error and on the correlation between the entries of the pair of n-dimensional vectors defining the input matrix C. This algorithm is numerically stable as we will see throughout this section and the next section.

The main goal in this paper is to enrich the power of the multipoint algorithm by introducing and proving some new expressions for Cauchy matrix via other Cauchy matrices [7], which we may vary by changing one of their basis vectors. Under a certain choice of such a vector, the solution of Trummer's problem will be simplified; thus, the power of the multipoint algorithm can be developed as we will see in the next section.

Therefore, we will achieve our goal by using a simple transformation of the useful basic formula of [8], and the resulting expressions for C will give us further algorithmic opportunities.

Definition 5.1. For a pair of n-dimensional vectors $a = (a_i)_{i=0}^{n-1}$, $b = (b_j)_{j=0}^{n-1}$ let $C(a,b) = \left(1/(a_i - b_i)\right)_{i,j=0}^{n-1}$, $V(a) = \left(a_i^j\right)_{i,j=0}^{n-1}$, $H(a) = \left(h_{i,j}\right)_{i,j=0}^{n-1}$, $h_{i,j} = a_{i,j}$ for $i + j \leq n-1$, $h_{i,j} = 0$ for $i + j \geq n-1$, denote the associated $n \times n$ Cauchy, Vandermonde, and triangular Hankel matrices, respectively. For a vector $a = (a_i)_{i=0}^{n-1}$ with $a_i \neq a_j$ for $i \neq j$ a Cauchy degenerate matrix $C(a)$ has the diagonal entries zeros and the $(i, j)^{th}$ entry

$1/(a_i - a_j)$ for $i \neq j$. Furthermore, $p_b(x)$ denote the polynomial $p_b(x) = \prod_{j=0}^{n-1}(x - b_j)$ and its derivative $p_b'(x) = \sum_{i=0}^{n-1} \prod_{j=0(j \neq i)}^{n-1}(x - b_j)$. Lastly,

$$D(a,b) = \text{diag}\left(p_b(a_i)\right)_{i=0}^{n-1} = \text{diag}\left(\prod_{j=0}^{n-1}(a_i - b_j)\right)_{i=0}^{n-1}$$

and

$$D'(b) = \text{diag}\left(p_b'(b_i)\right)_{j=0}^{n-1} = \text{diag}\left(\sum_{i=0}^{n-1}\prod_{j=0}^{n-1}(b_i - b_j)\right)$$

denote a pair of $n \times n$ diagonal matrices, defined by the vectors a and b.

Theorem 5.1. (See [8]) Let $c_i \neq d_j$, $i, j = 0, 1, \cdots, n-1$. Then

$$C(c,d) = D(c,d)^{-1} V(c) H(d) V(d)^{\mathrm{T}} \tag{3}$$

$$C(c,d) = D(c,d)^{-1} V(c) V(d)^{-1} D'(d). \tag{4}$$

The main idea of the transformation of the basic vectors defining the problem is taken from [9], where this idea was used for multipoint polynomial evaluation and interpolation.

Definition 5.2. Trummer's problem is the problem of computing the vector $C(a,b)v$ for three given vectors $a = (a_i)_{i=0}^{n-1}$, $b = (b_j)_{j=0}^{n-1}$ and $v = (v_j)_{j=0}^{n-1}$ where $a_i \neq b_j$ for all pairs i, j. Trummer's degenerate problem is the problem of computing the vector $C(a)v$ for two given vectors $a = (a_i)_{i=0}^{n-1}$ and $v = (v_j)_{j=0}^{n-1}$ where $a_i \neq a_j$ for $i \neq j$.

Definition 5.3. $w_k = \exp(2\pi i/k)$, where $i = \sqrt{-1}$, is a primitive k^{th} root of 1, $w_k^k = 1$, $w_k^l \neq 1$ for $l = 1, \cdots, k-1$.

Lemma 5.1. $\sum_{l=0}^{k-1}\left(w_k^i\right)^l = 0$ for $i = 1, \cdots, k-1$.

Approximate solution of Trummer's degenerate problem can be reduced to Trummer's problem due to the next simple result.

Lemma 5.2. $C(c) = 1/h \sum_{i=0}^{h-1} C(c, c + \delta w_h^i e) + O(\delta^h)$ as $\delta \to 0$, where $e = (1)_{j=0}^{n-1}$ is the vector filled with the values one and δ is a scalar parameter.

Proof: $\sum_{g=0}^{h-1} 1/(c_i - c_j - \delta w_h^g) = 1/(c_i - c_j) \sum_{l=0}^{\infty} \sum_{g=0}^{h-1}\left(\delta w_h^g/c_i - c_j\right)^l = h/(c_i - c_j)\left(1 + O(\delta^h)\right)$ due to Lemma 5.1.

6. Transformations of Cauchy Matrices and Trummer's Problem

Theorem 6.1. For a triple of n-dimensional vector $b = (b_i)_{i=0}^{n-1}$, $c = (c_j)_{j=0}^{n-1}$, $d = (d_k)_{k=0}^{n-1}$ where $b_i \neq c_j$, $c_j \neq d_k$, $d_k \neq b_i$ for $i, j, k = 0, \cdots, n-1$ we have the following matrix equations:

$$C(c,d) = D(c,d)^{-1} V(c) V(b)^{-1} D(b,d) C(b,d) \tag{5}$$

$$C(c,d) = D(c,d)^{-1} D(c,b) C(c,b) D'(b)^{-1} D(b,d) C(b,d) \tag{6}$$

$$C(c,d) = C(c,b) D(b,c) V(b)^{-\mathrm{T}} V(d)^{\mathrm{T}} D(d,c)^{-1} \tag{7}$$

$$C(c,d) = -C(c,b) D(b,c) D'(b)^{-1} C(b,d) D(d,b) D(d,c)^{-1}. \tag{8}$$

Proof Theorem 6.1:

1) Proof of Equation (5):

From Equation (3), we obtain $C(b,d)^{-1} = V(d)^{-\mathrm{T}} H(d)^{-1} V(b)^{-1} V(b)^{-1} D(b,d)$. This is done by taking the inverse of Equation (3) and replacing the vectors c, d by b, d. Then substitute the equation

$$C(b,d)^{-1} = V(d)^{-\mathrm{T}} H(d)^{-1} V(b)^{-1} V(b)^{-1} D(b,d)$$

and Equation (3) for $C(c,d)$ into the following matrix identity

$$C(c,d) = C(c,d) C(b,d)^{-1} C(b,d).$$

This gives:

$$C(c,d) = C(c,d)\left[V(d)^{-T} H(d)^{-1} V(b)^{-1} D(b,d)\right] C(b,d)$$

$$= D(c,d)^{-1} V(c) H(d) V(d)^{T} \left[V(d)^{-T} H(d)^{-1} V(b)^{-1} D(b,d)\right] C(b,d)$$

$$= D(c,d)^{-1} V(c) H(d) \left[V(d)^{T} V(d)^{-T}\right] H(d)^{-1} V(b)^{-1} D(b,d) C(b,d)$$

$$= D(c,d)^{-1} V(c) \left[H(d) H(d)^{-1}\right] V(b)^{-1} D(b,d) C(b,d)$$

$$= D(c,d)^{-1} V(c) V(b)^{-1} D(b,d) C(b,d).$$

Clearly, the last one is just Equation (5).

2) Proof of Equation (6):

From Equation (4); $C(c,d) = D(c,d)^{-1} V(c) V(d)^{-1} D'(d)$ replace the vector d by b, then we will get:

$$C(c,b) = D(c,b)^{-1} V(c) V(b)^{-1} D'(b).$$

Then solve for $V(c) V(b)^{-1}$;

$$V(c) V(b)^{-1} = D(c,b) C(c,b) D'(b)^{-1}.$$

Now substitute the last expression into Equation (5) and obtain the following:

$$C(c,d) = D(c,d)^{-1} \left[D(c,b) C(c,b) D'(b)^{-1}\right] D(b,d) C(b,d)$$

$$= D(c,d)^{-1} D(c,b) C(c,b) D'(b)^{-1} D(b,d) C(b,d)$$

which is obviously Equation (6).

3) Proof of Equation (7):

From Equation (4); $C(c,d) = D(c,d)^{-1} V(c) V(d)^{-1} D'(d)$ first solve for $V(d)^{-1} D'(d)$ to get:

$$V(d)^{-1} D'(d) = V(c)^{-1} D(c,d) C(c,d),$$

replace $c \leftrightarrow d$ to get:

$$V(c)^{-1} D'(c) = V(d)^{-1} D(d,c) C(d,c).$$

Now, replace the vector d by b and obtain

$$V(c)^{-1} D'(c) = V(b)^{-1} D(b,c) C(b,c). \tag{9}$$

Since $C(c,d) = -C(d,c)^{T}$ we have also $-C(c,d) = C(d,c)^{T}$.

Start with Equation (4) which is $C(c,d) = D(c,d)^{-1} V(c) V(d)^{-1} D'(d)$ and replace the vector $c \leftrightarrow d$ to obtain $C(d,c) = D(d,c)^{-1} V(d) V(c) D'(c)$, then use the equation $-C(c,d) = C(d,c)^{T}$ to get:

$$-C(c,d) = C(d,c)^{T} = \left[D(d,c)^{-1} V(d)\left[V(c) D'(c)\right]\right]^{T}$$

$$= \left[D(d,c)^{-1} V(d)\left[V^{-1}(b) D(b,c) C(b,c)\right]\right]^{T} \text{ (from Equation (9))}$$

$$= \left[D(d,c)^{-1} V(d) V^{-1}(b) D(b,c) C(b,c)\right]^{T}$$

$$= C(b,c)^{T} D(b,c) V(b)^{-T} V^{T}(d) D(d,c)^{-1}.$$

Now replace $C(b,c)^{T}$ in the last equation by $-C(c,b)$ (that is, use the identity $-C(c,b) = C(b,c)^{T}$)

$$-C(c,d) = -C(b,c) D(b,c) V(b)^{-T} V(d)^{T} D(d,c)^{-1}.$$

This implies that $C(c,d) = C(c,b) D(b,c) V(b)^{-T} V(d)^{T} D(d,c)^{-1}$ which is Equation (7).

4) Proof of Equation (8):

From Equation (4): $C(c,d) = D(c,d)^{-1} V(c) V(d)^{-1} D'(d)$ solve for $V(c) V(d)^{-1}$ and obtain

$$D(c,d)C(c,d)D'(d)^{-1} = V(c)V(d)^{-1}.$$

Expand Equation (4) $C(c,d)D(c,d)D'(d)^{-1} = V(c)V(d)^{-1}$, and change $c \leftrightarrow d$ and $d \leftrightarrow b$ to get the equation: $D(d,b)C(d,b)D'(b)^{-1} = V(d)V(b)^{-1}$ and take the transpose of both side of the last equation to get:

$$V(b)^{-T}V(d)^{T} = D'(b)^{-1}C(d,b)^{T}D(d,b). \tag{10}$$

Substitute Equation (10) and the matrix equation $C(d,b)^{T} = -C(b,d)$ into Equation (7):

$$C(c,d) = C(c,b)D(b,c)\left[V(b)^{-T}V(d)^{T}\right]D(d,c)^{-1}$$

$$C(c,d) = C(c,b)D(b,c)\left[D'(b)^{-1}C(d,b)^{T}D(d,b)\right]D(d,c)^{-1} \text{ (this is from Equation (10))}$$

$$= -C(c,b)D(b,c)D'(b)^{-1}C(b,d)D(d,b)D(d,c)^{-1} \text{ which is Equation (8).}$$

7. Approximate Stable Solutions of Trummer's Problems

The algorithm we are studying and presenting in this section depends on the multipoint algorithm which approximates the solution of Trummer's problem in $O(n)$ ops in terms of n, and works efficiently for a large classes of input vectors but sometimes has problems with some of the input vectors, especially if the ratio of the input vectors is close to 1.

Recall, the power series expansion: $\sum_{i=0}^{\infty} x^i = 1 + x + x^2 + x^3 + \cdots$ this series converges whenever $|x| < 1$ and has a sum $\dfrac{1}{1-x}$.

The basis for the algorithm is the following expressions:

$$\frac{1}{s_i - t_j} = \frac{1}{s_i\left(1 - \left(\dfrac{t_j}{s_i}\right)\right)} = \frac{1}{s_i}\sum_{k=0}^{\infty}\left(\frac{t_j}{s_i}\right)^k \tag{11}$$

where $s = (s_i)_{i=0}^{n-1}$ and $t = (t_i)_{i=0}^{n-1}$. Clearly this series converges whenever $\left|\dfrac{t_j}{s_i}\right| < 1$. Now for large M the expression $\dfrac{1}{s_i}\sum_{k=0}^{M}\left(\dfrac{t_i}{s_i}\right)^k$ approximates $\dfrac{1}{s_i - t_j}$. On the other hand $\dfrac{1}{s_i - t_j}$ can be also written as

$$\frac{1}{s_i - t_j} = \frac{1}{-t_j\left(1 - \left(\dfrac{s_i}{t_j}\right)\right)} = -\frac{1}{t_j}\sum_{k=0}^{\infty}\left(\frac{s_i}{t_j}\right)^k. \tag{12}$$

Once again for large M the expression $-\dfrac{1}{t_j}\sum_{k=0}^{M}\left(\dfrac{s_i}{t_j}\right)$ approximate $\dfrac{1}{s_i - t_j}$.

The product of Cauchy matrix by a vector is $C(s,t)v = \sum_{j=0}^{n-1}\left(\dfrac{v_j}{s_i - t_j}\right)_{i=0}^{n-1}$. This is just Trummer's problem. If we simplify this expression, we get the following approximations:

$$\frac{v_j}{s_i - t_j} \simeq \sum_{j=0}^{n-1} -\frac{1}{t_j}\sum_{k=0}^{M}\left(\frac{s_i}{t_j}\right)^k v_j = -\sum_{j=0}^{n-1}\left(\frac{v_j}{t_j}\right)\sum_{k=0}^{M}\left(\frac{s_i}{t_j}\right)^k = -\sum_{j=0}^{n-1}\left(\frac{v_j}{t_j}\right)\sum_{k=0}^{M}\frac{s_i^k}{t_j^k} = -\sum_{j=0}^{n-1}\frac{v_j}{t_j^{k+1}}\sum_{k=0}^{M}s_i^k = \sum_{k=0}^{M}z_k s_i^k$$

where $z_k = -\sum_{j=0}^{n-1} \dfrac{v_j}{t_j^{k+1}}$.

For any $n \times n$ Cauchy matrix $C(s,t)$, the approximation requires $O(nM)$ ops for all i and it is numerically stable.

If either of the ratios $\left|\dfrac{s_i}{t_j}\right|$ or $\left|\dfrac{t_j}{s_i}\right|$ is small, then the approximate error will be small for large M. However, there will be a problem whenever one of the ratios is close to 1, in this case, the error will be large.

8. Discussions and Conclusions

Recall the following two formulas from Section 5:

$$C(s,t) = D(s,t)^{-1} D(s,q) C(s,q) D'(q)^{-1} D(q,t) C(q,t), \tag{13}$$

$$C(s,t) = -C(s,q) D(q,s) D'(q)^{-1} C(q,t) D(t,q) D(t,s)^{-1}. \tag{14}$$

Equations (13) and (14) are Vandermonde-free and Hankel-free, but they enable us to transform the basis vectors s and t for $C(s,t)$ into the two pairs of basis vectors s, q and q, t for any choice of the vector $q = (q_j)$, $q_j \neq s_i$, $q_j \neq t_k$, $i, j, k = 0, \cdots, n-1$. Then Trummer's problem is reduced to the evaluation of the diagonal matrices $D'(q)^{-1}$, $D(u,v)$ and/or $D(u,v)^{-1}$ for (s,t), (q,t), (s,q), (q,s), (t,q) and/or (t,s) and also reduced to recursive multiplication of the above matrices and $C(q,t)$ and $C(s,q)$ by vectors.

To compute the matrices $D'(v)$, $D(u,v)$ and $D(u,v)^{-1}$ for given (u,v) in general, we first compute the coefficients of the polynomial $p_u(x) = \prod_{j=0}^{n-1}(x - v_j)$ and then $p_v(u_i)$.

And $p_u'(v_i)$, $i = 0, \cdots, n-1$. We compute the coefficients by simply pairwise multiply the linear factors $x - v_j$ first and then, recursively, the computed products. The computation is numerically stable and uses $O(n\log^2 n)$ ops. Multipoint polynomial evaluation can be computed in $O(n\log^2 n)$ arithmetic operations (ops), but it is not numerically stable; therefore the fast and numerically stable approximation techniques of [10] can be used instead. If we choose any vector $q = (q_i)_{i=0}^{n-1}$, we will simplify the evaluation of the matrices

$$D(u,v), \; D(u,v)^{-1} \text{ and } D'(q) \text{ where } u = q \text{ or } v = q.$$

For example, if

$$q_i = aw_n^i, \quad i = 0,1,\cdots,n-1 \tag{15}$$

is the scaled n^{th} roots of unity for a scalar a and $w_n = \exp(2\pi i/n)$, where $i = \sqrt{-1}$. Then

$$p_q(x) = \prod_{i=0}^{n-1}(x - aw_n^i) = x^n - a^n, \quad p_q'(x) = nx^{n-1}$$

and the matrices $D(u,q)$ and $D(q)$ can be immediately evaluated in $O(n\log n)$ flops. In addition, any polynomial $p(x)$ of degree n can be evaluated at the scaled n^{th} roots of 1 in $O(n\log n)$ ops by means of Fast Fourier Transform (FFT). Trummer's problem is the multiplication of $C(q,t)$ by a vector or $C(s,q)$ by a vector. Its solution can be simplified under appropriate choice of the vector q. One way to do it is to restrict q to the above choice in Equation (15). Even with this particular choice, yet the scalar a allows faster convergence of the power series of the Multipole Algorithm presented in Section 7. This can be extended to the Equations (5) and (7). On the other hand, one can linearly map the vector q into $y = aq + be$, where $e = (1)_{j=0}^{n-1}$, and $0 \neq a$ and b are any scalars. In addition, the computations of the diagonal matrices will be simplified if our choice of the vector q is the scaled nth root of unity.

Remark 8.1. Trummer's problem frequently arises for Cauchy degenerate matrices that are defined as follows: $C(s) = (c_{i,j})$, $c_{i,i} = 0$, $c_{i,j} = \dfrac{1}{s_i - s_j}$ for all pairs of distinct i and j.

We have

$$C(s) = \frac{1}{h} \sum_{g=0}^{h-1} C(s, s + \delta w_h^g e) + O(\delta^h) \text{ as } \delta \to 0$$

where $\boldsymbol{e} = (1)_{j=0}^{n-1}$, $\boldsymbol{s} = (s_i)$, δ is a scalar parameter. Hence,

$$\sum_{g=0}^{h-1} \frac{1}{s_i - s_j - \delta w_h^g} = \frac{1}{s_i - s_j} \sum_{l=0}^{\infty} \sum_{g=0}^{h-1} \left(\frac{\delta w_h^g}{s_i - s_j} \right)^l = \frac{h}{s_i - s_j} \left(1 + O\left(\delta^h\right) \right)$$

because $\sum_{l=0}^{n-1} w_n^{gl} = 0$ for $g = 1, \cdots, n-1$.

Acknowledgements

References

[1] Rokhlin, V. (1985) Rapid Solution of Integral Equations of Classical Potential Theory. *Journal of Computational Physics*, **60**, 187-207. http://dx.doi.org/10.1016/0021-9991(85)90002-6

[2] Gerasoulis, A. (1987) A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors. *Mathematics of Computation*, **50**, 179-188. http://dx.doi.org/10.1090/S0025-5718-1988-0917825-9

[3] Greengard, L. and Rokhlin, V. (1987) A Fast Algorithm for Practice Simulation. *Journal of Computational Physics*, **73**, 325-348. http://dx.doi.org/10.1016/0021-9991(87)90140-9

[4] Pan, V.Y., Zheng, A., Huany, X. and Yu, Y. (1995) Fast Multipoint Polynomial Evaluation and Interpolation via Computation with Structured Matrices. *Annals of Numerical Mathematics*, **4**, 483-510.

[5] Pan, V.Y. (2001) Structured Matrices and Polynomials, Unified Superfast Algorithms. Birkhäuser, Boston. http://dx.doi.org/10.1007/978-1-4612-0129-8

[6] Bini, D. and Pan, V.Y. (1994) Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms. Birkhäuser, Boston. http://dx.doi.org/10.1007/978-1-4612-0265-3

[7] Pan, V.Y., Tabanjeh, M., Chen, Z., Landowne, E. and Sadikou, A. (1998) New Transformations of Cauchy Matrices and Trummer's Problem. *Computers & Mathematics with Applications*, **35**, 1-5. http://dx.doi.org/10.1016/S0898-1221(98)00091-1

[8] Fink, T., Heinig, G. and Rost, K. (1993) An Inversion Formula and Fast Algorithms for Cauchy-Vandermonde Matrices. *Linear Algebra & Its Applications*, **183**, 179-191. http://dx.doi.org/10.1016/0024-3795(93)90431-M

[9] Pan, V.Y., Landowne, E., Sadikou, A. and Tiga, O. (1993) A New Approach to Fast Polynomial Interpolation and Multipoint Evaluation. *Computers & Mathematics with Applications*, **25**, 25-30. http://dx.doi.org/10.1016/0898-1221(93)90129-J

[10] Pan, V.Y. (1995) An Algebraic Approach to Approximate Evaluation of a Polynomial on a Set of Real Points. *Advances in Computational Mathematics*, **3**, 41-58. http://dx.doi.org/10.1007/BF02431995

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.