Efficient Solution of Parameter Dependent Quasiseparable Systems and Computation of Meromorphic Matrix Functions

Paola Boito, Yuli Eidelman, Luca Gemignani

Efficient Solution of Parameter Dependent Quasiseparable Systems and Computation of Meromorphic Matrix Functions

P. Boito[a,b], Y. Eidelman[c], and L. Gemignani[d]

[a] *XLIM–DMI, UMR 7252 CNRS Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex, France. Email:* paola.boito@unilim.fr

[b] *CNRS, Université de Lyon, Laboratoire LIP (CNRS, ENS Lyon, Inria, UCBL), 46 allée d'Italie, 69364 Lyon Cedex 07, France.*

[c] *School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat-Aviv, 69978, Israel. Email:* eideyu@post.tau.ac.il

[d] *Dipartimento di Informatica, Università di Pisa, Largo Bruno Pontecorvo, 3 - 56127 Pisa, Italy. Email:* l.gemignani@di.unipi.it

Abstract

In this paper we focus on the solution of shifted quasiseparable systems and of more general parameter dependent matrix equations with quasiseparable representations. We propose an efficient algorithm exploiting the invariance of the quasiseparable structure under diagonal shifting and inversion. This algorithm is applied to compute various functions of matrices. Numerical experiments show the effectiveness of the approach.

Keywords: Quasiseparable matrices , shifted linear system, QR factorization, matrix function, matrix equation.
2010 MSC: 65F15

1 Introduction

In this paper we propose a novel method for computing the solution of shifted quasiseparable systems and of more general parameter dependent linear matrix equations with quasiseparable representations. We show that our approach has also a noticeable potential for effectively solving some large-scale algebraic problems that reduce to evaluating the action of a quasiseparable matrix function to a vector.

Quasiseparable matrices found their application in several branches of applied mathematics and engineering and, therefore, there has been major interest in developing fast algorithms for working with them in the last decade [6, 7, 17, 16]. The quasiseparable structure arises in the discretization of continuous operators due either to the local properties of the discretization schemes and/or the decaying properties of the operator or its finite approximations.

It is a celebrated fact that operations with quasiseparable matrices can be performed in linear time with respect to their size. In particular the QR factorization algorithm presented in [5] computes in linear time a QR decomposition of a quasiseparable matrix $A \in \mathbb{C}^{N\times N}$ of the form $A = V \cdot U \cdot R$, where R is upper triangular whereas U and V are banded unitary matrices and V only depends on the generators of the strictly lower triangular part of A. This implies that any shifted linear system $A + \sigma I_N$, $\sigma \in \mathbb{C}$, can also be factored as $A + \sigma I_N = V \cdot U_\sigma \cdot R_\sigma$ for suitable U_σ and R_σ.

Relying upon this fact, in this paper we design an efficient algorithm for solving a sequence of shifted quasiseparable linear systems. The invariance of the factor V can be exploited to halve the overall computational cost. Section 2 illustrates the potential of our approach using motivating examples and applications. In Section 3 we describe the novel algorithm by proving its correctness. Finally, in Section 4 we show the results of numerical experiments which confirm the effectiveness and the robustness of our method.

2 Motivating Examples

In this section we describe two motivating examples from applied fields that lead to the solution of several shifted or parameter dependent quasiseparable linear systems.

2.1 A Model Problem for Boundary Value ODEs

Consider the non-local boundary value problem in a linear finite dimensional normed space $X = \mathbb{R}^N$:

$$\frac{d\boldsymbol{v}}{dt} = A\boldsymbol{v}, \quad 0 < t < \tau, \tag{2.1}$$

$$\frac{1}{\tau}\int_0^\tau \boldsymbol{v}(t)\, dt = \boldsymbol{g} \tag{2.2}$$

where A is a linear operator in $\mathbb{R}^N$ and $\boldsymbol{g} \in \mathbb{R}^N$ is a given vector.

Under the assumption that all the numbers $\mu_k = 2\pi ik/\tau,\ k = \pm 1, \pm 2, \pm 3, \ldots$ are regular points of the operator A the problem (2.1), (2.2) has a unique

solution. Moreover this solution is given by the formula

$$\boldsymbol{v}(t) = q_t(A)\boldsymbol{g}, \quad q_t(z) = \frac{\tau z e^{zt}}{e^{z\tau} - 1}. \tag{2.3}$$

Without loss of generality one can assume that $\tau = 2\pi$.

Expanding the function $q_t(z)$ in the Fourier series of t we obtain

$$q_t(z) = 1 + \sum_{k\in\mathbb{Z}/\{0\}} \frac{ze^{ikt}}{z - ik}, \quad 0 < t < 2\pi.$$

We consider the equivalent representation with the real series given by

$$q_t(z) = 1 + 2\sum_{k=1}^{\infty} z(z\cos kt - k\sin kt)(z^2 + k^2)^{-1}, \quad 0 < t < 2\pi.$$

Using the formula

$$-\frac{k}{z^2 + k^2} = \frac{z^2}{k(k^2 + z^2)} - \frac{1}{k}$$

we find that

$$q_t(z) = 1+2\sum_{k=1}^{\infty} z(z\cos kt+\frac{1}{k}z^2\sin kt)(z^2+k^2)^{-1}-2z\sum_{k=1}^{\infty}\frac{1}{k}\sin kt, \quad 0 < t < 2\pi$$

and since

$$2\sum_{k=1}^{\infty}\frac{1}{k}\sin kt = \pi - t, \quad 0 < t < 2\pi$$

we arrive at the following formula

$$q_t(z) = 1 + 2\sum_{k=1}^{\infty}(z\cos kt + \frac{1}{k}z^2\sin kt)(z^2 + k^2)^{-1} - z(\pi - t), \quad 0 < t < 2\pi.$$

Thus we obtain the sought expansion of the solution $\boldsymbol{v}(t)$ of the problem (2.1), (2.2):

$$v(t) = g-(\pi-t)Ag+2\sum_{k=1}^{\infty}(A\cos kt+\frac{1}{k}A^2\sin kt)(A^2+k^2I_N)^{-1}Ag, \quad 0 \le t \le 2\pi. \tag{2.4}$$

Under the assumptions

$$\|(A - ikI_N)^{-1}\| \le \frac{C}{|k|}, \quad k = \pm1, \pm2, \dots \tag{2.5}$$

using the Abel transform one can check that the series in (2.4) converges uniformly in $t \in [0, 2\pi]$. Hence, by continuity we extend here the formula

(2.4) for the solution on all the segment $[0, 2\pi]$. The rate of convergence of the series in (2.4) is the same as for the series $\sum_{k=1}^{\infty} \frac{1}{k^2}$. The last may be improved using standard techniques for series acceleration but by including higher degrees of A. Indeed set

$$C_k(t) = A\cos kt + \frac{1}{k}A^2 \sin kt.$$

Using the identity

$$(A^2 + k^2 I)^{-1} = \frac{1}{k^2} I - \frac{1}{k^2} A^2 (A^2 + k^2 I)^{-1}$$

we get

$$\sum_{k=1}^{\infty} C_k(t)(A^2 + k^2 I_N)^{-1} Ag = \sum_{k=1}^{\infty} \frac{1}{k^2} C_k(t) Ag - \sum_{k=1}^{\infty} \frac{1}{k^2} C_k(t)(A^2 + k^2 I)^{-1} A^3 g.$$

The first entry here has the form

$$\sum_{k=1}^{\infty} \frac{1}{k^2} C_k(t) Ag = \sum_{k=1}^{\infty} \left(\frac{\cos kt}{k^2} A^2 g + \frac{\sin kt}{k^3} A^3 g \right).$$

Using the formulas

$$\sum_{k=1}^{\infty} \frac{\cos kt}{k^2} = \frac{\pi^2}{6} - \frac{\pi}{2} t + \frac{t^2}{4}, \quad \sum_{k=1}^{\infty} \frac{\sin kt}{k^3} = \frac{\pi^2}{6} t - \frac{\pi}{4} t^2 + \frac{t^3}{12}.$$

Thus we get

$$v(t) = V_0(t) g + V_1(t) Ag + V_2(t) A^2 g + V_3(t) A^3 g - 2 \sum_{k=1}^{\infty} \frac{1}{k^2} C_k(t)(A^2 + k^2 I)^{-1} A^3 g. \tag{2.6}$$

with

$$V_0(t) = 1,\ V_1(t) = t - \pi,\ V_2(t) = \frac{\pi^2}{3} - \pi t + \frac{t^2}{2},\ V_3(t) = \frac{\pi^2}{3} t - \frac{\pi}{2} t^2 + \frac{t^3}{6}.$$

Summing up, our proposal consists in approximating the solution $\boldsymbol{v}(t)$ of the problem (2.1), (2.2) by the finite sum

$$v_\ell(t) = g - (\pi - t) Ag + 2 \sum_{k=1}^{\ell} (A\cos kt + \frac{1}{k} A^2 \sin kt)(A^2 + k^2 I_N)^{-1} Ag, \quad 0 \le t \le 2\pi, \tag{2.7}$$

or using (2.6) by the sum

$$\hat{v}_\ell(t) = \sum_{j=0}^{3} V_j(t) A^j g$$

$$-2 \sum_{k=1}^{\ell} \frac{1}{k^2} (A\cos kt + \frac{1}{k} A^2 \sin kt)(A^2 + k^2 I_N)^{-1} A^3 g, \quad 0 \le t \le 2\pi, \tag{2.8}$$

where ℓ is suitably chosen by checking the convergence of the expansion.

The computation of $\boldsymbol{v}_\ell(t_i) \simeq \boldsymbol{v}(t_i)$, $0 \le i \le M+1$, requires the solution of a possibly large set of the shifted systems of the form

$$(A + \sigma_i I_N)\boldsymbol{x}_i = \boldsymbol{y}, \quad i = 1, \ldots, \ell. \tag{2.9}$$

The same conclusion applies to the problem of computing the function of a quasiseparable matrix whenever the function can be represented as a series of partial fractions. The classes of meromorphic functions admitting such a representation were investigated for instance in [13]. Other partial fraction approximations of certain analytic functions can be found in [11]. In the next section we describe an effective algorithm for this task.

A numerical approximation of the solution can be obtained by using the discretization of the boundary value problem on a grid and the subsequent application of the cyclic reduction approach discussed in [1]. In this approach the saving of an approximate quasiseparable structure in recursive LU-based solvers may be used as it was done in the recent papers [2, 4, 9]. The approximate quasiseparable structure of functions of quasiseparable matrices has also been investigated in [12].

2.2 Sylvester-type Matrix Equations

As a natural extension of the problem (2.9), the right-hand side $\boldsymbol{y}$ could also depend on the parameter, that is, $\boldsymbol{y} = \boldsymbol{y}(\sigma)$ and $\boldsymbol{y}_i = \boldsymbol{y}(\sigma_i)$, $i = 1, \ldots, \ell$. This situation is common in many applications such as control theory, structural dynamics and time-dependent PDEs [10]. In this case, the systems to be solved takes the form

$$AX + XD = Y, \quad A \in \mathbb{R}^{N\times N},\ D = \operatorname{diag}[\sigma_1, \ldots, \sigma_\ell],\ Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell].$$

Using the Kronecker product this matrix equation can be rewritten as a bigger linear system $\mathcal{A}\operatorname{vec}(X) = \operatorname{vec}(Y)$, where $\mathcal{A} = I_\ell \otimes A + D^T \otimes I_N \in \mathbb{R}^{N\ell\times N\ell}$, $\operatorname{vec}(X) = \left[\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_\ell^T\right]^T$, $\operatorname{vec}(Y) = \left[\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_\ell^T\right]^T$.

The extension to the case where D is replaced by a lower triangular matrix $L = (l_{i,j}) \in \mathbb{R}^{\ell\times\ell}$ is based on the backward substitution technique which amounts to solve

$$(A + l_{i,i}I_N)\boldsymbol{x}_i = \boldsymbol{y}_i - \sum_{j=i+1}^{\ell} l_{i,j}\boldsymbol{x}_j, \quad i = \ell\colon -1\colon 1. \tag{2.10}$$

Such approach has been used in different related contexts where the considered Sylvester equation is occasionally called *sparse-dense* equation [14]. Then the classical reduction proposed by Bartels and Stewart [3] makes possible to deal with a general matrix F by first computing its Schur decomposition $F = ULU^H$ and then solving $A(XU) + (XU)L = YU$. The

resulting approach is well suited especially when the size of A is large w.r.t. the number of shifts. If A is quasiseparable then (2.10) again reduces to computing a sequence of shifted systems having the same structure in the lower triangular part and the method presented in the next section can be used.

3 The basic algorithm

To solve the systems (2.9), (2.10) we rely upon the QR factorization algorithm described in [5] (see also Chapter 20 of [6]). On the first step we compute the factorization

$$A + \sigma I = V \cdot T_\sigma \tag{3.11}$$

with a unitary matrix V and a lower banded, or a block upper triangular, matrix T_σ. It turns out that the matrix V_σ does not depend on σ at all and moreover essential part of the quasiseparable generators of the matrix does not depend on σ also. So a relevant part of computations for all the values of σ may be performed only once. Thus the problem is reduced to the solution of the set of the systems

$$T_\sigma \boldsymbol{x}_\sigma = V^H \boldsymbol{y}_i, \quad \sigma = \sigma_i,\ i = 1, \ldots, \ell. \tag{3.12}$$

The inversion of every matrix T_σ as well as the solution of the corresponding linear system is basically simpler than for the original matrix. We compute the factorization

$$T_\sigma = U_\sigma R_\sigma$$

with a block upper triangular unitary matrix U_σ and upper triangular R_σ and solve the systems

$$R_\sigma \boldsymbol{x}_\sigma = U_\sigma^H V^H \boldsymbol{y}_i.$$

Thus we obtain the following algorithm. Recall that a block matrix $A = (A_{i,j})_{i,j=1}^N$, $A_{i,j} \in \mathbb{R}^{m_i \times m_j}$ is said to have lower quasiseparable generators $p(i) \in \mathbb{R}^{m_i \times r_{i-1}^L}$ $(i = 2, \ldots, N)$, $q(j) \in \mathbb{R}^{r_j^L \times m_j}$ $(j = 1, \ldots, N-1)$, $a(k) \in \mathbb{R}^{r_k^L \times r_{k-1}^L}$ $(k = 2, \ldots, N-1)$ of orders r_k^L $(k = 1, \ldots, N-1)$ and upper quasiseparable generators $g(i) \in \mathbb{R}^{m_i \times r_i^U} (i = 1, \ldots, N-1)$, $h(j) \in \mathbb{R}^{r_{j-1}^U \times m_j}$ $(j = 2, \ldots, N)$, $b(k) \in \mathbb{R}^{r_{k-1}^U \times r_k^U}; (k = 2, \ldots, N-1)$ of orders r_k^U $(k = 1, \ldots, N-1)$ if

$$A_{i,j} = \begin{cases} p(i) a_{i,j}^{>} q(j) \text{ if } 1 \le j < i \le N; \\ g(i) b_{i,j}^{<} h(j) \text{ if } 1 \le i < j \le N \end{cases}$$

where $a_{i,j}^{>} = a(i-1) \cdots a(j+1)$ for $i > j+1$ and $a_{j+1,j} = I_{r_j^L}$, and, similarly, $b_{i,j}^{<} = b(i+1) \cdots b(j-1)$ for $j > i+1$ and $b_{i,i+1} = I_{r_i^U}$.

Theorem 3.1. *Let $A = (A_{i,j})_{i,j=1}^{N}$ be a block matrix with entries of sizes $m_i \times m_j$, lower quasiseparable generators $p(i)$ $(i = 2, \ldots, N)$, $q(j)$ $(j = 1, \ldots, N-1)$, $a(k)$ $(k = 2, \ldots, N-1)$ of orders r_k^L $(k = 1, \ldots, N-1)$, upper quasiseparable generators $g(i)$ $(i = 1, \ldots, N-1)$, $h(j)$ $(j = 2, \ldots, N)$, $b(k)$ $(k = 2, \ldots, N-1)$ of orders r_k^U $(k = 1, \ldots, N-1)$ and diagonal entries $d(k)$ $(k = 1, \ldots, N)$. Let $\boldsymbol{y}_i = (y_i(k))_{k=1}^{N}$ be a vector column with m_k- dimensional coordinates $y(k)$ and let σ_i, $i = 1, \ldots, \ell$ be complex numbers. Then the set of solutions $\boldsymbol{x}^{(i)} = \boldsymbol{x}_{\sigma_i}$, $i = 1, \ldots, \ell$ of the systems* (3.12) *is obtained as follows.*

1. Set

$$\rho_N = 0, \quad \rho_{k-1} = \min\{m_k + \rho_k,\ r_{k-1}^L\}, \quad k = N, \ldots, 2,\ \rho_0 = 0,$$

$$\rho_k' = \rho_k + r_k^U,\ k = 1, \ldots, N-1, \quad \nu_k = m_k + \rho_k - \rho_{k-1},\ k = 1, \ldots, N.$$

Compute the quasiseparable representation of the matrix V and the basic elements of the representation of the matrix T_σ as well as the vector $\boldsymbol{w}_i = V^H \boldsymbol{y}_i$ as follows.

1.1. Using QR factorization or another algorithm compute the factorization

$$p(N) = V_N \begin{pmatrix} X_N \\ 0_{\nu_N \times r_{N-1}^L} \end{pmatrix}, \tag{3.13}$$

where V_N is a unitary matrix of sizes $m_N \times m_N$, X_N is a matrix of sizes $\rho_{N-1} \times r_{N-1}^L$.

Determine the matrices $p_V(N), d_V(N)$ of sizes $m_N \times \rho_{N-1}, m_N \times \nu_N$ from the partition

$$V_N = \begin{pmatrix} p_V(N) & d_V(N) \end{pmatrix}. \tag{3.14}$$

Compute

$$\begin{pmatrix} h_T(N) \\ d_T(N) \end{pmatrix} = \begin{pmatrix} h(N) \\ V_N^H d(N) \end{pmatrix}, \tag{3.15}$$

$$\begin{pmatrix} c_{N,i} \\ w_i(N) \end{pmatrix} = V_N^H y_i(N), \quad 1 \le i \le \ell \tag{3.16}$$

with the matrices $h_T(N), d_T(N), c_{N,i}, w_i(N)$ of sizes $\rho_{N-1}' \times m_N, \nu_N \times m_N, \rho_{N-1} \times 1, \nu_N \times 1$ respectively.

Compute

$$\gamma_N = \begin{pmatrix} 0_{r_{N-1}^U \times m_N} \\ p_V^H(N) \end{pmatrix}. \tag{3.17}$$

1.2. For $k = N-1, \ldots, 2$ perform the following.

1.2.1. Using QR factorization or another algorithm compute the factorization

$$\begin{pmatrix} p(k) \\ X_{k+1} a(k) \end{pmatrix} = V_k \begin{pmatrix} X_k \\ 0_{\nu_k \times r_{k-1}^L} \end{pmatrix}, \tag{3.18}$$

where V_k is a unitary matrix of sizes $(m_k+\rho_k)\times(m_k+\rho_k)$, X_k is a matrix of sizes $\rho_{k-1}\times r_{k-1}^L$.

Determine the matrices $p_V(k), q_V(k), a_V(k), d_V(k)$ of sizes $m_k\times\rho_{k-1}$, $\rho_k\times\nu_k$, $\rho_k\times\rho_{k-1}$, $m_k\times\nu_k$ from the partition

$$V_k=\begin{pmatrix} p_V(k) & d_V(k)\\ a_V(k) & q_V(k)\end{pmatrix}. \tag{3.19}$$

1.2.2. Compute

$$\begin{pmatrix} h_T(k) & b_T(k)\\ d_T(k) & g_T(k)\end{pmatrix}=\begin{pmatrix} I_{r_{k-1}^U} & 0\\ 0 & V_k^H\end{pmatrix}\begin{pmatrix} h(k) & b(k) & 0\\ d(k) & g(k) & 0\\ X_{k+1}q(k) & 0 & I_{\rho_k}\end{pmatrix}. \tag{3.20}$$

with the matrices $h_T(k), b_T(k), d_T(k), g_T(k)$ of sizes $\rho'_{k-1}\times m_k$, $\rho'_{k-1}\times\rho'_k$, $\nu_k\times m_k$, $\nu_k\times\rho'_k$ respectively.

Compute

$$\gamma_k=\begin{pmatrix} 0_{r_{k-1}^U\times m_k}\\ p_V^H(k)\end{pmatrix} \tag{3.21}$$

and

$$\begin{pmatrix} c_{k,i}\\ w_i(k)\end{pmatrix}=V_k^H\begin{pmatrix} y_i(k)\\ c_{k+1,i}\end{pmatrix},\quad 1\le i\le\ell \tag{3.22}$$

with the vector columns $c_{k,i}, w_i(k)$ of sizes ρ_{k-1}, ν_k respectively.

1.3. Set $V_1=I_{\nu_1}$ and

$$d_T(1)=\begin{pmatrix} d(1)\\ X_2q(1)\end{pmatrix},\ g_T(1)=\begin{pmatrix} g(1) & 0\\ 0 & I_{\rho_1}\end{pmatrix},\quad \gamma_1=\begin{pmatrix} I_{m_1}\\ 0_{\rho_1\times m_1}\end{pmatrix}, \tag{3.23}$$

$$w_i(1)=V_1^H\begin{pmatrix} y_i(1)\\ c_{2,i}\end{pmatrix},\quad 1\le i\le\ell. \tag{3.24}$$

2. For $i=1,\dots,\ell$ perform the following:

2.1. Compute the factorization $T_{\sigma_i}=U_{\sigma_i}R_{\sigma_i}$ and the vector $\boldsymbol{v}^{(i)}=\boldsymbol{v}_{\sigma_i}=U_{\sigma_i}^H\boldsymbol{w}_i$, $\boldsymbol{w}_i=V^H\boldsymbol{y}_i$.

2.1.1. Compute the QR factorization

$$d_T(1)+\sigma_i\gamma_1=U_1^{(i)}\begin{pmatrix} d_R^{(i)}(1)\\ 0_{\rho_1\times m_1}\end{pmatrix}, \tag{3.25}$$

where $U_1^{(i)}$ is a $\nu_1\times\nu_1$ unitary matrix and $d_R^{(i)}(1)$ is an upper triangular $m_1\times m_1$ matrix.

Compute

$$\begin{pmatrix} g_R^{(i)}(1)\\ Y_1^{(i)}\end{pmatrix}=(U_1^{(i)})^H g_T(1), \tag{3.26}$$

$$\begin{pmatrix} v^{(i)}(1) \\ \alpha_1^{(i)} \end{pmatrix} = (U_1^{(i)})^H w_i(1) \tag{3.27}$$

with the matrices $g_R^{(i)}(1), v^{(i)}(1), Y_1^{(i)}, \alpha_1^{(i)}$ *of sizes* $m_1 \times \rho_1', m_1 \times 1, \rho_1 \times \rho_1', \rho_1 \times 1$.

2.1.2. For $k = 2, \ldots, N-1$ *perform the following.*

Compute the QR factorization

$$\begin{pmatrix} Y_{k-1}^{(i)}(h_T(k) + \sigma_i \gamma_k) \\ d_T(k) + \sigma_i d_V^H(k) \end{pmatrix} = U_k^{(i)} \begin{pmatrix} d_R^{(i)}(k) \\ 0_{\rho_k \times m_k} \end{pmatrix}, \tag{3.28}$$

where $U_k^{(i)}$ *is an* $(m_k + \rho_k) \times (m_k + \rho_k)$ *unitary matrix and* $d_R^{(i)}(k)$ *is an* $m_k \times m_k$ *upper triangular matrix.*

Compute

$$\begin{pmatrix} g_R^{(i)}(k) \\ Y_k^{(i)} \end{pmatrix} = (U_k^{(i)})^H \begin{pmatrix} Y_{k-1}^{(i)} b_T(k) \\ g_T(k) \end{pmatrix}, \tag{3.29}$$

$$\begin{pmatrix} v^{(i)}(k) \\ \alpha_k^{(i)} \end{pmatrix} = (U_k^{(i)})^H \begin{pmatrix} \alpha_{k-1}^{(i)} \\ w_i(k) \end{pmatrix} \tag{3.30}$$

with the matrices $g_R^{(i)}(k), v^{(i)}(k), Y_k^{(i)}, \alpha_k^{(i)}$ *of sizes* $m_k \times \rho_k', m_k \times 1, \rho_k \times \rho_k', \rho_k \times 1$.

2.1.3. Compute the QR factorization

$$\begin{pmatrix} Y_N^{(i)} h_T(N) + \sigma_i \gamma_N \\ d_T(N) + \sigma_i d_V^H(N) \end{pmatrix} = U_N^{(i)} d_R^{(i)}(N), \tag{3.31}$$

where $U_N^{(i)}$ *is a unitary matrix of sizes* $(\nu_N + \rho_{N-1}) \times (\nu_N + \rho_{N-1})$ *and* $d_R^{(i)}(N)$ *is an upper triangular matrix of sizes* $m_N \times m_N$.

Compute

$$v^{(i)}(N) = (U_N^{(i)})^H \begin{pmatrix} \alpha_{N-1}^{(i)} \\ w_i(N) \end{pmatrix}. \tag{3.32}$$

2.2. Solve the system $R_{\sigma_i} \boldsymbol{x}^{(i)} = \boldsymbol{v}^{(i)}$.

2.2.1. Compute

$$x^{(i)}(N) = (d_R^{(i)}(N))^{-1} v^{(i)}(N), \quad \eta_{N-1}^{(i)} = (h_T(N) + \sigma_i \gamma_N) v^{(i)}(N)$$

2.2.2. For $k = N-1, \ldots, 2$ *compute*

$$x^{(i)}(k) = (d_R^{(i)}(k))^{-1}(v^{(i)}(k) - g_R^{(i)}(k)\eta_k^{(i)}), \quad \eta_{k-1}^{(i)} = b_T(k)\eta_k^{(i)} + (h_T(k) + \sigma_i \gamma_k) x^{(i)}(k).$$

2.2.3. Compute

$$x^{(i)}(1) = (d_R^{(i)}(1))^{-1}(v^{(i)}(1) - g_R^{(i)}(1)\eta_1^{(i)})$$

Proof. The shifted matrix $A + \sigma I_N$ has the same lower and upper quasiseparable generators as the matrix A and diagonal entries $d(k)+\sigma I_{m_k}$ $(k = 1, \ldots, N)$. To compute the factorization (3.11) we apply Theorem 20.5 from [6]. Directly from Theorem 20.5 we obtain the representation of the unitary matrix V in the form

$$V = \tilde{V}_N \tilde{V}_{N-1} \cdots \tilde{V}_2 \tilde{V}_1, \tag{3.33}$$

where

$$\tilde{V}_1 = V_1 \oplus I_{\phi_1}, \quad \tilde{V}_k = I_{\eta_k} \oplus V_k \oplus I_{\phi_k},\ k = 2, \ldots, N-1, \quad \tilde{V}_N = I_{\eta_N} \oplus V_N$$

with $\eta_k = \sum_{j=1}^{k-1} m_j$, $\phi_k = \sum_{j=k+1}^{N}$ and $(m_k + \rho_k) \times (m_k + \rho_k)$ unitary matrices V_k, as well as the formulas (3.13), (3.14), (3.18), (3.19), $V_1 = I_{\nu_1}$. Here we see that the matrix V does not depend of σ. Moreover the representation (3.33) yields the formulas (3.16), (3.22), (3.24) to compute the vectors $\boldsymbol{w}_i = V^H \boldsymbol{y}_i$, $1 \le i \le \ell$.

Next we apply the corresponding formulas from the same theorem to compute diagonal entries $d_T^\sigma(k)$ $(k = 1, \ldots, N)$ and upper quasiseparable generators $g_T^\sigma(i)$ $(i = 1, \ldots, N-1)$, $h_T^\sigma(j)$ $(j = 2, \ldots, N)$, $b_T^\sigma(k)$ $(k = 2, \ldots, N-1)$ of the matrix T_σ. Hence, we obtain that

$$\begin{pmatrix} h_T^\sigma(N) \\ d_T^\sigma(N) \end{pmatrix} = \begin{pmatrix} I_{r_{N-1}^U} & 0 \\ 0 & V_N^H \end{pmatrix} \begin{pmatrix} h(N) \\ d(N) + \sigma I_{m_N} \end{pmatrix},$$

$$\begin{pmatrix} d_T^\sigma(1) & g_T^\sigma(1) \end{pmatrix} = \begin{pmatrix} d(1) + \sigma I_{m_1} & g(1) & 0 \\ X_2 q(1) & 0 & I_{\rho_1} \end{pmatrix},$$

and for $k = N-1, \ldots, 2$,

$$\begin{pmatrix} h_T^\sigma(k) & b_T^\sigma(k) \\ d_T^\sigma(k) & g_T^\sigma(k) \end{pmatrix} = \begin{pmatrix} I_{r_{k-1}^U} & 0 \\ 0 & V_k^H \end{pmatrix} \begin{pmatrix} h(k) & b(k) & 0 \\ d(k) + \sigma I_{m_k} & g(k) & 0 \\ X_{k+1} q(k) & 0 & I_{\rho_k} \end{pmatrix}.$$

From here we obtain the formulas

$$\begin{gathered} h_T^\sigma(k) = h_T(k) + \sigma \gamma_k,\ k = N, \ldots, 2, \\ d_T^\sigma(k) = d_T(k) + \sigma d_V^*(k),\ k = N, \ldots, 2, \quad d_T^\sigma(1) = d_T(1) + \sigma \gamma_1, \\ g_T^\sigma(k) = g_T(k),\ k = 1, \ldots, N-1, \quad b_T^\sigma(k) = b_T(k),\ k = 2, \ldots, N-1 \end{gathered} \tag{3.34}$$

with $h_T(k), d_T(k), g_T(k), b_T(k)$ and γ_k as in (3.15), (3.17), (3.20), (3.21) and (3.23).

Now by applying Theorem 20.7 from [6] to the matrices T_{σ_i}, $i = 1, 2, \ldots, M$ with the generators determined in (3.34) we obtain the formulas (3.25), (3.26), (3.28), (3.29), (3.31) to compute unitary matrices $U_k^{(i)}$ and quasiseparable generators of the upper triangular R_{σ_i} such that $T_{\sigma_i} = U_{\sigma_i} R_{\sigma_i}$, where

$$U_{\sigma_i} = \tilde{U}_1^{(i)} \tilde{U}_2^{(i)} \cdots \tilde{U}_{N-1}^{(i)} \tilde{U}_N^{(i)} \tag{3.35}$$

with

$$\tilde{U}_1^{(i)} = U_1^{(i)} \oplus I_{\phi_1}, \quad \tilde{U}_k^{(i)} = I_{\eta_k} \oplus U_k^{(i)} \oplus I_{\phi_k},\ k = 2, \ldots, N-1, \quad \tilde{U}_N^{(i)} = I_{\eta_N} \oplus U_N^{(i)}.$$

Moreover the representation (3.35) yields the formulas (3.27), (3.30), (3.32) to compute the vector $\boldsymbol{v}^{(i)} = U_{\sigma_i}^H \boldsymbol{w}_i$, $1 \le i \le l$.

Finally applying Theorem 13.13 from [6] we obtain Step 2.2 to compute the solutions of the systems $R_{\sigma_i} \boldsymbol{x}^{(i)} = \boldsymbol{v}_{(i)}$. □

Concerning the complexity of the previous algorithm we observe that under the simplified assumptions of $r_k^L = r_k^U = r$, $m_i = m_j = m$ and $r << m$ the cost of step 1 is of order $(6r^2m + 2m^2)(Nm)$ whereas the cost of step 2 can be estimated as $(2m^2\ell)(Nm)$ arithmetic operations. Therefore, the proposed algorithm saves at least half of the overall cost of solving ℓ shifted quasiseparable linear systems.

4 Numerical Experiments

The proposed fast algorithm has been implemented in MATLAB.[1] All experiments were performed on a MacBookPro equipped with MATLAB R2016b.

Example 4.1. *Let us test the computation of functions of quasiseparable matrices via series expansion, as outlined in Section 2. We choose A as the 100×100 one-dimensional discretized Laplacian with zero boundary conditions, and g as a random vector with entries taken from a uniform distribution over $[0, 1]$. Define*

$$v_{ex} = 2\pi A e^{At} (e^{2\pi A} - I)^{-1} g,$$

as exact solution (computed in multiprecision) to the problem (2.1), (2.2). Let v_ℓ and $\hat{v}_\ell$ be the approximate solutions obtained from (2.7) and (2.8), respectively, with ℓ expansion terms. Figures 1 and 2 show logarithmic plots of the absolute normwise errors $\|v_{ex} - v_\ell\|_2$ and $\|v_{ex} - \hat{v}_\ell\|_2$ for $t = \pi/2$ and $t = \pi/12$, and values of ℓ ranging from 10 to 500. The results clearly confirm that the formulation (2.8) has improved convergence properties with respect to (2.7). Note that the decreasing behavior of the errors is not always monotone.

Example 4.2. *This example is taken from [15], Example 3.3. We consider here the matrix $A \in \mathbb{R}^{2500 \times 2500}$ stemming from the centered finite difference discretization of the differential operator $-\Delta u + 10u_x$ on the unit square with homogeneous Dirichlet boundary condition. Note that A has (scalar) quasiseparability order 50, but it can also be seen as block 1-quasiseparable with block size 50.*

[1] The code is available at `http://www.unilim.fr/pages_perso/paola.boito/software.html`.

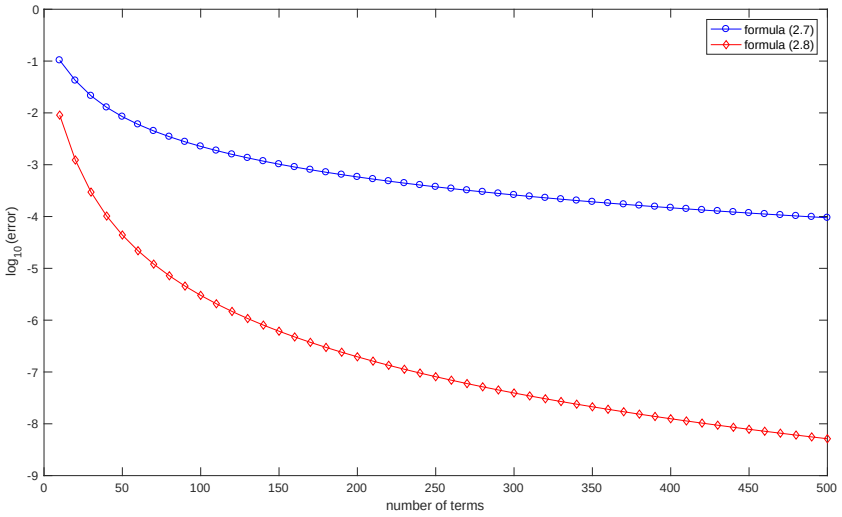

Figure 1: Results for Example 4.1 for $t = \pi/2$. This is a logarithmic plot of the errors given by the application of formulas (2.7) (blue circles) and (2.8) (red diamonds) for the computation of a matrix function times a vector.

We want to compute the matrix function

$$A^{\frac{1}{2}}b \approx \sum_{k=1}^{\ell} \kappa_k(\omega_k^2 I - A)^{-1}Ab,$$

where b is the vector of all ones. We apply the rational approximations presented in [11] as `method2` *and* `method3` *(the latter is designed for the square root function and it is known to have better convergence properties).*

Table 1 shows 2-norm relative errors with respect to the result computed by the MATLAB command `sqrtm(A)*b`*, for several values of ℓ (number of terms in the expansion or number of integration nodes). We have tested three approaches to solve the shifted linear systems involved in the expansion: classical backslash solver, fast structured solver with blocks of size 1 and quasiseparability order 50, and fast structured solver with blocks of size 50 and quasiseparability order 1. For each value of ℓ, the errors are roughly the same for all three approaches: in particular, the fast algorithms appear to be as accurate as standard solvers.*

Example 4.3. *The motivation for this example comes from the classical problem of solving the Poisson equation on a rectangular domain with uni-*

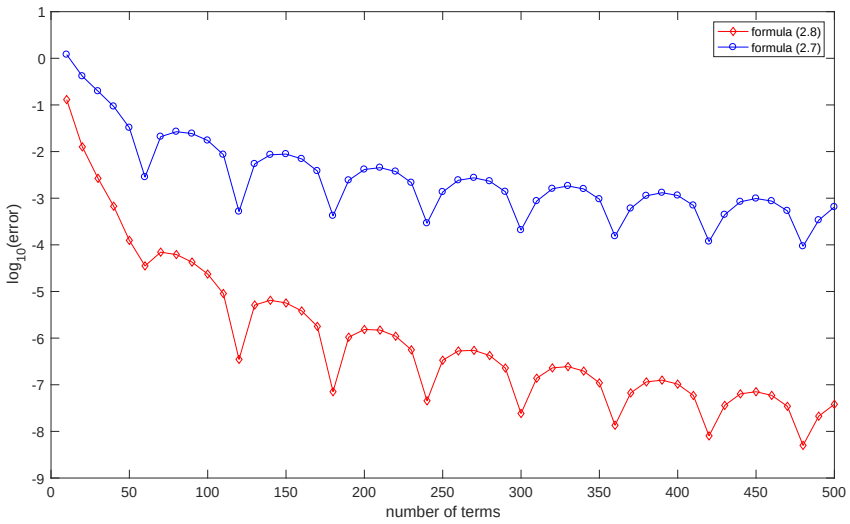

Figure 2: Results for Example 4.1 for $t = \pi/12$. This is a logarithmic plot of the errors given by the application of formulas (2.7) (blue circles) and (2.8) (red diamonds) for the computation of a matrix function times a vector. In this example the errors do not decrease monotonically.

Table 1: Relative errors for Example 4.2.

ℓ	rel. error (`method 2`)	rel. error (`method 3`)
6	4.58e−4	3.25e−5
8	3.03e−5	6.77e−7
10	9.67e−7	9.90e−9
12	1.55e−8	1.16e−10
14	2.06e−9	1.56e−12
16	2.25e−11	2.16e−13
18	2.29e−12	2.09e−13
20	2.44e−13	2.14e−13

Table 2: Relative errors for Example 4.3.

N_b \ N_a	10	25	50	75	100
10	2.18e−15	3.51e−15	2.78e−15	7.25e−15	1.18e−14
25	1.80e−15	9.66e−15	1.34e−14	3.48e−14	5.74e−14
50	9.58e−16	1.98e−14	2.05e−14	9.34e−14	2.14e−13
75	3.93e−15	2.14e−14	2.04e−14	1.93e−13	4.15e−13
100	5.55e−15	2.30e−14	4.58e−14	2.11e−13	5.61e−13

form zero boundary conditions. The equation takes the form

$$\Delta u(x,y) = f(x,y), \quad \textit{with} \quad 0 < x < a,\, 0 < y < b,$$

and its finite difference discretization yields a matrix equation

$$AX + XB = F \quad \textit{with} \quad X, F \in \mathbb{R}^{N_b \times N_a}, \tag{4.36}$$

where N_a is the number of grid points taken along the x direction and N_b is the number of grid points along the y direction. Here A and B are matrices of sizes $N_b \times N_b$ and $N_a \times N_a$, respectively, and both are Toeplitz symmetric tridiagonal with nonzero entries $\{-1, 2, 1\}$. See e.g., [18] for a discussion of this problem.

A widespread approach consists in reformulating the matrix equation (4.36) *as a larger linear system of size $N_a N_b \times N_a N_b$ via Kronecker products. Here we apply instead the idea outlined in Section 2.2: compute the (well-known) Schur decomposition of B, that is, $B = UDU^H$, and solve the equation $A(XU) + (XU)D = FU$. Note that, since D is diagonal, this equation can be rewritten as a collection of shifted linear systems, where the right-hand side vector may depend on the shift.*

Table 2 shows relative errors on the solution matrix X, computed w.r.t. the solution given by a standard solver applied to the Kronecker linear system. Here we take F as the matrix of all ones.

In the next examples we test experimentally the complexity of our algorithm.

Example 4.4. *We consider matrices $A_n \in \mathbb{R}^{n \times n}$ defined by random quasiseparable generators of order* 3*. The first column of Table 3 shows the running times of our structured algorithm applied to randomly shifted systems $(A_n + \sigma_i I_n)\mathbf{x_i} = \mathbf{y}$, for $i = 1, \ldots, 50$ and growing values of n. The same data are plotted in Figure 3: the growth of the running times looks linear with n, as predicted by theoretical complexity estimates.*

Table 3: Running times in seconds for Example 4.4.

system size n	fast algorithm	sequential algorithm	ratio
100	0.6016	1.2049	2.0029
200	0.8473	1.6382	1.9334
300	1.2291	2.4377	1.9833
400	1.6639	3.2492	1.9528
500	2.1976	4.0544	1.8449
600	2.6654	5.1508	1.9325
700	2.9765	5.8691	1.9718
800	3.3367	6.5671	1.9681
900	3.8411	7.6630	1.9950
1000	4.2435	8.4006	1.9796

The second column of Table 3 shows timings for the structured algorithm applied sequentially (i.e., without re-using the factorization (3.11)*) to the same set of shifted systems. The gain obtained by the fast algorithm of Section 3 w.r.t. a sequential structured approach amounts to a factor of about* 2*, which is consistent with the discussion at the end of Section 3. Experiments with a different number of shifts yield similar results.*

Example 4.5. *In this example we study the complexity of our algorithm w.r.t. block size (that is, the parameter* m *at the end of Section 3). We choose* $N = 2$*,* $\ell = 2$*,* $r_L = r_U = 1$*, with random quasiseparable generators, and focus on large values of* m*. Running times, each of them averaged over ten trials, are shown in Table 4. A log-log plot is given in Figure 4, together with a linear fit, which shows that the experimental growth of these running times is consistent with theoretical complexity estimates.*

5 Conclusion

In this paper we have presented an effective algorithm based on QR decomposition for solving a possibly large number of shifted quasiseparable systems. Two main motivations are the computation of a meromorphic function of a quasiseparable matrix and the solution of linear matrix equations with quasiseparable matrix coefficients. The experiments performed suggest that our algorithm has good numerical properties.

Future work includes the analysis of methods based on the Mittag-Leffler's theorem for computing meromorphic functions of quasiseparable matrices. Approximate expansions of Mittag-Leffler type can be obtained by using the

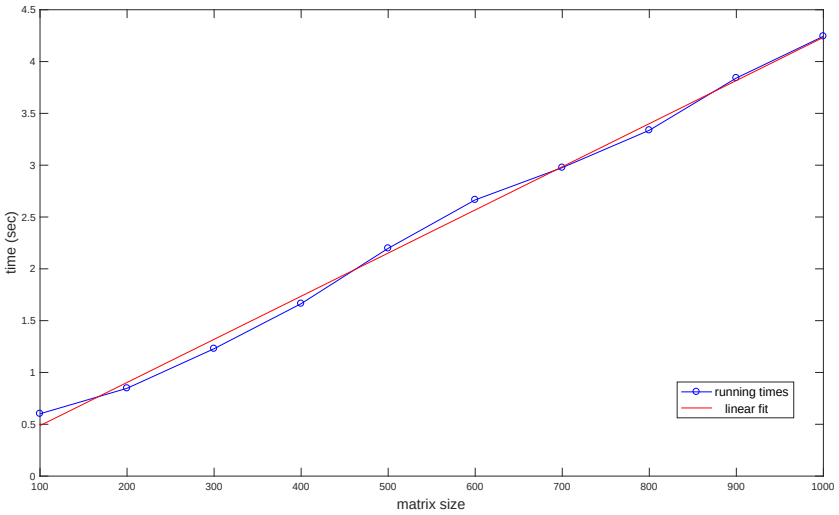

Figure 3: Running times for Example 4.4. The linear fit appears to be a good approximation of the actual data. Its equation is $y = 0.0042x + 0.071$.

Table 4: Running times in seconds for Example 4.5.

block size m	running time (sec)
400	0.1481
800	1.0527
1200	3.4238
1600	7.6118
2000	15.2020
2400	26.3695
2800	40.1458
3200	61.2488

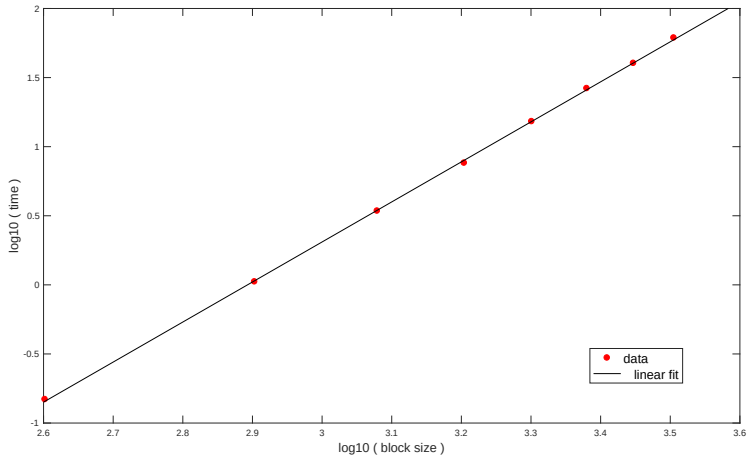

Figure 4: Log-log plot of running times versus block size m for Example 4.5. The equation of the linear fit is $y = 2.9x - 8.4$, which confirms that the complexity of the algorithm grows as m^3.

Carathéodory-Fejér approximation theory (see [8]). The application of these techniques for computing quasiseparable matrix functions is an ongoing research.

References

[1] Pierluigi Amodio and Marcin Paprzycki. A cyclic reduction approach to the numerical solution of boundary value ODEs. *SIAM J. Sci. Comput.*, 18(1):56–68, 1997. Dedicated to C. William Gear on the occasion of his 60th birthday.

[2] J. Ballani and D. Kressner. Matrices with hierarchical low-rank structure. In *CIME Summer School on Exploiting Hidden Structure in Matrix Computations.*

[3] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the Matrix Equation $ax + xb = c$. *Comm. of the ACM*, 15(9):820–826, 1972.

[4] Dario A Bini, Stefano Massei, and Leonardo Robol. Efficient cyclic reduction for quasi-birth–death problems with rank structured blocks. *Applied Numerical Mathematics*, 2016.

[5] Y. Eidelman and I. Gohberg. A modification of the Dewilde-van der Veen method for inversion of finite structured matrices. *Linear Algebra*

Appl., 343/344:419–450, 2002. Special issue on structured and infinite systems of linear equations.

[6] Yuli Eidelman, Israel Gohberg, and Iulian Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 1*, volume 234 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer, Basel, 2014. Basics. Completion problems. Multiplication and inversion algorithms.

[7] Yuli Eidelman, Israel Gohberg, and Iulian Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 2*, volume 235 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer Basel AG, Basel, 2014. Eigenvalue method.

[8] Roberto Garrappa and Marina Popolizio. On the use of matrix functions for fractional partial differential equations. *Mathematics and Computers in Simulation*, 81(5):1045–1056, 2011.

[9] J. Gondzio and P. Zhlobich. Multilevel quasiseparable matrices in pde-constrained optimization. Technical Report ERGO-11-021, School of Mathematics, The University of Edinburgh, 2011.

[10] G.-D. Gu and V. Simoncini. Numerical solution of parameter-dependent linear systems. *Numer. Linear Algebra Appl.*, 12(9):923–940, 2005.

[11] Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. Computing $\mathbf{A}^{\alpha}$, $\log(\mathbf{A})$, and related matrix functions by contour integrals. *SIAM J. Numer. Anal.*, 46(5):2505–2523, 2008.

[12] Stefano Massei and Leonardo Robol. Decay bounds for the numerical quasiseparable preservation in matrix functions. *arXiv preprint arXiv:1608.01576*, 2016.

[13] V. B. Sherstyukov. Expansion of the reciprocal of an entire function with zeros in a strip in the Kreĭn series. *Mat. Sb.*, 202(12):137–156, 2011.

[14] V. Simoncini. Computational Methods for Linear Matrix Equations. *SIAM Rev.*, 58(3):377–441, 2016.

[15] Valeria Simoncini. Extended Krylov subspace for parameter dependent systems. *Applied numerical mathematics*, 60(5):550–560, 2010.

[16] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. *Matrix computations and semiseparable matrices. Vol. 1*. Johns Hopkins University Press, Baltimore, MD, 2008. Linear systems.

[17] Raf Vandebril, Marc Van Barel, and Nicola Mastronardi. *Matrix computations and semiseparable matrices. Vol. II.* Johns Hopkins University Press, Baltimore, MD, 2008. Eigenvalue and singular value methods.

[18] Frederic Y.M. Wan. An in-core finite difference method for separable boundary value problems on a rectangle. *Studies in Applied Mathematics*, 52(2):103–113, 1973.