Linear Algebra and its Applications xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Linear Algebra and its Applications

journal homepage: www.elsevier.com/locate/laa

Stable solutions of linear systems involving long chain of matrix multiplications☆

Zhaojun Bai [a,*], Che-Rung Lee [b], Ren-Cang Li [c], Shufang Xu [d]

[a] Department of Computer Science, University of California, Davis, CA 95616, USA
[b] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[c] Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019, USA
[d] School of Mathematical Sciences, Peking University, Beijing 100871, PR China

ARTICLE INFO

Article history:
Available online xxxx

Submitted by V. Mehrmann

Dedicated to Prof. G.W. Stewart on the occasion of his 70th birthday

AMS classification:
15A09
15A12
65F05
65F35
82B80

Keywords:
Graded QR decomposition
Singular value decomposition
Condition number
Numerical stability
Quantum Monte Carlo

ABSTRACT

This paper is concerned with solving linear system $(I_n + B_L \cdots B_2 B_1)x = b$ arising from the Green's function calculation in the quantum Monte Carlo simulation of interacting electrons. The order of the system and integer L are adjustable. Also adjustable is the conditioning of the coefficient matrix to give rise an extreme ill-conditioned system. Two numerical methods based on the QR decomposition with column pivoting and the singular value decomposition, respectively, are studied in this paper. It is proved that the computed solution $\widetilde{x}$ by each of the methods is *weakly backward stable* in the sense that the computed $\widetilde{x}$ is close to the exact solution of a nearby linear system

$$[I_n + (B_L + \Delta B_L) \cdots (B_2 + \Delta B_2)(B_1 + \Delta B_1)]\widetilde{x} = b$$

with each ΔB_i small in norm relatively to B_i.

☆ Part of this work was completed while this author was visiting Department of Mathematics, University of Texas at Arlington.

* Corresponding author.
E-mail addresses: bai@cs.ucdavis.edu (Z. Bai), cherung@gmail.com (C. Lee), rcli@uta.edu (R.-C. Li), xsf@math.pku.edu.cn (S. Xu).

0024-3795/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2010.06.023

1. Introduction

We are concerned with numerically solving the following linear system of equations involving a long chain of matrix multiplication:

$$(I_n + B_L \cdots B_2 B_1)x = b, \tag{1.1}$$

where each B_i is $n \times n$, L is an integer, and I_n is the $n \times n$ identity matrix. The linear system of the form (1.1) arises from the quantum Monte Carlo (QMC) simulation of interacting electrons in condensed-matter physics [1–3,13,15]. In the QMC simulation, matrices B_i depend on several parameters which, along with n and L, can be adjusted to give linear systems of any sizes, any number of B_i's, and any difficulty in terms of the condition number

$$\kappa(I_n + B_L \cdots B_2 B_1) \equiv \|I_n + B_L \cdots B_2 B_1\| \left\|(I_n + B_L \cdots B_2 B_1)^{-1}\right\|$$

being arbitrarily large, where $\|\cdot\|$ is a matrix norm. In view of this fact, getting accurate solutions by conventional means, e.g., first forming $I_n + B_L \cdots B_2 B_1$ and then factorizing it, is very difficult, if at all possible. The standard perturbation theory for linear systems suggests that the computed solution $\widetilde{x}$ would be contaminated with a relative error in the order of $\epsilon_{\mathsf{m}}\,\kappa(I_n + B_L \cdots B_2 B_1)$, i.e.,

$$\frac{\|\widetilde{x} - x\|}{\|x\|} = \mathcal{O}(\epsilon_{\mathsf{m}}\,\kappa(I_n + B_L \cdots B_2 B_1)),$$

where ϵ_{m} is the machine unit roundoff ($\epsilon_{\mathsf{m}} = 2^{-24}$ for IEEE single precision and 2^{-53} for IEEE double precision); see for example [5,18,11]. Since the quantity $\epsilon_{\mathsf{m}}\,\kappa(I_n + B_L \cdots B_2 B_1)$ can easily be 1 or larger, it means potentially that the computed $\widetilde{x}$ has no correct significant digits at all. Therefore different methods are called for in order to solve (1.1).

In this paper, we will study two numerical methods to meet the challenge. One is based on the QR decomposition (QRD) with column pivoting and the other is based on the singular value decomposition (SVD). The first one is based on the current practice by computational physicists in the field [12,13] with modifications. The second method replaces all QR decompositions by singular value decompositions to take advantage of highly accurate one-sided Jacobi SVD algorithms [6–9]. Our error analysis shows that the computed solution by either method is *weakly backward stable*, namely it is close to the exact solution of a nearby system of (1.1), a weaker statement than saying the methods are backward stable. The essence of the first method is about how to accurately compute the *graded* QR decompositions of the product of matrices $B_L \cdots B_2 B_1$ before solving the linear system (1.1) for x. In this sense, it is a classical matrix computational problem and has been studied by Stewart [16] and others, see [20] and references therein.

This paper focus on the real case only, i.e., B_i and b are real. This is because the linear systems (1.1) from Hubbard quantum Monte Carlo simulation in condensed matter physics, which motivate our investigation here in the first place, are real. Our presentation can be straightforwardly modified to deal with the complex case.

The rest of this paper is organized as follows. In Section 2, we discuss two schemes to transform the linear system (1.1) to better-conditioned systems via QRD and SVD, and present the resulting numerical algorithms. Error analysis of two methods are presented in Section 3. Numerical examples and concluding remarks are given in Sections 4 and 5, respectively.

We will adopt a MATLAB-like convention to access a matrix's row, column, and diagonal: $X_{(i,:)}$ and $X_{(:,j)}$ are X's ith row and jth column, respectively, and $\mathrm{diag}(X)$ is the diagonal matrix having the same diagonal entries as X, and $u_{(i)}$ is the ith entry of the column vector u. X^{T} is X's transpose, $|X|$ takes entry-wise absolute values, and $|X| \leqslant |Y|$ is understood entry-wisely. For $1 \leqslant p \leqslant \infty$, the ℓ_p vector norm of a column vector u and the ℓ_p-operator norm of a matrix X are defined as

$$\|u\|_p = \left(\sum_i |u_{(i)}|^p\right)^{1/p}, \quad \|X\|_p = \max_u \frac{\|Xu\|_p}{\|u\|_p}.$$

When X is invertible, we define $\kappa_p(X) = \|X\|_p \|X^{-1}\|_p$, the ℓ_p-condition number of X.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

2. Transforming to better-conditioned systems

We have pointed out that linear system (1.1) is often very ill-conditioned. Naturally one would attempt to improve its conditioning by certain equivalent transformations. In this section, we shall present two ways to do the transformations: via QRD with column pivoting [4, p. 103], [10, p. 248], [17, p. 370], or via SVD by the one-sided Jacobi method [6–9]. The former is faster, and the latter is provably more robust as our later error analysis will show. The approach via QRD is being used in [12]. What distinguishes ours here from theirs is that we do one more step beyond their equivalent linear system to arrive at a well-condition one in the sense that the condition number of our final transformed system is usually modest.

2.1. Via QRD

Let

$$B_1 = Q_1 R_1 P_1$$

be B_1's QRD with column pivoting, Q_1 is orthogonal, R_1 is upper triangular, and P_1 is the permutation matrix as the result of the column pivoting. While offering no guarantee in general, the diagonal entries of the R-factor in QRD with column pivoting often reflects the singular value magnitudes well in practice. Now set[1]

$$D_1 = \mathrm{diag}(\|(R_1)_{(i,:)}\|_p), \quad T_1 = D_1^{-1} R_1 P_1 \tag{2.1}$$

we have

$$B_1 = Q_1 R_1 P_1 = Q_1 D_1 \left(D_1^{-1} R_1 P_1\right) = Q_1 D_1 T_1. \tag{2.2}$$

This pick of D_1 serves two purposes: to make T_1 well-conditioned (as much as possible and yet numerically cheap to do) and to make $\|T_1\|_p$ of $\mathcal{O}(1)$. The need to have $\|T_1\|_p$ of $\mathcal{O}(1)$ shows up later in our forward error bound for the computed solution of the transformed linear system. There is no need to have a well-conditioned T_1 as a whole, but rather that the first many rows of T_1 must be well-conditioned as we shall explain later in Remark 2.1. Exactly how many first rows of T_1 are needed to be so depends, but making whole T_1 well-conditioned will make sure the well-conditionedness of any number of rows of T_1. A theorem of van der Sluis [19] (see also [11, p. 125]) guarantees that with this D_1, T_1 is nearly optimally conditioned among all possible diagonal matrices in the sense that

$$\kappa_p(T_1) \leqslant n^{1/p} \min_{\text{diagonal } D} \kappa_p(D^{-1} R_1 P_1).$$

Now for j from 2 to L, perform QRD with column pivoting on $B_j Q_{j-1} D_{j-1}$ to get

$$(B_j Q_{j-1}) D_{j-1} = Q_j R_j P_j = Q_j D_j \left(D_j^{-1} R_j P_j\right) \equiv Q_j D_j T_j, \tag{2.3}$$

where

$$D_j = \mathrm{diag}(\|(R_j)_{(i,:)}\|_p), \quad T_j = D_j^{-1} R_j P_j. \tag{2.4}$$

Here the parentheses in $(B_j Q_{j-1}) D_{j-1}$ must be respected. The pick of D_j in (2.4) serves the same two purposes as D_1 does before. It follows from (2.2) and (2.3) that

$$B_L \cdots B_2 B_1 = Q_L D_L (T_L \cdots T_2 T_1), \tag{2.5}$$

and finally the linear system (1.1) is transformed into

$$[I_n + Q_L D_L (T_L \cdots T_2 T_1)] x = b. \tag{2.6}$$

[1] Here we present the transformation in any given p, $1 \leqslant p \leqslant \infty$, in an effort to be a little bit more general. Practically, p is likely to be 1, 2, or ∞. In fact, it takes the least effort to extract D_j in (2.1) and (2.4) below when $p = 1$ because then $D_j = \mathrm{diag}(R_j)$. The ℓ_p norm of a row vector should be understood by regarding the row vector as a matrix with 1 row, and the definition of the ℓ_p-operator norm of a matrix applies.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

The decomposition (2.5) is referred to as a *column-stratified matrix decomposition* in [13] since the diagonal entries of D_L are typically ordered in their magnitudes from the largest to smallest, while $T_L \cdots T_2 T_1$ is modestly well-conditioned.

Up to this point, we are doing exactly what have been done in [12], namely obtaining (2.6). In [12], it further rewrites the system (2.6) as

$$\left[Q_L^{\mathrm{T}}(T_L \cdots T_2 T_1)^{-1} + D_L\right] y = Q_L^{\mathrm{T}} b \quad \text{and} \quad (T_L \cdots T_2 T_1)x = y. \tag{2.7}$$

Thus two linear systems need to be solved. The standard perturbation theory for linear systems suggests the computed solution could suffer from a relative error as much as

$$\mathcal{O}\left(\epsilon_{\mathrm{m}} \kappa_2 \left(Q_L^{\mathrm{T}} T^{-1} + D_L\right) \times \kappa_2(T)\right), \tag{2.8}$$

where $T = T_L \cdots T_2 T_1$. This is bad news because while $\kappa_2(T)$ appears to be under control (still it can be nontrivial such as in the order of thousands), $\kappa_2(Q_L^{\mathrm{T}} T^{-1} + D_L)$ is comparable to $\kappa_2(I_n + B_L \cdots B_2 B_1)$. This seems that no better solution can be gotten this way than to solve the original system (1.1) by any conventional approach. However, computational physicists have been doing it in this way and getting numerical results that conform to the underlying physics more often than not, but no theoretical analysis has been done to show whether the current practice works or otherwise [13–15]. What makes this method right in this case? One plausible explanation for this discrepancy between theory and practice may be the following. We may safely assume that $Q_L^{\mathrm{T}} T^{-1}$ has modest magnitude and condition numbers. Since the first many diagonal entries of D_L are typically huge, the first many rows of $Q_L^{\mathrm{T}} T^{-1} + D_L$ are diagonally dominant, and in fact these rows are pretty much a diagonal matrix appended by a zero block to its right. Therefore, the relative error in the computed y is proportional to the condition number of the remaining rows. So roughly speaking, $\kappa_2(Q_L^{\mathrm{T}} T^{-1} + D_L)$ can be effectively reduced to the condition number of the remaining rows which is much smaller. But to put this explanation into precise mathematical statement can be necessarily messy and complicated. Fortunately there is a better approach to solve (2.6) which we will be proposing. It will lead to a more accurate numerical solution.

In view of the wide magnitudes of D_L's diagonal entries, care must be taken. For this purpose, we define two $n \times n$ diagonal matrices D_{b} and D_{s} as follows: for $1 \leqslant i \leqslant n$

$$(D_{\mathrm{b}})_{(i,i)} = \begin{cases} (D_L)_{(i,i)} & \text{if } |(D_L)_{(i,i)}| > 1, \\ 1 & \text{otherwise,} \end{cases} \tag{2.9}$$

and

$$(D_{\mathrm{s}})_{(i,i)} = \begin{cases} (D_L)_{(i,i)} & \text{if } |(D_L)_{(i,i)}| \leqslant 1, \\ 1 & \text{otherwise.} \end{cases} \tag{2.10}$$

Then $D_L = D_{\mathrm{b}} D_{\mathrm{s}}$. Now (2.6) becomes

$$\left[D_{\mathrm{b}}^{-1} Q_L^{\mathrm{T}} + D_{\mathrm{s}} T\right] x = D_{\mathrm{b}}^{-1} Q_L^{\mathrm{T}} b \tag{2.11}$$

and thus can be solved as

$$x = \left[D_{\mathrm{b}}^{-1} Q_L^{\mathrm{T}} + D_{\mathrm{s}} T\right]^{-1} \left[D_{\mathrm{b}}^{-1} (Q_L^{\mathrm{T}} b)\right]. \tag{2.12}$$

Remark 2.1. There is a variation to the above derivation. Consider $p = 1$ in both (2.1) and (2.4), and thus $D_j = \mathrm{diag}(R_j)$. Doing so effectively eliminates the discrepancy of the magnitudes in diagonal entries of each R_j and eventually propagates the discrepancy to D_L. In (2.11), we split D_L into two: D_{b} with larger magnitudes and D_{s} with smaller magnitudes, and then pull out D_{b} while leaving D_{s} in place. D_{b}^{-1} effectively annihilates the top many rows of Q_L^{T} while D_{s} does the same to the bottom many rows of T, and finally the sum $D_{\mathrm{b}}^{-1} Q_L^{\mathrm{T}} + D_{\mathrm{s}} T$ becomes fairly well-conditioned. This suggests that it may not be necessary to pull out the smaller diagonal entries, along with the larger ones, of R_j in the absolute value out in the first place. Namely in (2.1) and (2.4), instead of $D_j = \mathrm{diag}(R_j)$, we may set for $1 \leqslant i \leqslant n$

$$(D_j)_{(i,i)} = \begin{cases} (R_j)_{(i,i)} & \text{if } |(R_j)_{(i,i)}| > 1, \\ 1 & \text{otherwise.} \end{cases} \tag{2.13}$$

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

Then, we still have (2.6) in the same form, but instead of (2.11), we will just have

$$\left[D_L^{-1}Q_L^{\mathrm{T}} + T\right]x = D_L^{-1}\,Q_L^{\mathrm{T}}b. \tag{2.14}$$

We have compared the numerical solutions via solving (2.11) and (2.14), respectively, and found that for our tests, there is little difference in the conditioning of (2.11) and (2.14), but $\prod_i \|T_i\|$ grows much faster for with (2.13) than with (2.1) and (2.4) for $p = 1$ as the conditioning of the original system (1.1) gets worse. That makes solving (2.14) accurately potentially more difficult than solving (2.11).

In summary, the above transformation via QRD with column pivoting naturally lead to the following algorithm to solve the linear system (1.1). As we pointed out above, it is the same as the existing practice up to line 7. After that line, the existing practice goes as in (2.7) which can be less accurate than what comes out of lines 8–11 as we explained immediately following (2.7).

Algorithm. ASvQRD (Accurate Solution via QRD with column pivoting)
Input: $B_1, B_2, \ldots, B_L$ and b.
Output: Solution of (1.1)

1. $Q_1R_1P_1 = B_1$ (QRD with column pivoting);
2. $D_1 = \mathrm{diag}(R_1)$, $T_1 = D_1^{-1}R_1P_1$;
3. for $j = 2, 3, \ldots, L$ do
4. $\quad C_j = (B_jQ_{j-1})D_{j-1}$ (respect the parentheses);
5. $\quad Q_jR_jP_j = C_j$ (QRD with column pivoting);
6. $\quad D_j = \mathrm{diag}(R_j)$, $T_j = D_j^{-1}R_jP_j$;
7. enddo
8. Decompose $D_L = D_{\mathrm{b}}\,D_{\mathrm{s}}$ as in (2.9) and (2.10);
9. $T = T_L \cdots T_2T_1$;
10. $H = D_{\mathrm{b}}^{-1}Q_L^{\mathrm{T}} + D_{\mathrm{s}}T$;
11. Solve $Hx = D_{\mathrm{b}}^{-1}\,(Q_L^{\mathrm{T}}b)$ for x.

Remark 2.2. In lines 2 and 6, D_j is chosen as in (2.1) and (2.4) with $p = 1$. But any other p, in particular 2 or ∞, gives good D_j, too. QRD with column pivoting in lines 1 and 5 can be implemented with Householder transformations. Our later analysis for line 11 assumes a backward stable solution. This can be done, for example, by a QRD (with/without column pivoting). In practice, often Gaussian elimination with partial pivoting suffices, although with no guarantee [5].

2.2. Via SVD by one-sided Jacobi method

The part of transforming (1.1) into an equivalent one with a manageable condition number is very similar to what we have done in Section 2.1, except here we will use SVD computed by the one-sided Jacobi method [6–9]. Let

$$B_1 = U_1\,\Sigma_1 V_1^{\mathrm{T}}, \tag{2.15}$$

be B_1's SVD, where U_1 and V_1 are orthogonal, Σ_1 is diagonal. It is not necessary for *this* SVD to be computed by a one-sided Jacobi method, but rather any stable methods [5], e.g., the QR algorithm or the divide-and-conquer algorithm, will be sufficient.

For j from 2 to L, compute SVD of $B_jU_{j-1}\,\Sigma_{j-1}$ by the one-sided Jacobi method from the left to get

$$(B_jU_{j-1})\,\Sigma_{j-1} = U_j\,\Sigma_jV_j^{\mathrm{T}}. \tag{2.16}$$

Here also the parentheses in $(B_jU_{j-1})\,\Sigma_{j-1}$ must be respected. It follows from $B_1 = U_1\,\Sigma_1V_1^{\mathrm{T}}$ and (2.16) that SVD of $B_L \cdots B_2B_1$ is

$$B_L \cdots B_2B_1 = U_L\,\Sigma_L\,(V_1V_2 \cdots V_L)^{\mathrm{T}},$$

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

and finally the linear system (1.1) is transformed into

$$\left[I_n + U_L \Sigma_L (V_1 V_2 \cdots V_L)^{\mathrm{T}}\right] x = b. \tag{2.17}$$

For ill-conditioned system (1.1), the diagonal entries of Σ_L have wide range of magnitudes, while $(V_1 V_2 \cdots V_L)^{\mathrm{T}}$, being orthogonal, is perfectly well-conditioned. This latter is the major advantage of using SVD over QRD, upon comparing (2.17) with (2.6).

We adopt the same strategy to solve (2.17) as we did for (2.6). Define two $n \times n$ diagonal matrices Σ_{b} and Σ_{s} as follows: for $1 \leqslant i \leqslant n$

$$(\Sigma_{\mathrm{b}})_{(i,i)} = \begin{cases} (\Sigma_L)_{(i,i)} & \text{if } (\Sigma_L)_{(i,i)} > 1, \\ 1 & \text{otherwise,} \end{cases} \tag{2.18}$$

and

$$(\Sigma_{\mathrm{s}})_{(i,i)} = \begin{cases} (\Sigma_L)_{(i,i)} & \text{if } (\Sigma_L)_{(i,i)} \leqslant 1, \\ 1 & \text{otherwise.} \end{cases} \tag{2.19}$$

Then $\Sigma_L = \Sigma_{\mathrm{b}}\, \Sigma_{\mathrm{s}}$. Now (2.17) becomes

$$\left[\Sigma_{\mathrm{b}}^{-1} U_L^{\mathrm{T}} + \Sigma_{\mathrm{s}} (V_1 V_2 \cdots V_L)^{\mathrm{T}}\right] x = \Sigma_{\mathrm{b}}^{-1}\, U_L^{\mathrm{T}} b \tag{2.20}$$

and thus can be solved as

$$x = \left[\Sigma_{\mathrm{b}}^{-1} U_L^{\mathrm{T}} + \Sigma_{\mathrm{s}} (V_1 V_2 \cdots V_L)^{\mathrm{T}}\right]^{-1} \left[\Sigma_{\mathrm{b}}^{-1}\, (U_L^{\mathrm{T}} b)\right]. \tag{2.21}$$

Remark 2.3. A remark similar to Remark 2.1 is applicable here. Namely, instead of (2.16), we do, for j from 2 to L,

$$(B_j U_{j-1}) \Omega_{j-1} = U_j \Sigma_j V_j^{\mathrm{T}}, \tag{2.22}$$

where Ω_j is diagonal and defined by for $1 \leqslant i \leqslant n$

$$(\Omega_j)_{(i,i)} = \begin{cases} (\Sigma_j)_{(i,i)} & \text{if } (\Sigma_j)_{(i,i)} > 1, \\ 1 & \text{otherwise.} \end{cases} \tag{2.23}$$

Finally, instead of (2.17) and (2.20), we will have

$$\left[\Omega_L^{-1} U_L^{\mathrm{T}} + \Lambda_L V_L^{\mathrm{T}} \cdots \Lambda_2 V_2 \Lambda_1 V_1\right] x = \Omega_L^{-1}\, U_L^{\mathrm{T}} b, \tag{2.24}$$

where Λ_j is diagonal and defined by for $1 \leqslant i \leqslant n$

$$(\Lambda_j)_{(i,i)} = \begin{cases} (\Sigma_j)_{(i,i)} & \text{if } (\Sigma_j)_{(i,i)} \leqslant 1, \\ 1 & \text{otherwise.} \end{cases}$$

We have also compared the numerical solutions via solving (2.20) and (2.24), respectively, and found that the solutions were about equally good.

In summary, we have the following SVD-based method to solve the linear system (1.1).

Algorithm. ASvSVD (Accurate Solution via SVD)
Input: $B_1, B_2, \ldots, B_L$ and b.
Output: Solution of (1.1)

1. $U_1 \Sigma_1 V_1^{\mathrm{T}} = B_1$ (SVD by any stable method);
2. for $j = 2, 3, \ldots, L$ do
3. $\quad C_j = (B_j U_{j-1}) \Sigma_{j-1}$ (respect the parentheses);
4. $\quad U_j \Sigma_j V_j^{\mathrm{T}} = C_j$ (SVD by one-sided Jacobi method);
5. enddo

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

6. Decompose $\Sigma_L = \Sigma_b \Sigma_s$ as in (2.18) and (2.19);
7. $V = V_1 V_2 \cdots V_L$;
8. $H = \Sigma_b^{-1} U_L^T + \Sigma_s V^T$;
9. Solve $Hx = \Sigma_b^{-1} (U_L^T b)$ for x.

Remark 2.4. SVD in line 1 can be computed by any backward stable method as we already pointed out. Typically SVD by one-sided Jacobi method for a dense matrix starts by computing its QR decomposition with (column) pivoting (or any rank-revealing QR decomposition) and then performs Jacobi iterations on the R-factor [8,9]. Any other general SVD method (possibly used in line 1 in ASvSVD) starts by bidiagonalizing the matrix and then computes SVD of the resulting bidiagonal matrix by, e.g., the QR, divide-and-conquer, or bisection algorithms [5]. Thus ASvSVD is expected to be much slower than ASvQRD because the first phases in each of these mentioned SVD algorithms cost about as much as a QRD (with column pivoting). But the gain is a much more accurate method for (1.1) as our later examples will show.

Our later analysis for line 9 assumes a backward stable solution, similarly to what we remarked for Algorithm ASvQRD.

3. Error analysis

In this section, we will show that the transformations in Section 2, if done in the IEEE floating arithmetic, will lead to transformed systems that have the same solutions as certain nearby systems of (1.1), and that the computed solutions of the transformed systems have small forward errors. This, however, is not the same as the usual notion of being backward stable, but a weaker statement. In view of this, we call any computed solution that is close to the exact solution of a nearby problem is a *weakly backward stable solution*, and any algorithm that computes such a solution is a *weakly backward stable algorithm*.

Assume the entries in B_j and b are already stored in the IEEE floating point format, and let ϵ_m be the machine unit roundoff. In exact arithmetic, the linear system (2.6) is equivalent to the original system (1.1), i.e., both have the same solution. Computationally, we do not have (2.6) exactly. Instead, we have the following computed one by ASvQRD:

$$[I_n + \widehat{Q}_L \widehat{D}_L (\widehat{T}_L \cdots \widehat{T}_2 \widehat{T}_1)]\,\widehat{x} = b. \tag{3.1}$$

Likewise, we do not have (2.17) exactly but have the following computed one by ASvSVD:

$$\left[I_n + \widehat{U}_L \widehat{\Sigma}_L (\widehat{V}_1 \widehat{V}_2 \cdots \widehat{V}_L)^T\right]\widehat{x} = b. \tag{3.2}$$

In the rest of the analysis in this section, we shall adopt the following notation convention: denote their computed counterparts for those objects in (2.2)–(2.6) and in (2.15)–(2.17) by the same symbols with a hat, i.e., the computed Q_j is $\widehat{Q}_j$, with an exception that $\widehat{x}$ *is* the exact solution of (3.1) in the case of ASvQRD, and the exact solution of (3.2) in the case of ASvSVD. We will also use $\mathrm{fl}(\cdot)$ to denote the computed result of an expression whenever convenient.

Our analysis is intended to demonstrate only the order of error magnitudes, instead of precise error bounds. Doing so significantly simplifies the analysis, making it much easier to understand and yet suggestive as to how big the errors may be. In particular, $X = \mathcal{O}(\alpha)$ means $\|X\|_p \leqslant f(n)\alpha$ for some low degree polynomial of n, where X is either a vector or matrix. In view of this simplification, the choice of which norm becomes insignificant, and thus $\|\cdot\|_2$ will be used throughout.

We begin by analyzing ASvQRD. The theorem below says that the computed counterpart (3.1) of (2.6) is equivalent to a structurally nearby system of (1.1).

Theorem 3.1. *The computed system* (3.1) *by lines 1–7 of* ASvQRD *is structurally backward stable. Specifically* (3.1) *is equivalent to*

$$[I_n + (B_L + \Delta B_L) \cdots (B_2 + \Delta B_2)(B_1 + \Delta B_1)]\widehat{x} = b, \tag{3.3}$$

where $\Delta B_j = \mathcal{O}(\epsilon_m \|B_j\|_2)$ *for* $1 \leqslant j \leqslant L$.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

Proof. It is known that for the QR decomposition (2.2) [11, pp. 360–361]

$$\widehat{Q}_1\widehat{R}_1P_1 = B_1 + E_1,$$

where $\widehat{Q}_1 = Q_1 + \Delta Q_1$, $\Delta Q_1 = \mathcal{O}(\epsilon_{\mathsf{m}})$ and $(E_1)_{(:,i)} = \mathcal{O}(\epsilon_{\mathsf{m}}\|(B_1)_{(:,i)}\|_2)$. Since

$$\widehat{T}_1 = \mathrm{fl}\left((\widehat{D}_1)^{-1}\widehat{R}_1P_1\right) = \left[(\widehat{D}_1)^{-1}\widehat{R}_1 + F_1\right]P_1, \quad |F_1| \leqslant \epsilon_{\mathsf{m}}\,|\widehat{D}_1|^{-1}|\widehat{R}_1|, \tag{3.4}$$

we have

$$\begin{aligned}\widehat{Q}_1\widehat{D}_1\widehat{T}_1 &= \widehat{Q}_1\widehat{R}_1P_1 + \widehat{Q}_1\widehat{D}_1F_1P_1,\\ &= B_1 + E_1 + \widehat{Q}_1\widehat{D}_1F_1P_1,\\ &\equiv B_1 + \Delta B_1,\end{aligned} \tag{3.5}$$

where

$$\Delta B_1 \overset{\mathrm{def}}{=} E_1 + \widehat{Q}_1\widehat{D}_1F_1P_1 = \mathcal{O}\left(\epsilon_{\mathsf{m}}\|B_1\|_2\right). \tag{3.6}$$

Now for the decomposition (2.3), we have similarly,[2]

$$\begin{aligned}\widehat{Q}_j\widehat{R}_jP_j &= \mathrm{fl}\left(\mathrm{fl}\left(B_j\widehat{Q}_{j-1}\right)\widehat{D}_{j-1}\right) + E_j,\\ &= \left[\left(B_j\widehat{Q}_{j-1} + F_{j,1}\right)\widehat{D}_{j-1} + F_{j,2}\right] + E_j,\end{aligned} \tag{3.7}$$

$$\widehat{Q}_j = Q_j + \Delta Q_j, \tag{3.8}$$

where

$$\begin{aligned}&F_{j,1} = \mathcal{O}\left(\epsilon_{\mathsf{m}}\|B_j\|_2\right),\\ &|F_{j,2}| \leqslant \epsilon_{\mathsf{m}}\,|B_j\widehat{Q}_{j-1} + F_{j,1}|\,|\widehat{D}_{j-1}|,\\ &\Delta Q_j = \mathcal{O}\left(\epsilon_{\mathsf{m}}\right),\\ &(E_j)_{(:,i)} = \mathcal{O}\left(\epsilon_{\mathsf{m}}\left\|\left[(B_j\widehat{Q}_{j-1} + F_{j,1})\widehat{D}_{j-1} + F_{j,2}\right]_{(:,i)}\right\|_2\right).\end{aligned}$$

Since

$$\widehat{T}_j = \mathrm{fl}\left((\widehat{D}_j)^{-1}\widehat{R}_jP_j\right) = \left[(\widehat{D}_j)^{-1}\widehat{R}_j + F_{j,3}\right]P_j, \quad |F_{j,3}| \leqslant \epsilon_{\mathsf{m}}\,|\widehat{D}_j|^{-1}|\widehat{R}_j|, \tag{3.9}$$

we have

$$\begin{aligned}\widehat{Q}_j\widehat{D}_j\widehat{T}_j &= \widehat{Q}_j\widehat{R}_jP_j + \widehat{Q}_j\widehat{D}_jF_{j,3}P_j,\\ &= \left[(B_j\widehat{Q}_{j-1} + F_{j,1})\widehat{D}_{j-1} + F_{j,2}\right] + E_j + \widehat{Q}_j\widehat{D}_jF_{j,3}P_j,\\ &\equiv (B_j + \Delta B_j)\widehat{Q}_{j-1}\widehat{D}_{j-1},\end{aligned} \tag{3.10}$$

where

$$\Delta B_j\widehat{Q}_{j-1} \equiv F_{j,1} + F_{j,2}\widehat{D}_{j-1}^{-1} + E_j\,\widehat{D}_{j-1}^{-1} + \widehat{Q}_j\widehat{D}_jF_{j,3}P_j\widehat{D}_{j-1}^{-1}. \tag{3.11}$$

We claim that each of the four summands in the right-hand side of this equation is of $\mathcal{O}(\epsilon_{\mathsf{m}}\|B_j\|_2)$. Consequently we have

$$\Delta B_j = \mathcal{O}\left(\epsilon_{\mathsf{m}}\|B_j\|_2\right) \tag{3.12}$$

[2] Technically speaking, Q_j in (3.8) is not the same as the one in (2.3). But rather it is the Q-factor in QRD with column pivoting for $[(B_j\widehat{Q}_{j-1} + F_{j,1})\widehat{D}_{j-1} + F_{j,2}]$ in the exact arithmetic. This abuse of the notation Q_j will unlikely cause any problem in this analysis. What we really need from (3.8) is the mere fact that $\widehat{Q}_j$ is away from an orthogonal matrix by a perturbation $\Delta Q_j = \mathcal{O}(\epsilon_{\mathsf{m}})$.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

since $\widehat{Q}_{j-1}$ is orthogonal to the working precision.[3] We now look into the summands in the right-hand side of (3.11). $F_{j,1} = \mathcal{O}(\epsilon_{\mathrm{m}}\|B_j\|_2)$ by (3.7) whose second equation says $|F_{j,2}\widehat{D}_{j-1}^{-1}| \leqslant \epsilon_{\mathrm{m}}\,|B_j\widehat{Q}_{j-1} + F_{j,1}|$ and therefore $F_{j,2}\widehat{D}_{j-1}^{-1} = \mathcal{O}(\epsilon_{\mathrm{m}}\|B_j\|_2)$ also. For the third summand, we have for $1 \leqslant i \leqslant n$,

$$\left(E_j\,\widehat{D}_{j-1}^{-1}\right)_{(:,i)} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\left\|\left(B_j\widehat{Q}_{j-1} + F_{j,1} + F_{j,2}\widehat{D}_{j-1}^{-1}\right]_{(:,i)}\right\|_2\right) = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|B_j\|_2\right)$$

which leads to $E_j\,\widehat{D}_{j-1}^{-1} = \mathcal{O}(\epsilon_{\mathrm{m}}\|B_j\|_2)$. Finally for the fourth summand, we notice by (3.9) and (3.7) that

$$\left|\widehat{D}_jF_{j,3}P_j\widehat{D}_{j-1}^{-1}\right| \leqslant \epsilon_{\mathrm{m}}\,\left|\widehat{R}_jP_j\widehat{D}_{j-1}^{-1}\right|,$$
$$\widehat{Q}_j\widehat{R}_jP_j\widehat{D}_{j-1}^{-1} = B_j\widehat{Q}_{j-1} + F_{j,1} + F_{j,2}\widehat{D}_{j-1}^{-1} + E_j\,\widehat{D}_{j-1}^{-1}.$$

Therefore

$$\begin{aligned}
\widehat{Q}_j\widehat{D}_jF_{j,3}P_j\widehat{D}_{j-1}^{-1} &= \mathcal{O}\left(\left\|\widehat{D}_jF_{j,3}P_j\widehat{D}_{j-1}^{-1}\right\|_2\right),\\
&= \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\left\|\widehat{R}_jP_j\widehat{D}_{j-1}^{-1}\right\|_2\right),\\
&= \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\left\|\widehat{Q}_j\widehat{R}_jP_j\widehat{D}_{j-1}^{-1}\right\|_2\right),\\
&= \mathcal{O}\left(\epsilon_{\mathrm{m}}\left\|B_j\widehat{Q}_{j-1} + F_{j,1} + F_{j,2}\widehat{D}_{j-1}^{-1} + E_j\,\widehat{D}_{j-1}^{-1}\right\|_2\right),\\
&= \mathcal{O}\left(\epsilon_{\mathrm{m}}\|B_j\|_2\right),
\end{aligned}$$

as was to be shown. Thus (3.12) holds. It follows from (3.5) and (3.10) that

$$\begin{aligned}
(B_L + \Delta B_L)\cdots(B_1 + \Delta B_1) &= (B_L + \Delta B_L)\cdots(B_2 + \Delta B_2)\widehat{Q}_1\widehat{D}_1\,\widehat{T}_1,\\
&= (B_L + \Delta B_L)\cdots(B_3 + \Delta B_3)\widehat{Q}_2\widehat{D}_2\,\widehat{T}_2\widehat{T}_1,\\
&= \widehat{Q}_L\widehat{D}_L\,(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1).
\end{aligned}$$

This completes the proof. □

Our next theorem shows that the numerical solution to (3.1) by lines 8–11 of ASvQRD algorithm suffers from an error, relative to the exact solution $\widehat{x}$ of (3.1), approximately $\mathcal{O}(\epsilon_{\mathrm{m}}\kappa(H))$ modulo a factor typically of $\mathcal{O}(1)$ in practice. This is done with the modest assumption that line 11 of ASvQRD is backward stable.

Theorem 3.2. *The computed solution $\widetilde{x}$ of (3.1) by lines 8–11 of* ASvQRD *satisfies*

$$\frac{\|\widetilde{x} - \widehat{x}\|_2}{\|\widehat{x}\|_2} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\kappa_2(\widehat{H}_{qr})\left[\frac{1 + \|\widehat{T}_L\|_2\cdots\|\widehat{T}_2\|_2\|\widehat{T}_1\|_2}{\|\widehat{H}_{qr}\|_2} + \frac{\|b\|_2}{\left\|\widehat{D}_{\mathrm{b}}^{-1}[\widehat{Q}_L^{-1}b]\right\|_2}\right]\right), \tag{3.13}$$

assuming line 11 of ASvQRD *is backward stable, where*

$$\widehat{H}_{qr} = \widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{\mathrm{T}} + \widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1). \tag{3.14}$$

Proof. The exact solution $\widehat{x}$ satisfies, upon substituting $\widehat{D}_L = \widehat{D}_{\mathrm{b}}\,\widehat{D}_{\mathrm{s}}$,

$$\left[\widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{-1} + \widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1)\right]\widehat{x} = \widehat{D}_{\mathrm{b}}^{-1}[\widehat{Q}_L^{-1}b]. \tag{3.15}$$

The computed solution $\widetilde{x}$ is obtained through solving

[3] By that X is orthogonal to the working precision, we mean that $X + \Delta X$ is orthogonal for some $\Delta X = \mathcal{O}(\epsilon_{\mathrm{m}})$.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

$$\widetilde{H}y = \mathrm{fl}\left(\widehat{D}_{\mathrm{b}}^{-1}\left(\widehat{Q}_L^{\mathrm{T}}b\right)\right), \quad \text{where} \quad \widetilde{H} = \mathrm{fl}\left(\widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{\mathrm{T}} + \widehat{D}_{\mathrm{s}}\left(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1\right)\right). \tag{3.16}$$

It can be seen that[4]

$$\begin{aligned}
\mathrm{fl}(\widehat{D}_{\mathrm{b}}^{-1}(\widehat{Q}_L^{\mathrm{T}}b)) &= \widehat{D}_b^{-1}(\widehat{Q}_L^{-1}b + f_1) + f_2\\
&\equiv \widehat{D}_{\mathrm{b}}^{-1}[\widehat{Q}_L^{-1}b] + f,\\
\widetilde{H} &= \left(\widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{-1} + F_1\right) + [\widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1) + F_2] + F_3\\
&\equiv \left[\widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{-1} + \widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1)\right] + F,
\end{aligned}$$

where[5]

$$\begin{aligned}
&f_1 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|b\|_2\right),\\
&|f_2| \leqslant \epsilon_{\mathrm{m}}\left|D_{\mathrm{b}}^{-1}\right|\left|\widehat{Q}_L^{-1}b + f_1\right| \leqslant \epsilon_{\mathrm{m}}\left|\widehat{Q}_L^{-1}b + f_1\right|,\\
&f = \widehat{D}_{\mathrm{b}}^{-1}f_1 + f_2 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|b\|_2\right),\\
&F_1 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\right),\\
&F_2 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|\widehat{T}_L\|_2\cdots\|\widehat{T}_2\|_2\|\widehat{T}_1\|_1\right),\\
&F_3 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\max\left\{1, \|\widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1)\|_2\right\}\right),\\
&F = F_1 + F_2 + F_3
\end{aligned} \tag{3.17}$$

$$= \mathcal{O}\left(\epsilon_{\mathrm{m}}(1 + \|\widehat{T}_L\|_2\cdots\|\widehat{T}_2\|_2\|\widehat{T}_1\|_2)\right). \tag{3.18}$$

Therefore the exact solution $\widehat{x}$ of (3.15) and the exact solution y of (3.16) satisfy [5, p. 32]

$$\frac{\|y-\widehat{x}\|_2}{\|y\|_2} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\kappa_2(\widehat{H}_{qr})\left[\frac{1 + \|\widehat{T}_L\|_2\cdots\|\widehat{T}_2\|_2\|\widehat{T}_1\|_2}{\|\widehat{H}_{qr}\|_2} + \frac{\|b\|_2}{\|\widehat{D}_{\mathrm{b}}^{-1}[\widehat{Q}_L^{-1}b]\|_2}\right]\right). \tag{3.19}$$

Since it is assumed that the computed solution $\widetilde{x}$ of (3.16) is backward stable, we have

$$\frac{\|\widetilde{x}-y\|_2}{\|y\|_2} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\kappa(\widehat{H}_{qr})\right). \tag{3.20}$$

Finally (3.13) is a consequence of (3.19) and (3.20), upon noticing that $\|y\|_2 \approx \|\widehat{x}\|_2$ by (3.19). □

Remark 3.1. In Section 2.1, we mentioned two purposes of picking of D_i, i.e., to make T_j well-conditioned and to make $\|T_j\|_2$ near 1, that dictate the choices of D_1 and D_j as in (2.1) and (2.4). We now see why. Making $\|T_j\|_2$ nearly 1 is to make sure that the ratio

$$\frac{1 + \|\widehat{T}_L\|_2\cdots\|\widehat{T}_2\|_2\|\widehat{T}_1\|_2}{\|\widehat{H}_{qr}\|_2}$$

in the right-hand side of (3.13) does not grow out of control. It also keeps the two summands in $\widehat{H}_{scqr}$ to have similar magnitudes and potentially removes any ill-conditionedness in $\widehat{H}_{qr}$, otherwise due to potentially large differences between their magnitudes. To explain why T_j should be made well-conditioned, we notice that the top many rows of $\widehat{D}_{\mathrm{b}}^{-1}\widehat{Q}_L^{\mathrm{T}}$ and the bottom many rows of $\widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1)$ are very much negligible because of the behavior of the magnitudes of the entries in $\widehat{D}_{\mathrm{b}}^{-1}$ and $\widehat{D}_{\mathrm{s}}$. Thus

[4] Both f_1 and F_1 contain the rounding errors from floating point arithmetic operations and the $\mathcal{O}(\epsilon_{\mathrm{m}})$ error from replacing $\widehat{Q}_L^{\mathrm{T}}$ by $\widehat{Q}_L^{-1}$.

[5] In contributing to f, part of f_1 is considerably offset by the first many diagonal entries of $\widehat{D}_{\mathrm{b}}^{-1}$ with extremely tiny magnitudes. This is often the case even for modest well-conditioned B_j and modest L such as $\kappa_2(B_j) \geqslant 100$ and $L \geqslant 8$. But since f_1 is unknown, it is very difficult to incorporate such an observation into the error estimate. In general, the estimate for F_2 in (3.17) is attainable, but often in practice it may be more like $\mathcal{O}(\epsilon_{\mathrm{m}}\|\widehat{D}_{\mathrm{s}}(\widehat{T}_L\cdots\widehat{T}_2\widehat{T}_1)\|_2)$.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

roughly speaking the top many rows of $\widehat{H}_{qr}$ are pretty much those of $\widehat{T}_L \cdots \widehat{T}_2\widehat{T}_1$, and thus for $\widehat{H}_{qr}$ to be well-conditioned, it is necessary that the top many rows $\widehat{T}_L \cdots \widehat{T}_2\widehat{T}_1$ must be well-conditioned. To make sure of that, one thing we can do is to make sure all $\widehat{T}_j$ well-conditioned.

We now analyze Algorithm ASvSVD. The technicality is very much similar.

Theorem 3.3. *The computed* (3.2) *by lines 1–5 of* ASvSVD *is structurally backward stable. Specifically* (3.2) *is equivalent to some*

$$[I_n + (B_L + \Delta B_L)\cdots(B_2 + \Delta B_2)(B_1 + \Delta B_1)]\widehat{x} = b, \tag{3.21}$$

where $\Delta B_j = \mathcal{O}(\epsilon_{\mathrm{m}}\|B_j\|_2)$ *for* $1 \leqslant j \leqslant L$.

Proof. It is well-known that for the decomposition (2.15) [5]

$$\widehat{U}_1\widehat{\Sigma}_1\widehat{V}_1^{\mathrm{T}} = B_1 + \Delta B_1, \tag{3.22}$$

where $\Delta B_1 = \mathcal{O}(\epsilon_{\mathrm{m}}\|B_1\|_2)$, $\widehat{U}_1$ and $\widehat{V}_1$ are orthogonal to the working precision. For the SVD (2.16) by the one-sided Jacobi method on $\mathrm{fl}((B_j\widehat{U}_{j-1})\widehat{\Sigma}_{j-1})$, we have [7–9]

$$\mathrm{fl}((B_j\widehat{U}_{j-1})\widehat{\Sigma}_{j-1}) = (B_j\widehat{U}_{j-1} + F_{j,1})\widehat{\Sigma}_{j-1} + F_{j,2}, \tag{3.23}$$

$$\begin{aligned}\widehat{U}_j\widehat{\Sigma}_j\widehat{V}_j^{\mathrm{T}} &= \mathrm{fl}((B_j\widehat{U}_{j-1})\widehat{\Sigma}_{j-1}) + F_{j,3},\\ &= \left[(B_j\widehat{U}_{j-1} + F_{j,1})\widehat{\Sigma}_{j-1} + F_{j,2}\right] + F_{j,3},\\ &\equiv (B_j + \Delta B_j)\widehat{U}_{j-1}\widehat{\Sigma}_{j-1},\end{aligned} \tag{3.24}$$

where $\widehat{U}_j$ and $\widehat{V}_j$ are orthogonal to the working precision, and

$$\begin{aligned}&F_{j,1} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|B_j\|_2\right),\\ &|F_{j,2}| \leqslant \epsilon_{\mathrm{m}}\,|B_j\widehat{U}_{j-1} + F_{j,1}|\,\widehat{\Sigma}_{j-1},\\ &\|(F_{j,3})_{(:,i)}\|_2 = \mathcal{O}\left(\epsilon_{\mathrm{m}}\left\|\left[(B_j\widehat{U}_{j-1} + F_{j,1})\widehat{\Sigma}_{j-1} + F_{j,2}\right]_{(:,i)}\right\|_2\right),\\ &\Delta B_j \equiv F_{j,1}\widehat{U}_{j-1}^{-1} + (F_{j,2} + F_{j,3})\widehat{\Sigma}_{j-1}^{-1}\widehat{U}_{j-1}^{-1}\\ &\qquad = \mathcal{O}\left(\epsilon_{\mathrm{m}}\|B_j\|_2\right).\end{aligned} \tag{3.25}$$

It follows from (3.22) and (3.25) that

$$\begin{aligned}(B_L + \Delta B_L)\cdots(B_1 + \Delta B_1) &= (B_L + \Delta B_L)\cdots(B_2 + \Delta B_2)\widehat{U}_1\widehat{\Sigma}_1\widehat{V}_1^{\mathrm{T}},\\ &= (B_L + \Delta B_L)\cdots(B_3 + \Delta B_3)\widehat{U}_2\widehat{\Sigma}_2\widehat{V}_2^{\mathrm{T}}\widehat{V}_1^{\mathrm{T}},\\ &= \widehat{U}_L\widehat{\Sigma}_L\,(\widehat{V}_1\widehat{V}_2\cdots\widehat{V}_L)^{\mathrm{T}}.\end{aligned}$$

This completes the proof. □

Theorem 3.4. *The computed solution* $\widetilde{x}$ *of* (3.2) *by lines 6–9 of* ASvSVD *satisfies*

$$\frac{\|\widetilde{x} - \widehat{x}\|_2}{\|\widehat{x}\|_2} = \mathcal{O}\left(\epsilon_{\mathrm{m}}\,\kappa_2(\widehat{H}_{svd})\left[\frac{1}{\|\widehat{H}_{svd}\|_2} + \frac{\|b\|_2}{\|\widehat{\Sigma}_{\mathrm{b}}^{-1}[\widehat{U}_L^{-1}b]\|_2}\right]\right), \tag{3.26}$$

assuming that line 9 of ASvSVD *is backward stable, where*

$$\widehat{H}_{svd} = \widehat{\Sigma}_{\mathrm{b}}^{-1}\widehat{U}^{\mathrm{T}} + \widehat{\Sigma}_{\mathrm{s}}(\widehat{V}_1\widehat{V}_2\cdots\widehat{V}_L)^{\mathrm{T}}. \tag{3.27}$$

Proof. It is similar to the proof of Theorem 3.2. □

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

Theorems 3.1 and 3.3 guarantee that the transformed linear systems via QRD and SVD at the intermediate step of ASvQRD and ASvSVD in the floating point environment are equivalent to some nearby linear systems of the original one, and neither one of the nearby systems is more accurate than the other as far as the sizes of the backward errors are concerned. However, when taking Theorems 3.2 and 3.4 into consideration, the computed solutions by ASvSVD are expected to be closer to their nearby systems than the ones by ASvQRD. This is especially so when $\|\widehat{T}_L\|_2 \cdots \|\widehat{T}_2\|_2\|\widehat{T}_1\|_2$ is much larger than 1. But the extra accuracy is achieved at additional cost since the SVD by the one-sided Jacobi method is more expensive than the QRD (see Remark 2.4).

Theorems 3.1 to 3.4 together prove that both ASvQRD and ASvSVD are weakly backward stable.

4. Numerical examples

In this section, we present numerical results for the two methods presented in Section 2 and analyzed in Section 3. All our test problems (1.1) are drawn from the quantum Monte Carlo simulation of the Hubbard model in condensed-matter physics [1–3,13,15]. Specifically, for $i = 1, 2, \ldots, L$, the $n \times n$ matrix $B_i = e^{(\Delta\tau)K}e^{\Gamma_i}$, K is the so-called *hopping matrix*. It is an adjacency matrix of the $m \times m$ square lattice, i.e., $K = K_1 \otimes I_m + I_m \otimes K_1$,

$$K_1 = \begin{bmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 0 \end{bmatrix}.$$

$n = m^2$. $\Delta\tau$ is the time discretization parameter. The product $\beta = (\Delta\tau)L$ is the inverse templates. Γ_i is a diagonal matrix of random diagonal elements λ or $-\lambda$ of equal probability, where $\lambda = \cosh^{-1}(e^{\mathcal{U}(\Delta\tau)/2})$, and $\mathcal{U}$ is a potential energy parameter for local repulsion between electrons. Two crucial parameters are β and $\mathcal{U}$, which dictate the conditioning of B_j; the larger β and/or $\mathcal{U}$ are, the worse the conditioning of B_j is and consequently the worse-conditioning of (1.1) becomes. Note that there are certain randomness in generating B_j, too. The right-hand side b is simply taken to be a random vector with entries chosen from a normal distributions with mean zero and variance one.

Both methods are tested for different parameter values of the linear systems (1.1). Let us examine a typical set of numerical results in detail. Consider $n = 16 \times 16 = 256$ and $L = 16$ and various β and $\mathcal{U}$. Fig. 4.1 plots the absolute values of the diagonal entries of D_L as the results of ASvQRD and the diagonal entries of Σ_L as the results of ASvSVD. Tables 4.1 and 4.2 display quantities needed by Theorems 3.2 and 3.4. The relative error bound (3.13) in Theorem 3.2 is given by

$$\epsilon_{qr} = \epsilon_{\mathrm{m}}\kappa_2(\widehat{H}_{qr})(\alpha_1 + \alpha_2),$$

where

$$\alpha_1 = \frac{1 + \|\widehat{T}_L\|_2 \cdots \|\widehat{T}_2\|_2\|\widehat{T}_1\|_2}{\|\widehat{H}_{qr}\|_2}, \quad \alpha_2 = \frac{\|b\|_2}{\|\widehat{D}_{\mathrm{b}}^{-1}[\widehat{Q}_L^{\mathrm{T}}b]\|_2}.$$

Similarly, the relative error bound (3.26) in Theorem 3.4 is given by

$$\epsilon_{svd} = \epsilon_{\mathrm{m}}\kappa_2(\widehat{H}_{svd})(\alpha_3 + \alpha_4),$$

where

$$\alpha_3 = \frac{1}{\|\widehat{H}_{svd}\|_2}, \quad \alpha_4 = \frac{\|b\|_2}{\|\widehat{\Sigma}_{\mathrm{b}}^{-1}[\widehat{U}_L^{\mathrm{T}}b]\|_2}.$$

These sets of tests lead us to draw the following observations:

1. The diagonal entries of D_L and Σ_L modestly vary in magnitudes for small β and $\mathcal{U}$, but wildly as β and U get larger and larger (see Fig. 4.1). Since roughly $\kappa_2(I_n + B_L \cdots B_2B_1)$ is comparable to

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

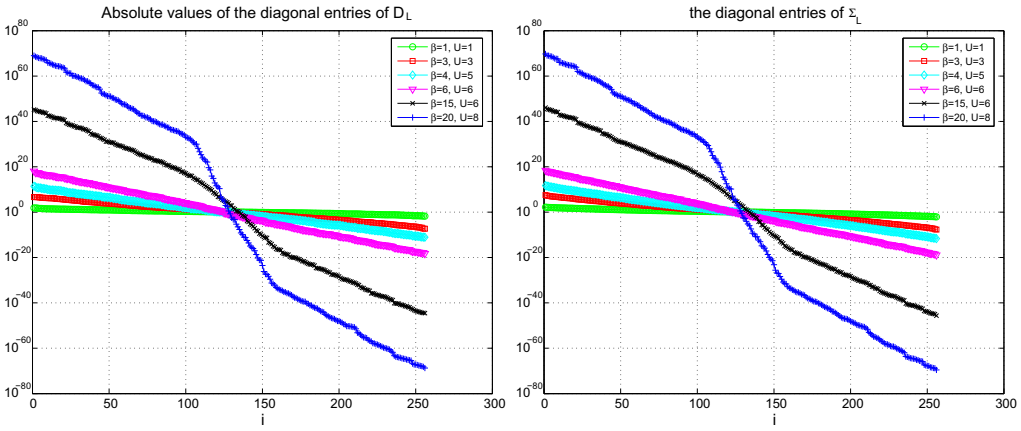

Fig. 4.1. *Left:* The absolute values of the diagonal entries of D_L by ASvQRD. *Right:* The diagonal entries of Σ_L by ASvSVD.

Table 4.1
Results by ASvQRD for various β and $\mathcal{U}$, $d = \|\widehat{D}_b^{-1}[\widehat{Q}^T b]\|_2$.

$(\beta,\mathcal{U})$	$\kappa_2(\widehat{H}_{qr})$	$\|\widehat{H}_{qr}\|_2$	$\prod_i \|\widehat{T}_i\|_2$	d	α_1	α_2	ϵ_{qr}
(1,1)	1.6e+1	6.5e+0	3.0e+1	1.1e+1	4.5e+0	1.4	2.1e−14
(3,3)	1.5e+2	1.2e+1	9.2e+2	9.8e+0	7.9e+1	1.6	2.8e−12
(4,3)	3.5e+2	1.4e+1	1.1e+3	1.1e+1	7.9e+1	1.5	6.4e−12
(3,4)	2.6e+2	1.3e+1	1.3e+3	1.0e+1	1.0e+2	1.5	6.1e−12
(4,5)	2.0e+3	1.7e+1	6.1e+3	1.1e+1	3.6e+2	1.4	1.6e−10
(5,6)	2.2e+3	1.2e+1	1.0e+4	1.1e+1	8.7e+2	1.5	4.2e−10
(6,6)	1.7e+4	1.8e+1	1.8e+4	1.1e+1	9.8e+2	1.5	3.8e−09
(10,6)	7.9e+5	1.8e+1	4.7e+4	9.8e+0	2.6e+3	1.6	4.5e−07
(15,6)	1.0e+5	2.0e+1	4.1e+4	9.6e+0	2.0e+3	1.6	4.5e−08
(20,8)	7.4e+5	1.7e+1	7.6e+4	1.0e+1	4.5e+3	1.5	7.4e−07

$|(D_L)_{(1,1)}|$ and $(\Sigma_L)_{(1,1)}$, it grows rapidly with β and $\mathcal{U}$. For the listed parameter pairs $(\beta,\mathcal{U})$, it is unlikely for the conventional approach of forming $I_n + B_L \cdots B_2 B_1$ explicitly before solving (1.1) to produce meaningful numerical solutions for $\beta,\ \mathcal{U} \geqslant 6$. On the other hand, $\kappa_2(\widehat{H}_{qr})$ and $\kappa_2(\widehat{H}_{svd})$ grow fairly slowly with respect to β and $\mathcal{U}$, relative to the growth of $\kappa_2(I_n + B_L \cdots B_2 B_1)$.

2. According to Table 4.1 and Theorem 3.2, the numerical solutions by ASvQRD have have roughly 8 up to 14 significant decimal digits correct, comparing to the solutions of nearby systems of (1.1).
3. According to Table 4.2 and Theorem 3.4, the numerical solutions by ASvSVD have roughly 12 up to 15 significant decimal digits correct, comparing to the solutions of nearby systems of (1.1).
4. Using the SVD-based algorithm ASvSVD produces more accurate solutions than by the QRD-based algorithm ASvQRD. The difference stems primarily from α_1 which has $\prod_i \|\widehat{T}_i\|_2$ in its numerator and α_3 whose numerator is always 1. $\prod_i \|\widehat{T}_i\|_2$ grows initially with β and $\mathcal{U}$, but quickly settle down to a level, in this case about 10^3.
5. Given all B_i and thus D_L and Q_L by ASvQRD and Σ_L and U_L by ASvSVD, it is not hard to see that artificial b can be constructed to make α_2 and α_4 huge. In fact, α_2 can be made arbitrarily large by making $\widehat{Q}_L^T b$ have nontrivial entries only at its top many entries, while α_4 can be made arbitrarily large by making $\widehat{U}_L^T b$ have nontrivial entries only at its top many entries. When these happen, the bounds by Theorems 3.2 and 3.4 will be very big, suggesting that the computed solution by either algorithms unlikely be close to a nearby system of (1.1). But such highly correlated b is hardly realistic from a practical point of view. In our tests, b is a random vector and both α_2 and α_4 are very modest.
6. $\|\widehat{H}_{qr}\|_2$ and $\|\widehat{H}_{svd}\|_2$ does not vary much with β and $\mathcal{U}$.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

Table 4.2
Results by ASvSVD for various β and $\mathcal{U}$.

$(\beta, \mathcal{U})$	$\kappa_2(\widehat{H}_{svd})$	$\|\widehat{H}_{svd}\|_2$	$\|\widehat{\Sigma}_b^{-1}[\widehat{U}^T b]\|_2$	α_3	α_4	ϵ_{svd}
(1,1)	2.6e+0	2.0	1.1e+1	5.1e−1	1.4	1.1e−15
(3,3)	8.4e+0	1.8	1.1e+1	5.5e−1	1.4	3.6e−15
(4,3)	1.1e+1	1.8	9.7e+0	5.5e−1	1.5	5.0e−15
(3,4)	1.6e+1	1.8	9.8e+0	5.7e−1	1.5	7.5e−15
(4,5)	7.4e+1	1.7	1.0e+1	6.0e−1	1.4	3.3e−14
(5,6)	1.3e+2	1.7	1.1e+1	6.0e−1	1.4	5.9e−14
(6,6)	6.5e+2	1.6	1.1e+1	6.4e−1	1.4	2.9e−13
(10,6)	1.8e+4	1.6	1.0e+1	6.4e−1	1.4	8.1e−12
(15,6)	1.9e+3	1.6	1.0e+1	6.3e−1	1.4	8.9e−13
(20,8)	1.7e+4	1.4	9.7e+0	7.1e−1	1.5	8.6e−12

5. Conclusions

In this paper, we studied two numerical methods for solving linear system (1.1). Algorithm ASvQRD is based on the QR decomposition with column pivoting and Algorithm ASvSVD is based on the singular value decomposition computed by the one-sided Jacobi method. ASvQRD is an improved version of an algorithm already used in the quantum Monte Carlo simulation [12]. Both methods share similarities. Our error analysis suggest that both methods are weakly backward stable, meaning that the computed solutions are close to the exact solutions of (structurally) nearby linear systems.

Numerical results are presented to illustrate the error analysis. As suggested by our analysis, ASvSVD is more accurate than ASvQRD and the gained accuracy becomes more prominent as the conditioning of (1.1) gets worse. But the former is more expensive. A natural recommendation would be to use ASvQRD when its accuracy is sufficient for the application of interest, and switch to ASvSVD otherwise. For the test problems in Section 4, likely ASvQRD is good enough for the choice of the parameters ($\beta, \mathcal{U}, n$, and L). However, it will not be if any of the parameters is much bigger. If time permits, ASvSVD should be always favored, however.

Acknowledgments

We wish to thank Professor Zlatko Drmač for sending them his MATLAB code for the one-sided Jacobi SVD method that was used in Section 4. We are grateful to the referees for valuable suggestions.

Bai was supported in part by the Department of Energy Grant DE-FC02-06ER25793 and NSF Grants DMS-0611548 and OCI-0749217. Li was supported in part by the NSF Grant DMS-0702335 and DMS-0810506. Xu was supported in part by NSF of China Grant 10731060. Xu's visit to University of Texas at Arlington was supported in part by the NSF Grant DMS-0702335.

References

[1] Z. Bai, W. Chen, R.T. Scalettar, I. Yamazaki, Numerical methods for quantum Monte Carlo simulations of the Hubbard model, T. Hou et al., (Ed), Multi-scale Phenomena in Complex Fluids, Higher Education Press, 2009, pp. 1–110.
[2] R. Blankenbecler, D.J. Scalapino, R.L. Sugar, Monte Carlo calculations of coupled Boson-fermion systems I, Phys. Rev. D 24 (1981) 2278–2286.
[3] D.J. Scalapino, R.L. Sugar, Monte Carlo calculations of coupled Boson-fermion systems II, Phys. Rev. B 24 (1981) 4295–4308.
[4] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.
[5] J. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, 1997.
[6] J. Demmel, K. Veselić, Jacobi's method is more accurate than QR, SIAM J. Matrix Anal. Appl. 13 (1992) 1204–1245.
[7] J.W. Demmel, M. Gu, S.C. Eisenstat, I. Slapničar, K. Veselić, Z. Drmač, Computing the singular value decomposition with high relative accuracy, Linear Algebra Appl. 299 (1999) 21–80.
[8] Z. Drmač, K. Veselić, New fast and accurate Jacobi SVD algorithm. I, SIAM J. Matrix Anal. Appl. 29 (2008) 1322–1342.
[9] Z. Drmač, K. Veselić, New fast and accurate Jacobi SVD algorithm. II, SIAM J. Matrix Anal. Appl. 29 (2008) 1343–1362.
[10] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, Maryland, 1996.
[11] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM, Philadelphia, 2002.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023

[12] E.Y. Loh Jr., J.E. Gubernatis, R.T. Scalettar, R.L. Sugar, S.R. White, Stable matrix-multiplication algorithms for the low-temperature simulations of fermions, in: D. Baeriswyl, D.K. Campbell (Eds.), Interacting Electronics in Reduced Dimensions, Plenum, New York, 1989.
[13] E.Y. Loh Jr., J.E. Gubernatis, Stable numerical simulations of models of interacting electrons in condensed-matter physics, in: W. Hanke, Yu.V. Kopaev (Eds.), Electronic Phase Transitions, Elsevier Science Publishers B.V., 1992, pp. 177–235.
[14] E.Y. Loh Jr., J.E. Gubernatis, R.T. Scalettar, S.R. White, D.J. Scalapino, R.L. Sugar, Numerical stability and the sign problem in the determinant quantum Monte Carlo method, Internat. J. Modern Phys. 16 (2005) 1319–1322.
[15] A. Muramatsu, Quantum Monte Carlo for lattice fermions, in: M.P. Nightingale, C.J. Umriga (Eds.), Proceedings of the NATO Advanced Study Institute on Quantum Monte Carlo Methods in Physics and Chemistry, Kluwer Academic Publishers, 1999.
[16] G.W. Stewart, On graded QR decompositions of products of matrices, Electron. Trans. Numer. Anal. 3 (1995) 39–49.
[17] G.W. Stewart, Matrix Algorithms Vol. I: Basic Decompositions, SIAM, Philadelphia, 1998.
[18] G.W. Stewart, J.-G. Sun, Matrix Perturbation Theory, Academic Press, Boston, 1990.
[19] A. van der Sluis, Condition numbers and equilibration of matrices, Numer. Math. 14 (1969) 14–23.
[20] Future directions in tensor-based computation and modeling, NSF workshop report prepared by C. Van Loan. May 1, 2009. Available from: <http://www.cs.cornell.edu/cv/TenWork/FinalReport.pdf>.

Please cite this article in press as: Z. Bai et al., Stable solutions of linear systems involving long chain of matrix multiplications, Linear Algebra Appl. (2010), doi:10.1016/j.laa.2010.06.023