THE QR FACTORIZATION FOR BANDED-PLUS-SEMISEPARABLE MATRICES IS COMPUTABLE IN LINEAR COMPLEXITY

TAO CHEN* AND SHEEHAN OLVER*

Abstract. We show that each stage of the QR factorization of banded-plus-semiseparable matrices computed using Householder reflections has a specific structured perturbation. This theoretical result enables the design of linear-complexity algorithms for QR factorization and for solving the associated linear systems. Numerical experiments validate the optimal linear complexity and demonstrate substantial speedups compared with existing hierarchical approaches. The algorithms have been implemented in an open-source Julia package, providing an efficient and accessible platform for practical use.

Key words. banded-plus-semiseparable matrices, QR factorization, linear complexity, structured matrices, direct solvers

AMS subject classifications. 65F05, 65F50, 15A23, 65Y20

1. Introduction.

Do we want to allow for complex numbers?

What to say about QR algorithm?

Add a nullspace problem coming from the ultraspherical spectral method?

Does it generalise to rectangular?

Say something about inverses of banded matrices being banded-plus-semiseparable?

Add stability plot

Banded-plus-semiseparable (BPS) matrices, expressible as

Change all the ^{T} to $^\top$

$$A = \underbrace{B}_{\text{banded}} + \underbrace{\operatorname{tril}(UV^T, -1)}_{\text{lower semiseparable, rank } r} + \underbrace{\operatorname{triu}(WS^T, 1)}_{\text{upper semiseparable, rank } p} \in \mathbb{R}^{n \times n},$$

arise in numerous applications from PDEs with non-local interactions [12] to signal processing, control theory, and eigenvalue problems [18]. Their structure requires only $O(n)$ storage which invites the development of $O(n)$ algorithms, a goal successfully achieved for iterations of the QR algorithm for symmetric semiseparable systems [17], and for solving linear systems with *diagonal*-plus-semiseparable matrices [9]. However, generalizing these results to the case where the banded part B is a genuine banded matrix, rather than merely a diagonal one, presents significant algorithmic challenges.

This is a bizarre reference and description. Is it AI generated?

Add citations to Arieh Iserles W-systems papers

Clarify exact relationship between prior work and ours

Pioneering work established $O(n)$ solvers for sequentially semiseparable matrices [2] and later for the banded-plus-semiseparable case via ULV factorization [3]. Banded-plus-semiseparable matrices can be viewed as hierarchically semiseparable (HSS) matrices, and solvers using HSS structure is a well-developed area [4, 1, 20, 13]. A parallel line of research extensively developed the theory and algorithms for semiseparable and quasiseparable matrices, including implicit QR algorithms for *symmetric* semiseparable matrices [17], structure-preserving analyses [10, 6, 5], approaches lever-

What's a sequentially semiseparable matrix?

*Department of Mathematics, Imperial College London, London, UK (t.chen24@imperial.ac.uk, s.olver@imperial.ac.uk).

aging rational Krylov techniques [11, 19], and alternative representations [18, 7]. Despite these advances, a clear theoretical guarantee that the standard QR factorization preserves the BPS structure has been missing, with most existing solvers relying on more complex ULV or intricate Givens-based schemes [14, 3, 16, 8]. A special case of BPS matrices are almost banded matrices which were used in [15] to represent discretisations of differential equations using the ultraspherical spectral method. An optimal complexity adaptive QR factorization was introduced, which also gives an optimal complexity QR factorization for BPS matrices with now lower semiseparable part ($r = 0$). It also introduced an optimal complexity back-substitution for upper-triangular BPS matrices, an algorithm we also use.

In this paper, we close this theoretical gap. We prove that the QR factorization of a BPS matrix yields a factor matrix F, which is the matrix containing both R and the Householder reflectors encoding Q, that is itself BPS, with precisely characterized lower rank r, upper rank $r + p$, and bandwidths l and $l + m$. This pivotal result, proven via an inductive framework involving a new class of Householder-Modified BPS Matrices (HMBPSM), which enables the design of an $O(n)$ QR factorization. Furthermore, it facilitates a complete direct solver: applying Q^T and performing backward substitution on the structured factor R are also achieved in linear time. Our work thus provides a unified, QR-based, end-to-end $O(n)$ solution for BPS systems, backed by a rigorous structure-preservation theorem.

The rest of this paper is organized as follows. Section 2 presents our main theoretical contributions: the definitions, the core lemma on structure preservation under Householder transformations, and the main theorem with its proof. Section 3 details the resulting $O(n)$ algorithms for QR factorization, application of Q^T, and backward substitution. Section 4 presents numerical experiments that confirm the linear complexity and demonstrate performance advantages. We conclude in Section 5 with a discussion of future work.

2. Main results.

2.1. Problem Formulation and Notation.

Before we start, an important notation will be: for a matrix M, let $M[i : j, m : n]$ represent the submatrix of M from row i to row j and from column m to column n. When $i = j$ or $m = n$, the notation will be simplified as $M[i, m : n]$ or $M[i : j, m]$. We also adopt the convention that writing **end** in an index(such as $M[i : \text{end}, :]$) to indicates the last valid index in that dimension.

ctually de- And I it would aner if its a n vector

t think we d use end

Definition 2.1. *A banded-plus-semseparable matrix (BPS) with lower-semisepable rank r, upper-semiseparable rank p, lower-bandwidth l and upper-bandwidth m is $A \in \mathrm{BPS}^{n\times n}_{(r,p),(l,u)} \subset \mathbb{R}^{n\times n}$ such that*

$$A = B + \mathrm{tril}(UV^T, -1) + \mathrm{triu}(WS^T, 1)$$

lly use u for -bandwidth

where $U, V \in \mathbb{R}^{n\times r}$, $W, S \in \mathbb{R}^{n\times p}$, and $B = (b_{ij})^n_{i,j=1} \in \mathbb{R}^{n\times n}$ is a banded matrix satisfying $b_{ij} = 0$ for $i - j > l$ or $j - i > m$.

ge i to k

Define vectors $\bar{\mathbf{u}}_i = U[i, :]^T \in \mathbb{R}^r$, $\bar{\mathbf{v}}_i = V[i, :]^T \in \mathbb{R}^r$, $\bar{\mathbf{w}}_i = W[i, :]^T \in \mathbb{R}^p$, and $\bar{\mathbf{s}}_i = S[i, :]^T \in \mathbb{R}^p$ for $i = 1, \ldots, n$. The matrix A can then be expressed element-wise

dn't these be lumns of V? hy the bar?

as:

number
ions that
ted

$$A = \begin{bmatrix} b_{11} & \bar{\mathbf{w}}_1^T \bar{\mathbf{s}}_2 + b_{12} & \cdots & \bar{\mathbf{w}}_1^T \bar{\mathbf{s}}_n + b_{1n} \\ \bar{\mathbf{u}}_2^T \bar{\mathbf{v}}_1 + b_{21} & b_{22} & \cdots & \bar{\mathbf{w}}_2^T \bar{\mathbf{s}}_n + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{u}}_n^T \bar{\mathbf{v}}_1 + b_{n1} & \bar{\mathbf{u}}_n^T \bar{\mathbf{v}}_2 + b_{n2} & \cdots & b_{nn} \end{bmatrix}. \tag{2.1}$$

Applying the QR factorization to A yields a factor matrix F, whose upper triangular part stores the matrix R and whose lower triangular part contains the Householder reflection vectors $\mathbf{y}$ generated during the factorization. We will demonstrate that F itself retains a banded-plus-semiseparable structure. Specifically, its lower semiseparable part has rank r, its upper semiseparable part has rank $r+p$, its lower bandwidth is l, and its upper bandwidth is $l+m$.

Before proceeding with the detailed proof, let us clarify the precise structure of the factor matrix F obtained from the QR factorization. In this work, following the convention of LAPack , we employ a compact representation that stores the complete information of the QR factorization in a single matrix:

add citation

$$F = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ y_{2,1} & r_{22} & r_{23} & \cdots & r_{2n} \\ y_{3,1} & y_{3,2} & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n,1} & y_{n,2} & y_{n,3} & \cdots & r_{nn} \end{bmatrix} \tag{2.2}$$

where: the upper triangular part (including the main diagonal) of F stores the elements of the upper triangular matrix R, i.e.: $R = \mathrm{triu}(F)$, and thestrictly lower triangular part (excluding the main diagonal) of F stores the last $n-k$ elements of (rescaled) Householder reflection vectors $\mathbf{y}_k$ generated at each step.

More specifically, at the k-th Householder transformation step ($k = 1, 2, \ldots, n-1$), we construct a reflection vector $\mathbf{y}_k$ to eliminate the subdiagonal entries of the k-th column. This vector takes the form:

$$\mathbf{y}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ y_{k+1,k} \\ \vdots \\ y_{n,k} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} k-1 \text{ zeros} \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n-k \text{ elements} \end{matrix} \tag{2.3}$$

Following the LAPack format, we normalize $\mathbf{y}_k$ such that its first nonzero element (the k-th element) equals 1. Consequently, we only need to store the elements from position $k+1$ to n of this vector, which are placed in the k-th column of F, from row $k+1$ to n.

The advantage of this representation is that it compactly stores the information of both the orthogonal matrix Q (via the Householder vectors) and the upper triangular matrix R within a single matrix F. The central result of this paper will demonstrate that for a banded-plus-semiseparable matrix A, this factor matrix F itself maintains a banded-plus-semiseparable structure.

It is important to note that with this normalization convention (where the first nonzero element of each Householder vector $\mathbf{y}_k$ is 1), the full Householder transformation at the k-th step is given by $I - \tau_k \mathbf{y}_k \mathbf{y}_k^T$, where τ_k is a scalar coefficient. Therefore, in addition to the factor matrix F, a vector $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_{n-1})^T$ is required to completely represent the QR factorization. The orthogonal matrix Q can be reconstructed as the product $Q = (I - \tau_1 \mathbf{y}_1 \mathbf{y}_1^T)(I - \tau_2 \mathbf{y}_2 \mathbf{y}_2^T) \cdots (I - \tau_{n-1} \mathbf{y}_{n-1} \mathbf{y}_{n-1}^T)$.

Throughout our analysis, we will focus on the structure of the factor matrix F, while acknowledging that the complete QR representation consists of the pair $(F, \boldsymbol{\tau})$. Our main theorem establishes that F maintains the banded-plus-semiseparable structure; the scaling coefficients τ_k can be stored separately without affecting the structural properties of the algorithm.

We proceed to prove this by induction. First, we introduce two key definitions and a pivotal lemma.

2.2. Core Definitions and a Key Lemma. While the final factor matrix of a QR factorization is a BPS matrix, at intermediate stages it has a specific structured perturbation. Here we describe this structure in terms of a linear space that, at each stage, the principle submatrix lies in:

DEFINITION 2.2 (BPS Factors Perturbation Matrix). *Given*

$$A = B + \mathrm{tril}(UV^T, -1) + \mathrm{triu}(WS^T, 1) \in \mathrm{BPS}^{n\times n}_{(r,p),(l,u)}$$

a BPS Factors Perturbation Matrix *(BPSFPM) is in the vector space*

$$\begin{aligned}
\mathcal{P}(A) := \Bigg\{ & UQS^T + UKU^TA + UE + XS^T + YU^TA + Z : \\
& Q \in \mathbb{R}^{r\times p}, K \in \mathbb{R}^{r\times r}, \\
& E = [E_s \in \mathbb{R}^{r\times \min(l+m,n)}, \mathbf{0}] \in \mathbb{R}^{r\times n}, \\
& X = \begin{bmatrix} X_s \in \mathbb{R}^{\min(l,n)\times p} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n\times p}, \\
& Y = \begin{bmatrix} Y_s \in \mathbb{R}^{\min(l,n)\times r} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n\times r}, \\
& Z = \begin{bmatrix} Z_s \in \mathbb{R}^{\min(l,n)\times \min(l+m,n)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n\times n} \Bigg\} \subset \mathbb{R}^{n\times n}.
\end{aligned}$$

In addition, we need to describe the structure of the upper-triangular part in terms of a structured vector:

DEFINITION 2.3 (BPS Factors Vector). *Given $A \in \mathrm{BPS}^{n\times n}_{(r,p),(l,u)}$ a* BPS Factors Vector *(BPSFV) is a row vector in the vector space*

d we change
umn vector?

$$\begin{aligned}
\mathcal{V}(A) := \Bigg\{ & \mathbf{d}^T + \boldsymbol{\alpha}^T (S^T[:, 2:n]) + \boldsymbol{\beta}^T ((U^TA)[:, 2:n]) : \\
& \mathbf{d} = \begin{bmatrix} \mathbf{d}_s \in \mathbb{R}^{\min(l+m,n-1)} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n-1}, \boldsymbol{\alpha} \in \mathbb{R}^p, \boldsymbol{\beta} \in \mathbb{R}^r \Bigg\} \subset \mathbb{R}^{1\times(n-1)}.
\end{aligned}$$

LEMMA 2.4. *Given $A \in \mathrm{BPS}^{n\times n}_{(r,p),(l,u)}$ and $P \in \mathcal{P}(A)$, suppose a Householder transformation is applied to $A + P$ to eliminate the subdiagonal entries of its first column, yielding $\tilde{C} = (I - \tau \mathbf{y}\mathbf{y}^T)(A + P)$. Then the following hold:*

alise so it
't need to be
eholder

1. *The principal submatrix satisfies* $\tilde{C}[2:n,2:n] = A[2:n,2:n] + \tilde{P}$ *for* $\tilde{P} \in \mathcal{P}(A[2:n,2:n])$.
2. *The first row satisfies* $\tilde{C}[1,2:n] \in \mathcal{V}(A)$.

Proof.

I'll start updating the proof tomorrow

Let us introduce the necessary notation:

- $A = B + \mathrm{tril}(UV^T,-1) + \mathrm{triu}(WS^T,1)$, where $U = (\mathbf{u}_1,...,\mathbf{u}_r) \in \mathbb{R}^{n\times r}$, $V = (\mathbf{v}_1,...,\mathbf{v}_r) \in \mathbb{R}^{n\times r}$, $W = (\mathbf{w}_1,...,\mathbf{w}_p) \in \mathbb{R}^{n\times p}$, and $S = (\mathbf{s}_1,...,\mathbf{s}_p) \in \mathbb{R}^{n\times p}$. Here $\mathbf{u}_i = (u_1^{(i)},...,u_n^{(i)})^T \in \mathbb{R}^n$ and $\mathbf{v}_i = (v_1^{(i)},...,v_n^{(i)})^T \in \mathbb{R}^n$ for $i = 1,...,r$; $\mathbf{w}_i = (w_1^{(i)},...,w_n^{(i)})^T \in \mathbb{R}^n$ and $\mathbf{s}_i = (s_1^{(i)},...,s_n^{(i)})^T \in \mathbb{R}^n$ for $i = 1,...,p$. Also, $B = (b_{ij})_{i,j=1}^n \in \mathbb{R}^{n,n}$ with $b_{ij} = 0$ if $i - j > l$ or $j - i > m$.
- $C = A + UQS^T + UKU^TA + UE + XS^T + YU^TA + Z$, where Q, K, E, X, Y, Z are as in Definition Definition 2.2.
- $\tilde{C} = (I - \tau\mathbf{y}\mathbf{y}^T)C$, where the Householder vector $\mathbf{y}$ can be expressed as $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$. Here, $\mathbf{e}_1 = (1,0,...,0)^T \in \mathbb{R}^n$, $U^{(2)} \in \mathbb{R}^{n\times r}$ satisfies $U^{(2)}[1,:] = \mathbf{0}$ and $U^{(2)}[2:n,:] = U[2:n,:]$, $\bar{\mathbf{k}} \in \mathbb{R}^r$, $\mathbf{b} = (0, b_2, \dots, b_{\min(l+1,n)}, 0, \dots, 0)^T \in \mathbb{R}^n$, and τ is a coefficient found to satisfy the definition of a Householder transformation.

Let $\bar{\mathbf{u}}_1 = (u_1^{(1)}, \dots, u_1^{(r)})^T \in \mathbb{R}^r$. We can write:

$$\mathbf{e}_1^T A = \underbrace{\mathbf{d}_1^T}_{\min(m+1,n)\text{ nonzero entries}} + \underbrace{\bar{\mathbf{w}}_1^T}_{\in\mathbb{R}^{1\times p}} S^T,$$

where $\mathbf{d}_1 = B[1,:]^T \in \mathbb{R}^n$, $\bar{\mathbf{w}}_1 = (w_1^{(1)}, \dots, w_1^{(p)})^T \in \mathbb{R}^p$, and

$$\mathbf{b}^T A = \underbrace{\bar{\mathbf{d}}^T}_{\min(l+m+1,n)\text{ nonzero entries}} + \underbrace{\mathbf{f}^T}_{\mathbf{b}^T W \in \mathbb{R}^{1\times p}} S^T.$$

Define the auxiliary vectors:

$$\mathbf{c}_1 = Q^T U^T \mathbf{y} \in \mathbb{R}^p \tag{2.4}$$

$$\mathbf{c}_2 = K^T U^T \mathbf{y} \in \mathbb{R}^r \tag{2.5}$$

$$\mathbf{c}_3 = U^T \mathbf{y} \in \mathbb{R}^r \tag{2.6}$$

$$\mathbf{c}_4 = X^T \mathbf{y} \in \mathbb{R}^p \tag{2.7}$$

$$\mathbf{c}_5 = Y^T \mathbf{y} \in \mathbb{R}^r \tag{2.8}$$

$$\mathbf{c}_6 = Z^T \mathbf{y} \in \mathbb{R}^n, \quad \text{which has the form } \mathbf{c}_6 = \begin{bmatrix} \mathbf{c}_{6s} \\ \mathbf{0} \end{bmatrix} \text{ with } \mathbf{c}_{6s} \in \mathbb{R}^{\min(l+m,n)}. \tag{2.9}$$

Also, let $\mathbf{x}^{(1)} = X[1,:]^T \in \mathbb{R}^p$, $\mathbf{y}^{(1)} = Y[1,:]^T \in \mathbb{R}^r$, and $\mathbf{z}^{(1)} = Z[1,:]^T \in \mathbb{R}^n$.

We now compute $(I - \tau\mathbf{y}\mathbf{y}^T)C$ by distributing the transformation over each term in the definition of C.

(i) Transformation of A**:** Substituting the expressions $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$,

$\mathbf{e}_1^T A = \mathbf{d}_1^T + \bar{\mathbf{w}}_1^T S^T$, and $\mathbf{b}^T A = \bar{\mathbf{d}}^T + \mathbf{f}^T S^T$, we obtain:

(2.10)
$$\begin{aligned}(I-\tau\mathbf{y}\mathbf{y}^T)A =& A + \mathbf{e}_1\big[\underbrace{(-\tau\mathbf{d}_1^T - \tau\bar{\mathbf{d}}^T)}_{\min(l+m+1,n)\text{ nonzero entries}} + \underbrace{(-\tau\bar{\mathbf{w}}_1^T - \tau\mathbf{f}^T)}_{\in\mathbb{R}^{1\times p}} S^T \\ &+ \underbrace{(-\tau\bar{\mathbf{k}}^T)}_{\in\mathbb{R}^{1\times r}} U^{(2)T} A\big] + U^{(2)}\underbrace{(-\tau\bar{\mathbf{k}}\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{k}}\mathbf{f}^T)}_{\in\mathbb{R}^{r\times p}} S^T \\ &+ U^{(2)}\underbrace{(-\tau\bar{\mathbf{k}}\bar{\mathbf{k}}^T)}_{\in\mathbb{R}^{r\times r}} U^{(2)T} A + U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{d}_1^T - \tau\bar{\mathbf{k}}\bar{\mathbf{d}}^T) \\ &+ (-\tau\mathbf{b}\bar{\mathbf{w}}_1^T - \tau\mathbf{b}\mathbf{f}^T)S^T + (-\tau\mathbf{b}\bar{\mathbf{k}}^T)U^{(2)T}A + (-\tau\mathbf{b}\mathbf{d}_1^T - \tau\mathbf{b}\bar{\mathbf{d}}^T).\end{aligned}$$

Dropping the first column, we see that the first row of the term in brackets is in $\mathcal{V}(A)$. Dropping the first row and column of the remaining terms are in $\mathcal{P}(A[2:n,2:n])$.

(ii) Transformation of UQS^T: Substituting the expressions $\mathbf{y}^T UQ = \mathbf{c}_1^T$, $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, and $U = \mathbf{e}_1\bar{\mathbf{u}}_1^T + U^{(2)}$ where $\bar{\mathbf{u}}_1 = (u_1^{(1)}, ..., u_1^{(r)}) \in \mathbb{R}^r$, we obtain

(2.11)
$$(I-\tau\mathbf{y}\mathbf{y}^T)UQS^T = \mathbf{e}_1(\bar{\mathbf{u}}_1^T Q - \tau\mathbf{c}_1^T)S^T + U^{(2)}(Q - \tau\bar{\mathbf{k}}\mathbf{c}_1^T)S^T + (-\tau\mathbf{b}\mathbf{c}_1^T)S^T.$$

(iii) Transformation of UKU^TA: Substituting the expressions $\mathbf{y}^T UK = \mathbf{c}_2^T$, $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, $U = \mathbf{e}_1\bar{\mathbf{u}}_1^T + U^{(2)}$, and $\mathbf{e}_1^T A = \mathbf{d}_1^T + \bar{\mathbf{w}}_1^T S^T$, we obtain

(2.12)
$$\begin{aligned}&(I-\tau\mathbf{y}\mathbf{y}^T)UKU^TA \\ =&\mathbf{e}_1(\bar{\mathbf{u}}_1^T K\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T) + \mathbf{e}_1(\bar{\mathbf{u}}_1^T K\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + \mathbf{e}_1(\bar{\mathbf{u}}_1^T K - \tau\mathbf{c}_2^T)U^{(2)T}A \\ &+ U^{(2)}(K\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + U^{(2)}(K - \tau\bar{\mathbf{k}}\mathbf{c}_2^T)U^{(2)T}A + U^{(2)}(K\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T) \\ &+ (-\tau\mathbf{b}\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + (-\tau\mathbf{b}\mathbf{c}_2^T)U^{(2)T}A + (-\tau\mathbf{b}\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T).\end{aligned}$$

(iv) Transformation of UE: Substituting the expressions $\mathbf{y}^T U = \mathbf{c}_3^T$, $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, and $U = \mathbf{e}_1\bar{\mathbf{u}}_1^T + U^{(2)}$, we obtain

$$(I-\tau\mathbf{y}\mathbf{y}^T)UE = \mathbf{e}_1(\bar{\mathbf{u}}_1^T E - \tau\mathbf{c}_3^T E) + U^{(2)}(E - \tau\bar{\mathbf{k}}\mathbf{c}_3^T E) + (-\tau\mathbf{b}\mathbf{c}_3^T E). \tag{2.13}$$

(v) Transformation of XS^T: Substituting the expressions $\mathbf{y}^T X = \mathbf{c}_4^T$ and $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, we obtain

$$(I-\tau\mathbf{y}\mathbf{y}^T)XS^T = \mathbf{e}_1(-\tau\mathbf{c}_4^T)S^T + U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{c}_4^T)S^T + (X - \tau\mathbf{b}\mathbf{c}_4^T)S^T. \tag{2.14}$$

(vi) Transformation of YU^TA: Substituting the expressions $\mathbf{y}^T Y = \mathbf{c}_5^T$, $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, $U = \mathbf{e}_1\bar{\mathbf{u}}_1^T + U^{(2)}$, and $\mathbf{e}_1^T A = \mathbf{d}_1^T + \bar{\mathbf{w}}_1^T S^T$, we obtain

(2.15)
$$\begin{aligned}&(I-\tau\mathbf{y}\mathbf{y}^T)YU^TA \\ =&\mathbf{e}_1(-\tau\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T) + \mathbf{e}_1(-\tau\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + \mathbf{e}_1(-\tau\mathbf{c}_5^T)U^{(2)T}A \\ &+ U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{c}_5^T)U^{(2)T}A + U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T) \\ &+ (Y\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\mathbf{b}\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)S^T + (Y - \tau\mathbf{b}\mathbf{c}_5^T)U^{(2)T}A + (Y\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{b}\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T).\end{aligned}$$

(vii) Transformation of Z: Substituting the expressions $\mathbf{y}^T Z = \mathbf{c}_6^T$ and $\mathbf{y} = \mathbf{e}_1 + U^{(2)}\bar{\mathbf{k}} + \mathbf{b}$, we obtain

$$(I-\tau\mathbf{y}\mathbf{y}^T)Z = \mathbf{e}_1(-\tau\mathbf{c}_6^T) + U^{(2)}(-\tau\bar{\mathbf{k}}\mathbf{c}_6^T) + (Z - \tau\mathbf{b}\mathbf{c}_6^T). \tag{2.16}$$

Combining equations (2.10) through (2.16), we can now identify the structure of the resulting matrix $\tilde{C}$.

Firstly, the submatrix $\tilde{C}[2:n,2:n]$ satisfies:

$$\tilde{C}[2:n,2:n] = \tilde{A} + \tilde{U}\tilde{Q}\tilde{S}^T + \tilde{U}\tilde{K}\tilde{U}^T\tilde{A} + \tilde{U}\tilde{E} + \tilde{X}\tilde{S}^T + \tilde{Y}\tilde{U}^T\tilde{A} + \tilde{Z}, \tag{2.17}$$

where

$$\begin{aligned}\tilde{A} &= A[2:n,2:n]\\ \tilde{U} &= U[2:n,:]\\ \tilde{S} &= S[2:n,:]\end{aligned}$$

and the updated modification matrices are given by:

$$\begin{aligned}
\tilde{Q} &= -\tau\bar{\mathbf{k}}\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{k}}\mathbf{f}^T + Q - \tau\bar{\mathbf{k}}\mathbf{c}_1^T + K\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_4^T - \tau\bar{\mathbf{k}}\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T,\\
\tilde{K} &= -\tau\bar{\mathbf{k}}\bar{\mathbf{k}}^T + K - \tau\bar{\mathbf{k}}\mathbf{c}_2^T - \tau\bar{\mathbf{k}}\mathbf{c}_5^T,\\
\tilde{E} &= [\tilde{E}_s, \mathbf{0}] \in \mathbb{R}^{r\times(n-1)}, \quad \text{with}\\
\tilde{E}_s &= (-\tau\bar{\mathbf{k}}\mathbf{d}_1^T - \tau\bar{\mathbf{k}}\bar{\mathbf{d}}^T + K\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T + E - \tau\bar{\mathbf{k}}\mathbf{c}_3^TE\\
&\quad - \tau\bar{\mathbf{k}}\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\bar{\mathbf{k}}\mathbf{c}_6^T)[:,2:\min(l+m+1,n)],\\
\tilde{X} &= \begin{bmatrix}\tilde{X}_s\\ \mathbf{0}\end{bmatrix} \in \mathbb{R}^{(n-1)\times p}, \quad \text{with}\\
\tilde{X}_s &= (-\tau\mathbf{b}\bar{\mathbf{w}}_1^T - \tau\mathbf{b}\mathbf{f}^T - \tau\mathbf{b}\mathbf{c}_1^T - \tau\mathbf{b}\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T + X - \tau\mathbf{b}\mathbf{c}_4^T\\
&\quad + Y\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\mathbf{b}\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)[2:\min(l+1,n),:],\\
\tilde{Y} &= \begin{bmatrix}\tilde{Y}_s\\ \mathbf{0}\end{bmatrix} \in \mathbb{R}^{(n-1)\times r}, \quad \text{with}\\
\tilde{Y}_s &= (-\tau\mathbf{b}\bar{\mathbf{k}}^T - \tau\mathbf{b}\mathbf{c}_2^T + Y - \tau\mathbf{b}\mathbf{c}_5^T)[2:\min(l+1,n),:],\\
\tilde{Z} &= \begin{bmatrix}\tilde{Z}_s & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}, \quad \text{with}\\
\tilde{Z}_s &= (-\tau\mathbf{b}\mathbf{d}_1^T - \tau\mathbf{b}\bar{\mathbf{d}}^T - \tau\mathbf{b}\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{b}\mathbf{c}_3^TE + Y\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{b}\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T\\
&\quad + Z - \tau\mathbf{b}\mathbf{c}_6^T)[2:\min(l+1,n), 2:\min(l+m+1,n)].
\end{aligned}$$

The forms of $\tilde{Q}, \tilde{K}, \tilde{E}, \tilde{X}, \tilde{Y}, \tilde{Z}$ confirm that $\tilde{C}[2:n,2:n]$ is an HMBPSM related to $A[2:n,2:n]$, thus establishing the first part of the lemma.

Secondly, the first row of the transformed matrix, $\tilde{C}[1,2:n]$, can be expressed as:

$$\tilde{C}[1,2:n] = \hat{\mathbf{d}}^T + \hat{\boldsymbol{\alpha}}^T(S^T[:,2:n]) + \hat{\boldsymbol{\beta}}^T((U^{(2)T}A)[:,2:n]), \tag{2.18}$$

where

$$\begin{aligned}
\hat{\mathbf{d}} &= \begin{bmatrix}\hat{\mathbf{d}}_s\\ \mathbf{0}\end{bmatrix} \in \mathbb{R}^{n-1}, \quad \text{with}\\
\hat{\mathbf{d}}_s &= (\mathbf{d}_1^T - \tau\mathbf{d}_1^T - \tau\bar{\mathbf{d}}^T + \bar{\mathbf{u}}_1^TK\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{c}_2^T\bar{\mathbf{u}}_1\mathbf{d}_1^T + \bar{\mathbf{u}}_1^TE - \tau\mathbf{c}_3^TE\\
&\quad + \mathbf{y}^{(1)T}\bar{\mathbf{u}}_1\mathbf{d}_1^T - \tau\mathbf{c}_5^T\bar{\mathbf{u}}_1\mathbf{d}_1^T + \mathbf{z}^{(1)T} - \tau\mathbf{c}_6^T)^T[2:\min(l+m+1,n)],\\
\hat{\boldsymbol{\alpha}} &= (\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{w}}_1^T - \tau\mathbf{f}^T + \bar{\mathbf{u}}_1^TQ - \tau\mathbf{c}_1^T + \bar{\mathbf{u}}_1^TK\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\mathbf{c}_2^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T
\end{aligned}$$

$$+\mathbf{x}^{(1)T} - \tau\mathbf{c}_4^T + \mathbf{y}^{(1)T}\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T - \tau\mathbf{c}_5^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)^T \in \mathbb{R}^p,$$

$$\hat{\boldsymbol{\beta}} = (-\tau\bar{\mathbf{k}}^T + \bar{\mathbf{u}}_1^T K - \tau\mathbf{c}_2^T + \mathbf{y}^{(1)T} - \tau\mathbf{c}_5^T)^T \in \mathbb{R}^r.$$

Noting that $U^{(2)} = U - \mathbf{e}_1\bar{\mathbf{u}}_1^T$ and $\mathbf{e}_1^T A = \mathbf{d}_1^T + \bar{\mathbf{w}}_1^T S^T$, we have $U^{(2)T}A = U^T A - \bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T S^T - \bar{\mathbf{u}}_1\mathbf{d}_1^T$. Substituting this yields an alternative expression:

$$\tilde{C}[1, 2:n] = \tilde{\mathbf{d}}^T + \tilde{\boldsymbol{\alpha}}^T(S^T[:, 2:n]) + \tilde{\boldsymbol{\beta}}^T((U^T A)[:, 2:n]), \tag{2.19}$$

where

$$\tilde{\mathbf{d}} = \begin{bmatrix} \tilde{\mathbf{d}}_s \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n-1}, \quad \text{with}$$

$$\tilde{\mathbf{d}}_s = (\mathbf{d}_1^T - \tau\mathbf{d}_1^T - \tau\bar{\mathbf{d}}^T + \bar{\mathbf{u}}_1^T E - \tau\mathbf{c}_3^T E + \mathbf{z}^{(1)T} - \tau\mathbf{c}_6^T + \tau\bar{\mathbf{k}}^T\bar{\mathbf{u}}_1\mathbf{d}_1^T)^T[2:\min(l+m+1, n)],$$

$$\tilde{\boldsymbol{\alpha}} = (\bar{\mathbf{w}}_1^T - \tau\bar{\mathbf{w}}_1^T - \tau\mathbf{f}^T + \bar{\mathbf{u}}_1^T Q - \tau\mathbf{c}_1^T + \mathbf{x}^{(1)T} - \tau\mathbf{c}_4^T + \tau\bar{\mathbf{k}}^T\bar{\mathbf{u}}_1\bar{\mathbf{w}}_1^T)^T \in \mathbb{R}^p,$$

$$\tilde{\boldsymbol{\beta}} = (-\tau\bar{\mathbf{k}}^T + \bar{\mathbf{u}}_1^T K - \tau\mathbf{c}_2^T + \mathbf{y}^{(1)T} - \tau\mathbf{c}_5^T)^T \in \mathbb{R}^r.$$

This confirms that $\tilde{C}[1, 2:n]$ is an HMBPSV related to A, completing the proof of the lemma. ∎

2.3. Main Theorem and its Proof. Equipped with Lemma 2.4, we now state and prove the main theorem concerning the structure of the QR factor matrix F.

THEOREM 2.5. *After applying the QR factorization to a banded-plus-semiseparable matrix A (as expressed in Eq. (2.1)) with lower semiseparable rank r, upper semiseparable rank p, lower bandwidth l, and upper bandwidth m, the resulting factor matrix F is also a banded-plus-semiseparable matrix. Specifically:*

- *Its lower semiseparable part has rank r.*
- *Its upper semiseparable part has rank $r + p$.*
- *Its banded part has lower bandwidth l and upper bandwidth $l + m$.*

Proof. The QR factorization is computed by performing a sequence of $n - 1$ Householder transformations, eliminating the subdiagonal entries of A column by column.

Let $A^{(i)}$ denote the matrix after the i-th Householder transformation, with $A^{(0)} = A$. Define $A_i = A[i:n, i:n]$, $U_i = U[i:n, :]$, $V_i = V[i:n, :]$, $S_i = S[i:n, :]$, and $W_i = W[i:n, :]$. Let $\bar{\mathbf{u}}_i = (u_i^{(1)}, \ldots, u_i^{(r)})^T$ and $\bar{\mathbf{w}}_i = (w_i^{(1)}, \ldots, w_i^{(p)})^T$. Let $\bar{S} = U^T A \in \mathbb{R}^{r \times n}$.

We prove by induction that after the j-th transformation ($0 \le j < n$):

1. The submatrix $A^{(j)}[j+1:n, j+1:n]$ is an HMBPSM related to A_{j+1}.
2. The j-th row of the final factor F, $F[j, j+1:n]$, is an HMBPSV related to A_j.
3. The j-th column of F below the diagonal, $F[j+1:n, j]$, has the form $(U\bar{\mathbf{k}}_{j+1})[j+1:n] + \mathbf{b}_{j+1}[2:n+1-j]$, where $\bar{\mathbf{k}}_{j+1} \in \mathbb{R}^r$ and $\mathbf{b}_{j+1} \in \mathbb{R}^{n+1-j}$ is non-zero only in its first $\min(l+1, n+1-j)$ entries.

Base Case (j=0): Initially, $A^{(0)} = A$ is trivially an HMBPSM (with $Q, K, E, X, Y, Z = \mathbf{0}$) related to $A_1 = A$.

Inductive Step: Assume the induction hypothesis holds for j. That is, $A^{(j)}[j+1:n, j+1:n]$ is an HMBPSM related to A_{j+1}, and for $i = 1, \ldots, j$:

$$F[i+1:n, i] = (U\bar{\mathbf{k}}_{i+1})[i+1:n] + \mathbf{b}_{i+1}[2:n+1-i], \tag{2.20}$$

$$F[i, i+1:n] = \tilde{\mathbf{d}}_{i+1} + (\tilde{\boldsymbol{\alpha}}_{i+1}^T S^T)[i+1:n] + (\tilde{\boldsymbol{\beta}}_{i+1}^T \bar{S})[i+1:n], \tag{2.21}$$

with $\tilde{\boldsymbol{\alpha}}_{i+1} \in \mathbb{R}^p$, $\tilde{\boldsymbol{\beta}}_{i+1} \in \mathbb{R}^r$, and $\tilde{\mathbf{d}}_{i+1} \in \mathbb{R}^{n-i}$ non-zero only in its first $\min(l+m, n-i)$ entries.

Furthermore, assume:

$$\begin{aligned} A^{(j)}[j+1:n, j+1:n] =& A_{j+1} + U_{j+1}Q_{j+1}S_{j+1}^T + U_{j+1}K_{j+1}U_{j+1}^T A_{j+1} \\ &+ U_{j+1}E_{j+1} + X_{j+1}S_{j+1}^T + Y_{j+1}U_{j+1}^T A_{j+1} + Z_{j+1}, \end{aligned} \tag{2.22}$$

where the modification matrices $Q_{j+1}, K_{j+1}, E_{j+1}, X_{j+1}, Y_{j+1}, Z_{j+1}$ possess the sparsity patterns specified in Definition 2.1.

If $j < n-1$, we now perform the $(j+1)$-th Householder transformation on this HMBPSM. Let $\mathbf{y}_{j+1}$ be the corresponding Householder vector(unlike the form given in (2.3), here $\mathbf{y}_{j+1}$ is a vector of length $n-j$ applied to the corresponding submatrix, and we will follow this convention until the end of section 3.1). It can be expressed as $\mathbf{y}_{j+1} = \mathbf{e}_{j+1} + U^{(j+2)}\bar{\mathbf{k}}_{j+2} + \mathbf{b}_{j+2}$, where $\mathbf{e}_{j+1} \in \mathbb{R}^{n-j}$ is the first standard basis vector, $U^{(j+2)} \in \mathbb{R}^{(n-j)\times r}$ satisfies $U^{(j+2)}[1,:] = \mathbf{0}$ and $U^{(j+2)}[2:n-j,:] = U[j+2:n,:]$, $\bar{\mathbf{k}}_{j+2} \in \mathbb{R}^r$, and $\mathbf{b}_{j+2} \in \mathbb{R}^{n-j}$ is non-zero only in its first $\min(l+1, n-j)$ entries. This vector defines the $(j+1)$-th column of F:

$$F[j+2:n, j+1] = (U\bar{\mathbf{k}}_{j+2})[j+2:n] + \mathbf{b}_{j+2}[2:n-j]. \tag{2.23}$$

We now apply Lemma Lemma 2.4 to the HMBPSM $C = A^{(j)}[j+1:n, j+1:n]$, which is related to A_{j+1}. The Householder transformation $(I - \tau_{j+1}\mathbf{y}_{j+1}\mathbf{y}_{j+1}^T)$ is applied to C, here τ_{j+1} is a coefficient found to satisfy the definition of a Householder transformation.

From the lemma, the resulting submatrix $A^{(j+1)}[j+2:n, j+2:n]$ is an HMBPSM related to A_{j+2}. Its structure is given by equations analogous to (2.17), with updated modification matrices $Q_{j+2}, K_{j+2}, E_{j+2}, X_{j+2}, Y_{j+2}, Z_{j+2}$, which retain the required sparsity patterns. This satisfies condition 1 for $j+1$.

Furthermore, the lemma states that $A^{(j+1)}[j+1, j+2:n]$ is an HMBPSV related to A_{j+1}. This row becomes $F[j+1, j+2:n]$ in the final factor matrix. Following the derivation (2.19) in the lemma, and using the relation

$$U_{j+1}^T A_{j+1}[:, 2:n-j] = (U^T A - \sum_{t=1}^{j} \bar{\mathbf{u}}_t \bar{\mathbf{w}}_t^T S^T)[:, j+2:n] - \sum_{t=\max(j-m+2,1)}^{j} \bar{\mathbf{u}}_t (B[t, j+2:n]), \tag{2.24}$$

we can express this row in the form:

$$F[j+1, j+2:n] = \tilde{\mathbf{d}}_{j+2}^T + (\tilde{\boldsymbol{\alpha}}_{j+2}^T S^T)[j+2:n] + (\tilde{\boldsymbol{\beta}}_{j+2}^T \bar{S})[j+2:n], \tag{2.25}$$

where $\tilde{\mathbf{d}}_{j+2}$ only nonzero in the first $\min(l+m, n-j-1)$ entries, $\tilde{\boldsymbol{\alpha}}_{j+2} \in \mathbb{R}^p$, and $\tilde{\boldsymbol{\beta}}_{j+2} \in \mathbb{R}^r$. This satisfies condition 2 for $j+1$. Condition 3 for $j+1$ is already established by (2.23).

By the principle of induction, the hypotheses hold for all $j = 0, \dots, n-1$.

Upon completion of all $n-1$ transformations, the factor matrix F is fully determined. Aggregating the results from (2.20) and (2.21), we conclude that F can be written in the form:

$$F = B_F + \mathrm{tril}(U\bar{K}^T, -1) + \mathrm{triu}([\bar{A}, \bar{B}][S, \bar{S}^T]^T, 1), \tag{2.26}$$

where

- $\bar{K} \in \mathbb{R}^{n\times r}$ is defined by $\bar{K}[i,:] = \bar{\mathbf{k}}_{i+1}^T$ for $i = 1,\dots,n-1$ and $\bar{K}[n,:] = \mathbf{0}$.
- $\bar{A} \in \mathbb{R}^{n\times p}$ is defined by $\bar{A}[i,:] = \boldsymbol{\alpha}_{i+1}^T$ for $i = 1,\dots,n-1$ and $\bar{A}[n,:] = \mathbf{0}$.
- $\bar{B} \in \mathbb{R}^{n\times r}$ is defined by $\bar{B}[i,:] = \boldsymbol{\beta}_{i+1}^T$ for $i = 1,\dots,n-1$ and $\bar{B}[n,:] = \mathbf{0}$.
- B_F is a banded matrix with lower bandwidth l and upper bandwidth $l+m$, defined by:

$$B_F[i,j] = \begin{cases} A^{(i)}[i,i], & i = j < n \\ A^{(n-1)}[n,n], & i = j = n \\ \mathbf{b}_{j+1}[i-j+1], & 0 < i-j \le l \\ \tilde{\mathbf{d}}_{i+1}[j-i], & 0 < j-i \le l+m \\ 0, & \text{otherwise.} \end{cases}$$

The representation in (2.26) explicitly shows that F is a banded-plus-semiseparable matrix with a lower semiseparable rank of r, an upper semiseparable rank of $r+p$, a lower bandwidth of l, and an upper bandwidth of $l+m$. This completes the proof. ∎

3. Algorithms.

3.1. Fast QR factorization for BPS Matrices. Based on the structure-preserving theorem proven in Section 2, we now present the detailed $O(n)$ algorithm for computing the QR factorization of a banded-plus-semiseparable matrix. The algorithm exploits the proven fact that the factor matrix F maintains a BPS structure.

Make an algorithm environment. Possibly split into sub-algorithms

ALGORITHM 3.1 (Fast QR).

This algorithm computes the QR factorization of a BPS matrix $A = B + \mathrm{tril}(UV^T, -1) + \mathrm{triu}(WS^T, 1)$, *producing the structured factor matrix F (which contains both R and the Householder vectors) and the scalar coefficients $\boldsymbol{\tau}$, in $O(n)$ operations.*

Input:

- *B: Banded matrix with lower bandwidth l, upper bandwidth m.*
- *$U, V \in \mathbb{R}^{n\times r}$: Generators for the lower semiseparable part of rank r.*
- *$W, S \in \mathbb{R}^{n\times p}$: Generators for the upper semiseparable part of rank p.*

Output:

- *F: The structured factor matrix, maintained as a BPS matrix with lower rank r, upper rank $r+p$, lower bandwidth l, and upper bandwidth $l+m$. It is stored via its components:*
  - *B_F: The updated banded part.*
  - *$\bar{K} \in \mathbb{R}^{n\times r}$: Generator for the lower semiseparable part of F.*
  - *$\bar{A} \in \mathbb{R}^{n\times p}, \bar{B} \in \mathbb{R}^{n\times r}$: Generators for the upper semiseparable part of F.*
- *$\boldsymbol{\tau} \in \mathbb{R}^n$: The scalar coefficients for the Householder transformations.*

Procedure:

1. ***Initialization:***

- *Set $A^{(0)} = A$. The initial state is an Householder-Modified BPS Matrix with modification matrices Q, K, E, X, Y, Z all set to zero. This corresponds to the original BPS matrix A.*
- *Initialize the output components B_F, $\bar{K}$, $\bar{A}$, $\bar{B}$ to zero matrices of their respective sizes. Also compute $\bar{S} = U^T A$. Note: The matrix $\bar{S}$ can be computed initially in $O(n(r+p))$ time due to the structure of A.*

2. ***For** $k = 1$ **to** $n-1$**, eliminate the subdiagonal entries of the** k**th column:***

We process the submatrix $A^{(k-1)}[k:n,k:n]$, which is an HMBPSM related to $A_k = A[k:n,k:n]$.

(a) Form the Householder vector $\mathbf{y}_{k+1}$:

- *Extract the first column of the current HMBPSM.*
- *According to the inductive proof of Theorem 2.5, the vector $\mathbf{y}_k$ has the specific form:*

$$\mathbf{y}_k = \mathbf{e}_k + U^{(k+1)}\bar{\mathbf{k}}_{k+1} + \mathbf{b}_{k+1}$$

- *Based on the definition of a Householder transformation and the structure of our current HMBPSM, we can obtain $\bar{\mathbf{k}}_{k+1}$, $\mathbf{b}_{k+1}$, τ_k, and the diagonal entry generated in the current column(denoted as $A^{(k)}[k,k]$), in $O(1)$.*
- *Set the k-th component to τ as τ_k*

(b) Store the k-th column of F:

- *The subdiagonal part of this column is given by $\mathbf{y}_k[2:end]$. From its structure, we have:*

$$F[k+1:n,k] = (U\bar{\mathbf{k}}_{k+1})[k+1:n] + \mathbf{b}_{k+1}[2:n-k+1]$$

- *Store $\bar{\mathbf{k}}_{k+1}^T$ as the k-th row of $\bar{K}$.*
- *The vector $\mathbf{b}_{k+1}[2:end]$, which is non-zero only in its first $\min(l,n-k-1)$ entries, is stored in the corresponding subdiagonal positions of the banded part B_F.*
- *Set the digonal part $B_F[k,k]$ to $A^{(k)}[k,k]$, which was obtained in the previous step.*

(c) Compute and store the k-th row of F:

- *This row, $F[k,k+1:n]$, is the first row of the transformed submatrix after the Householder reflection is applied. It is an Householder-Modified BPS Vector:*

$$F[k,k+1:n] = \tilde{\mathbf{d}}_{k+1}^T + (\tilde{\boldsymbol{\alpha}}_{k+1}^T S^T)[k+1:n] + (\tilde{\boldsymbol{\beta}}_{k+1}^T \bar{S})[k+1:n]$$

- *The vectors $\tilde{\boldsymbol{\alpha}}_{k+1} \in \mathbb{R}^p$ and $\tilde{\boldsymbol{\beta}}_{k+1} \in \mathbb{R}^r$ are computed based on the proof in theorem 2.5. Store $\tilde{\boldsymbol{\alpha}}_{k+1}^T$ and $\tilde{\boldsymbol{\beta}}_{k+1}^T$ as the k-th rows of $\bar{A}$ and $\bar{B}$, respectively.*
- *The vector $\tilde{\mathbf{d}}_{k+2}$, which is non-zero only in its first $\min(l+m,n-k)$ entries, is stored in the corresponding superdiagonal entries of the banded part B_F.*

(d) Update the remaining submatrix (Implicit HMBPSM update):

- *Update matrices Q, K, E, X, Y, Z as derived in the proof of Lemma 2.4.*
- *Actually, we only need to store and update the nonzero parts of E, X, Y, Z, which are E_s, X_s, Y_s, Z_s, and they require $O(1)$ storage only.*
- *These updates consist of low-rank operations and manipulations of matrices with limited non-zero rows/columns, which can be done in $O(1)$ time.*

3. ***Final step ($k = n$):***

- *The last diagonal element $F[n,n] = A^{(n-1)}[n,n]$ is simply the last remaining 1-by-1 submatrix after the $n-1$ Householder transformations. Store it in $B_F[n,n]$.*

4. ***Output:***

- *The complete QR factorization is represented by the structured factor matrix F, defined as $B_F + \mathrm{tril}(U\bar{K}^T,-1) + \mathrm{triu}([\bar{A},\bar{B}][S,\bar{S}^T]^T,1)$, and the vector $\boldsymbol{\tau}$.*

Complexity Analysis. The algorithm runs for $n-1$ steps. The cost per step can be expressed as a polynomial in term of r, p, l, and m. Since these are constants independent of n, the total complexity is $O(n)$. The memory footprint is also $O(n)$, as we store only the generators and banded components.

REMARK 3.1. *To maintain the $O(1)$ per-step complexity in Algorithm 3.1, two key quantities must be computed efficiently during the Householder updates:*

- ***Inner product matrix*** $U_k^T U_k$*: The computation of intermediate vectors* $\mathbf{c}_1, \dots, \mathbf{c}_6$ *requires evaluating expressions like* $U_k^T \mathbf{y}_k = U_k^T(\mathbf{e}_k + U^{(k+1)}\bar{\mathbf{k}}_{k+1} + \mathbf{b}_{k+1})$*, which involves* $U_k^T U^{(k+1)}$ *that is equal to* $U_{k+1}^T U_{k+1}$*. we precompute a lookup table:*

$$\mathit{UU\_lookup}[k] = U[k:n,:]^T U[k:n,:] \quad \textit{for } k = 1, \dots, n$$

  This can be computed in $O(nr^2)$ time via a backward accumulation.

- ***Partial sum*** $\sum_{t=1}^{j} \bar{\mathbf{u}}_t \bar{\mathbf{w}}_t^T$*: The update of the upper triangular part in equation (2.24) requires this sum. We precompute:*

$$\mathit{UV\_lookup}[j] = \sum_{t=1}^{j} \bar{\mathbf{u}}_t \bar{\mathbf{w}}_t^T \quad \textit{for } j = 1, \dots, n-1$$

  This is computed in $O(nrp)$ time via forward accumulation.

Both precomputations require $O(n)$ total time and enable $O(1)$ access to the required quantities at each step of the factorization, thus preserving the overall $O(n)$ complexity.

3.2. Fast Solver for BPS Matrices.

Theorem 2.5 not only enables an efficient QR factorization but also facilitates a complete direct solver for linear systems of the form $A\mathbf{x} = \mathbf{b}$, where A is a banded-plus-semiseparable matrix. The solver consists of two phases after the QR factorization $A = QR$: 1. Application of Q^T to the right-hand side vector $\mathbf{b}$ to form $\mathbf{c} = Q^T\mathbf{b}$. 2. Solution of the upper triangular system $R\mathbf{x} = \mathbf{c}$ via backward substitution.

We now present $O(n)$ algorithms for both phases, leveraging the structured representation of the factorization output by Algorithm 3.1.

3.2.1. Fast Application of Q^T.

The orthogonal matrix Q is represented as a product of Householder transformations:

$$Q = (I - \tau_1 \mathbf{y}_1 \mathbf{y}_1^T)(I - \tau_2 \mathbf{y}_2 \mathbf{y}_2^T) \cdots (I - \tau_{n-1} \mathbf{y}_{n-1} \mathbf{y}_{n-1}^T).$$

Applying Q^T to a vector $\mathbf{b}$ thus requires computing:

$$Q^T\mathbf{b} = (I - \tau_{n-1} \mathbf{y}_{n-1} \mathbf{y}_{n-1}^T) \cdots (I - \tau_2 \mathbf{y}_2 \mathbf{y}_2^T)(I - \tau_1 \mathbf{y}_1 \mathbf{y}_1^T)\mathbf{b}.$$

The Householder vectors $\mathbf{y}_k$ are stored in the factor matrix F according to the normalization convention established in Section 2:

$$\mathbf{y}_k[j] = \begin{cases} 0, & j < k \\ 1, & j = k \\ F[j,k], & j > k \end{cases} \quad \text{for } k = 1, \dots, n-1.$$

From Theorem 2.5, the factor matrix F admits the BPS representation:

$$F = B_F + \operatorname{tril}(U_F V_F^T, -1) + \operatorname{triu}(W_F S_F^T, 1), \tag{3.1}$$

where $U_F, V_F \in \mathbb{R}^{n \times r}$, $W_F, S_F \in \mathbb{R}^{n \times (r+p)}$, and B_F is banded with lower bandwidth l and upper bandwidth $l + m$.

This structure implies that each Householder vector $\mathbf{y}_k$ can be expressed as:

$$\mathbf{y}_k = \bar{\mathbf{e}}_k + U_F^{(k+1)} \bar{\mathbf{v}}_k + \mathbf{d}_k, \tag{3.2}$$

where:

- $\bar{\mathbf{e}}_k \in \mathbb{R}^n$ is the k-th standard basis vector,
- $U_F^{(k+1)} \in \mathbb{R}^{n\times r}$ satisfies $U_F^{(k+1)}[1:k,:] = \mathbf{0}$ and $U_F^{(k+1)}[k+1:n,:] = U_F[k+1:n,:]$,
- $\bar{\mathbf{v}}_k = V_F[k,:]^T \in \mathbb{R}^r$,
- $\mathbf{d}_k \in \mathbb{R}^n$ is non-zero only in positions $k+1$ to $\min(k+l,n)$, with $\mathbf{d}_k[j] = B_F[j,k]$ for $j = k+1,\dots,\min(k+l,n)$.

Algorithm 3.2.1 exploits this structure to compute $Q^T\mathbf{b}$ in $O(n)$ operations by maintaining a compressed representation of the intermediate vectors throughout the transformation process.

Algorithm 3.2.1 Fast Application of Q^T to a Vector

Input: Factor matrix F in BPS form: $F = B_F + \mathrm{tril}(U_F V_F^T, -1) + \mathrm{triu}(W_F S_F^T, 1)$; coefficient vector $\boldsymbol{\tau} = [\tau_1,\dots,\tau_{n-1},0]^T \in \mathbb{R}^n$; right-hand side vector $\mathbf{b} \in \mathbb{R}^n$
Output: $\mathbf{c} = Q^T\mathbf{b} \in \mathbb{R}^n$
Initialize:
- $O \leftarrow \mathbf{0}_{n\times r}$: Storage for accumulated low-rank updates
- $G \leftarrow \mathbf{0}_{n\times(l+1)}$: Storage for banded component updates
- $\mathbf{h} \leftarrow \mathbf{0}_r$: Accumulator for semiseparable component
- Let $\mathbf{o}_i$ denote the i-th column of O
- Let $\mathbf{g}_i$ denote the i-th column of G

Express initial vector: $\mathbf{b}^{(0)} = \mathbf{b} + U_F^{(1)}\mathbf{h} + \sum_{i=1}^r \mathbf{o}_i + \sum_{i=1}^{l+1} \mathbf{g}_i$
for $k = 1$ to $n-1$ **do**
  Compute inner product: $c \leftarrow \mathbf{y}_k^T\mathbf{b}^{(k-1)}$ (exploit BPS structure of $\mathbf{y}_k$ and precompute some lookup tables for for $O(1)$ computation)
  Update low-rank storage: $O[k,:] \leftarrow U_F[k,:] \odot \mathbf{h}^T$ (element-wise multiplication)
  Update semiseparable accumulator: $\mathbf{h} \leftarrow \mathbf{h} - \tau_k c \cdot V_F[k,:]^T$
  Update banded component:
  $G[k,1] \leftarrow -\tau_k c$ (diagonal contribution)
  for $t = 1$ to $\min(l, n-k)$ **do**
    $G[k+t, t+1] \leftarrow -\tau_k c \cdot B_F[k+t,k]$ (subdiagonal contributions)
  end for
  Current representation: $\mathbf{b}^{(k)} = \mathbf{b} + U_F^{(k+1)}\mathbf{h} + \sum_{i=1}^r \mathbf{o}_i + \sum_{i=1}^{l+1} \mathbf{g}_i$
end for
Compute final result explicitly: $\mathbf{c} \leftarrow \mathbf{b} + U_F^{(n)}\mathbf{h} + \sum_{i=1}^r \mathbf{o}_i + \sum_{i=1}^{l+1} \mathbf{g}_i$
return c

THEOREM 3.1. *Algorithm 3.2.1 correctly computes $\mathbf{c} = Q^T\mathbf{b}$ in $O(n)$ operations.*

Proof. The proof proceeds by induction on the transformation steps. Let $\mathbf{b}^{(0)} = \mathbf{b}$ and assume that after $k-1$ steps, the algorithm maintains the representation:

$$\mathbf{b}^{(k-1)} = \mathbf{b} + U_F^{(k)}\mathbf{h}^{(k-1)} + \sum_{i=1}^{r} \mathbf{o}_i^{(k-1)} + \sum_{i=1}^{l+1} \mathbf{g}_i^{(k-1)},$$

where the superscripts on $\mathbf{h}$, $\mathbf{o}$, and $\mathbf{g}$ denote the state after the $(k-1)$-th iteration.

The k-th Householder transformation gives:

$$\mathbf{b}^{(k)} = (I - \tau_k\mathbf{y}_k\mathbf{y}_k^T)\mathbf{b}^{(k-1)} = \mathbf{b}^{(k-1)} - \tau_k(\mathbf{y}_k^T\mathbf{b}^{(k-1)})\mathbf{y}_k.$$

Substituting the structured form of $\mathbf{y}_k$ from (3.2) and the inductive representation:

$$\begin{aligned}\mathbf{b}^{(k)} &= \mathbf{b} + U_F^{(k)}\mathbf{h}^{(k-1)} + \sum_{i=1}^{r}\mathbf{o}_i^{(k-1)} + \sum_{i=1}^{l+1}\mathbf{g}_i^{(k-1)} \\ &\quad - \tau_k c(\bar{\mathbf{e}}_k + U_F^{(k+1)}\bar{\mathbf{v}}_k + \mathbf{d}_k) \\ &= \mathbf{b} + U_F^{(k+1)}(\mathbf{h}^{(k-1)} - \tau_k c\bar{\mathbf{v}}_k) \\ &\quad + \left(\sum_{i=1}^{r}\mathbf{o}_i^{(k-1)} + (U_F^{(k)} - U_F^{(k+1)})\mathbf{h}^{(k-1)}\right) \\ &\quad + \left(\mathbf{g}_1^{(k-1)} - \tau_k c\bar{\mathbf{e}}_k\right) + \left(\sum_{i=2}^{l+1}\mathbf{g}_i^{(k-1)} - \tau_k c\mathbf{d}_k\right).\end{aligned}$$

The algorithm updates precisely these components:
- $\mathbf{h}^{(k)} = \mathbf{h}^{(k-1)} - \tau_k c\bar{\mathbf{v}}_k$,
- $O[k,:] = U_F[k,:] \odot \mathbf{h}^{(k-1)T}$ captures $(U_F^{(k)} - U_F^{(k+1)})\mathbf{h}^{(k-1)}$,
- Banded updates in G capture the remaining terms.

Thus, the representation is maintained correctly throughout all $n-1$ steps. Each step requires $O(1)$ operations due to the constant-bounded parameters r, p, l, m, yielding overall $O(n)$ complexity. ∎

3.2.2. Fast Backward Substitution. After computing $\mathbf{c} = Q^T\mathbf{b}$, we solve the upper triangular system $R\mathbf{x} = \mathbf{c}$, where $R = \mathrm{triu}(F)$ inherits the BPS structure of F. Specifically, the upper triangular part of F satisfies:

$$R = B_R + \mathrm{triu}(W_F S_F^T, 1),$$

where $B_R = \mathrm{triu}(B_F)$ is the upper triangular part of the banded component, maintaining upper bandwidth $l+m$.

Algorithm 3.2.2, which is equivalent to the one introduced in [15], exploits this structure to perform backward substitution in $O(n)$ operations by maintaining a running sum for the semiseparable contributions.

THEOREM 3.2. *Algorithm 3.2.2 solves $R\mathbf{x} = \mathbf{c}$ in $O(n)$ operations.*

Proof. For completeness we include the proof from [15]. The algorithm implements standard backward substitution while exploiting the structure of R. For each index j from n down to 1, the equation:

$$R[j,j]x_j + \sum_{k=j+1}^{n} R[j,k]x_k = c_j$$

is solved for x_j.

The key insight is that the off-diagonal entries $R[j,k]$ for $k > j$ can be decomposed as:

$$R[j,k] = B_R[j,k] + W_F[j,:]\cdot S_F[k,:]^T.$$

The banded contributions $B_R[j,k]$ are non-zero only for $k = j+1,\ldots,\min(j+l+m,n)$, requiring $O(1)$ operations per row. The semiseparable contributions are accumulated in the vector $\mathbf{s}$, which stores:

$$\mathbf{s} = \sum_{i=j+1}^{n} S_F[i,:]^T x_i.$$

Algorithm 3.2.2 Fast Backward Substitution for Structured R

```
Input: Upper triangular matrix R = triu(F) in structured form; transformed right-hand side c ∈ ℝ^n
Output: Solution x ∈ ℝ^n satisfying Rx = c
Initialize:
    • x ← 0_n: solution vector
    • s ← 0_{r+p}: Accumulator for semiseparable contributions
for j = n down to 1 do
    Initialize residual: res ← 0
    Add semiseparable contribution: res ← res + W_F[j, :] · s
    Add banded contributions:
    for k = j + 1 to min(j + l + m, n) do
        res ← res + B_R[j, k] · x[k]
    end for
    Solve for x_j: x[j] ← (c[j] − res)/B_R[j, j]
    Update semiseparable accumulator: s ← s + S_F[j, :]^T · x[j]
end for
return x
```

product? I
⊤ to ·

At step j, the product $W_F[j,:] \cdot \mathbf{s}$ thus captures all semiseparable contributions from previously computed solution components. After computing x_j, the accumulator is updated to include its contribution.

Each iteration requires $O(1)$ operations, yielding overall $O(n)$ complexity. The correctness follows by induction from $j = n$ down to 1. ∎

3.2.3. Overall Solver Complexity. Combining the QR factorization (Algorithm 3.1), the fast application of Q^T (Algorithm 3.2.1), and the fast backward substitution (Algorithm 3.2.2) yields a complete direct solver for BPS linear systems with $O(n)$ complexity.

COROLLARY 3.3. *For a banded-plus-semiseparable matrix $A \in \mathbb{R}^{n\times n}$ with constant-bounded ranks and bandwidths, the linear system $A\mathbf{x} = \mathbf{b}$ can be solved in $O(n)$ operations using the QR-based approach.*

Proof. Algorithm 3.1 computes the QR factorization in $O(n)$ operations. Algorithm 3.2.1 applies Q^T in $O(n)$ operations. Algorithm 3.2.2 solves the triangular system in $O(n)$ operations. The overall complexity is therefore linear in the problem size n. ∎

4. Numerical results. To validate the theoretical complexity and demonstrate the practical efficiency of our proposed algorithms, we implemented the fast QR factorization and the complete linear solver in Julia. The implementation is publicly available in the SemiseparableMatrices.jl package[1], providing an open-source resource for the scientific computing community. All numerical tests use banded-plus-semiseparable matrices with fixed structural parameters $l = 4$, $m = 5$, $r = 2$, $p = 3$ to isolate the scaling behavior with respect to the matrix size n. Computations were carried out on a MacBook Air equipped with an Apple M2 chip (8-core CPU, 8 GB RAM), without GPU acceleration or access to external computing resources.

Make this a citation

[1] https://github.com/JuliaLinearAlgebra/SemiseparableMatrices.jl

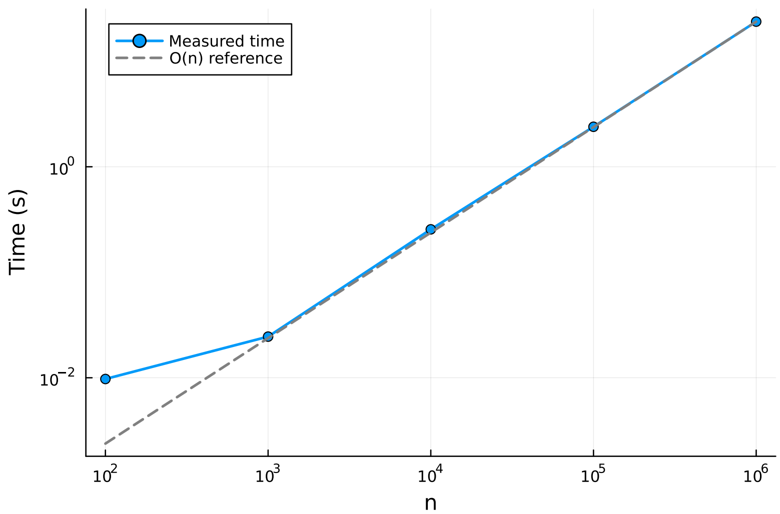

FIG. 1. *Log-log plot of the total solver time (QR factorization + application of Q^T + backward substitution) versus matrix size n. The dashed reference line has slope 1, indicating ideal linear scaling.*

Add different parameters for ranks and bands

Add more ticks on the y-axis

4.1. Linear Complexity Verification. Figure 1 demonstrates the linear time complexity of our complete solver for banded-plus-semiseparable linear systems. The total execution time, encompassing all three phases (QR factorization, application of Q^T, and backward substitution), scales as $O(n)$ across five orders of magnitude, from $n = 100$ to $n = 10^6$. The close alignment with the reference line of slope 1 confirms the complexity analysis in Section 3.

4.2. Comparison with HODLR QR. We compare our fast QR factorization against the state-of-the-art HODLR (Hierarchically Off-Diagonal Low-Rank) QR implementation from the hm-toolbox [13]. The hm-toolbox provides efficient MATLAB routines for various structured matrices, including HODLR and HSS matrices, and represents one of the most mature implementations for hierarchical matrix computations.

Figure 2 shows the execution times for QR factorization of BPS matrices using both approaches. Our algorithm demonstrates superior scaling for larger matrix sizes. This performance advantage stems from several factors:

- **Specialized structure exploitation**: Our algorithm is specifically designed for the banded-plus-semiseparable structure, avoiding the overhead of general hierarchical representations.
- **Reduced Computational Overhead**: By working directly with the semiseparable generators rather than building a hierarchical representation, we avoid the logarithmic factors inherent in tree-based approaches.

The performance gap widens with increasing n, confirming that our method is particularly well-suited for large-scale problems. For $n = 150,000$, our implementation achieves approximately $7\times$ speedup over the HODLR approach, demonstrating the practical benefits of our specialized algorithm.

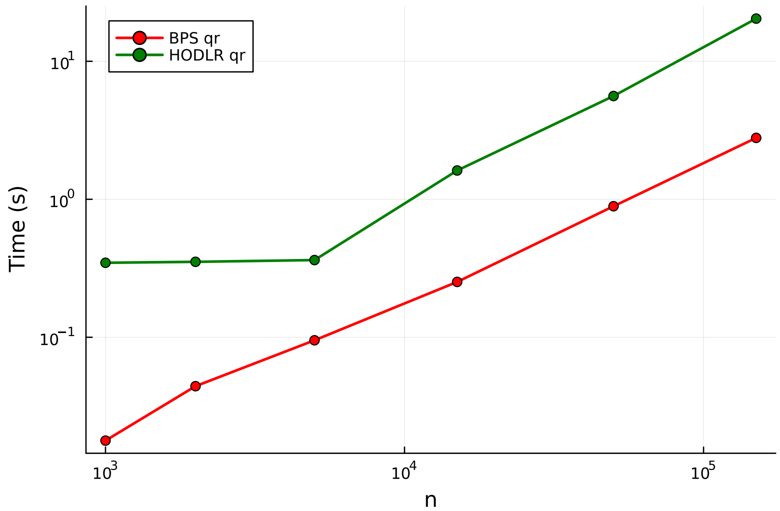

FIG. 2. *Comparison of QR factorization times between our fast BPS QR algorithm and the HODLR QR implementation from [13]. Both algorithms operate on banded-plus-semiseparable matrices with parameters $l = 4$, $m = 5$, $r = 2$, $p = 3$.*

Use different markers so clearer when black-and-white.

Go to $n = 10^6$

5. Conclusions. In this paper, we have established a fundamental theoretical result for BPS matrices and developed efficient algorithms based on this foundation. Our main contribution is the proof that the QR factorization of a BPS matrix preserves the banded-plus-semiseparable structure, with precisely characterized ranks and bandwidths in the resulting factor matrix. This theoretical insight enabled the design of a complete $O(n)$ direct solver for BPS linear systems, comprising:

- A structure-preserving QR factorization algorithm (Algorithm 3.1)
- An efficient $O(n)$ application of Q^T (Algorithm 3.2.1)
- A fast backward substitution routine (Algorithm 3.2.2)

The numerical experiments confirm the linear scaling of our approach and demonstrate significant performance advantages over existing HODLR-based methods. Our implementation in the SemiseparableMatrices.jl package provides the scientific computing community with efficient, open-source tools for working with this important class of structured matrices.

Future Work. A compelling extension involves applying our methodology to specific blocked banded matrices arising in *hp*-FEM [12]. These have optimal complexity so-called reverse Cholesky factorizations (Cholesky from the bottom right instead of the top left) for positive definite problems. One of our motativations for the present work is developing an optimal complexity QL factorization for these special block banded matrices. The key challenge is generalizing our framework to *block* banded-plus-semiseparable matrices while maintaining $O(N)$ complexity. The primary difficulty lies in applying Householder transformations from one block to subsequent blocks in $O(n)$ time(where n is block size and N the total size), rather than $O(n^3)$. While our current framework doesn't directly apply, the core insight of structure preservation provides a promising foundation for this challenging extension.

Fix capitalization

REFERENCES

[1] S. Chandrasekaran, P. Dewilde, M. Gu, W. Lyons, and T. Pals, *A fast solver for hss representations via sparse matrices*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 67–81.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast stable solver for sequentially semi-separable linear systems of equations*, in International Conference on High-Performance Computing, Springer, 2002, pp. 545–554.

[3] S. Chandrasekaran and M. Gu, *Fast and stable algorithms for banded plus semiseparable systems of linear equations*, SIAM Journal on Matrix Analysis and Applications, 25 (2003), pp. 373–384.

[4] S. Chandrasekaran, M. Gu, and T. Pals, *A fast ulv decomposition solver for hierarchically semiseparable representations*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 603–622.

[5] S. Delvaux and M. Van Barel, *Rank structures preserved by the qr-algorithm: the singular case*, Journal of Computational and Applied Mathematics, 189 (2006), pp. 157–178.

[6] S. Delvaux and M. Van Barel, *Structures preserved by the qr-algorithm*, Journal of Computational and Applied Mathematics, 187 (2006), pp. 29–40.

[7] S. Delvaux and M. Van Barel, *A givens-weight representation for rank structured matrices*, SIAM Journal on Matrix Analysis and Applications, 29 (2008), pp. 1147–1170.

[8] S. Delvaux and M. Van Barel, *A qr-based solver for rank structured matrices*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 464–490.

[9] Y. Eidelman and I. Gohberg, *Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices*, Computers & Mathematics with Applications, 33 (1997), pp. 69–79.

[10] Y. Eidelman, I. Gohberg, and V. Olshevsky, *The qr iteration method for hermitian quasiseparable matrices of an arbitrary order*, Linear Algebra and its Applications, 404 (2005), pp. 305–324.

[11] D. Fasino, *Rational krylov matrices and qr steps on hermitian diagonal-plus-semiseparable matrices*, Numerical linear algebra with applications, 12 (2005), pp. 743–754.

[12] K. Knook, S. Olver, and I. Papadopoulos, *Quasi-optimal complexity hp-fem for poisson on a rectangle*, arXiv preprint arXiv:2402.11299, (2024).

[13] S. Massei, L. Robol, and D. Kressner, *hm-toolbox: Matlab software for hodlr and hss matrices*, SIAM Journal on Scientific Computing, 42 (2020), pp. C43–C68.

[14] N. Mastronardi, S. Chandrasekaran, and S. Van Huffel, *Fast and stable two-way algorithm for diagonal plus semi-separable systems of linear equations*, Numerical linear algebra with applications, 8 (2001), pp. 7–12.

[15] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, siam REVIEW, 55 (2013), pp. 462–489.

[16] E. Van Camp, N. Mastronardi, and M. Van Barel, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, Journal of Computational and Applied Mathematics, 164 (2004), pp. 731–747.

[17] R. Vandebril, M. Van Barel, and N. Mastronardi, *An implicit qr algorithm for symmetric semiseparable matrices*, Numerical Linear Algebra with Applications, 12 (2005), pp. 625–658.

[18] R. Vandebril, M. Van Barel, and N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Numerical Linear Algebra with Applications, 12 (2005), pp. 839–858.

[19] R. Vandebril, M. Van Barel, and N. Mastronardi, *Rational qr-iteration without inversion*, Numerische Mathematik, 110 (2008), pp. 561–575.

[20] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, *Fast algorithms for hierarchically semiseparable matrices*, Numerical Linear Algebra with Applications, 17 (2010), pp. 953–976.