Numer. Math. (2017) 135:733–767
DOI 10.1007/s00211-016-0815-0

Numerische Mathematik

Highly accurate doubling algorithms for M-matrix algebraic Riccati equations

Jungong Xue[1] · **Ren-Cang Li**[2]

Received: 10 November 2015 / Revised: 18 April 2016 / Published online: 1 June 2016

Abstract The doubling algorithms are very efficient iterative methods for computing the unique minimal nonnegative solution to an M-matrix algebraic Riccati equation (MARE). They are globally and quadratically convergent, except for MARE in the critical case where convergence is linear with the linear rate $1/2$. However, the initialization phase and the doubling iteration kernel of any doubling algorithm involve inverting nonsingular M-matrices. In particular, for MARE in the critical case, the M-matrices in the doubling iteration kernel, although nonsingular, move towards singular M-matrices at convergence. These inversions are causes of concerns on entrywise relative accuracy of the eventually computed minimal nonnegative solution. Fortunately, a nonsingular M-matrix can be inverted by the GTH-like algorithm of Alfa et al. (Math Comp 71:217–236, 2002) to almost full entrywise relative accuracy, provided a triplet representation of the matrix is known. Recently, Nguyen and Poloni (Numer Math 130(4):763–792, 2015) discovered a way to construct triplet representations in a cancellation-free manner for all involved M-matrices in the doubling iteration kernel, for a special class of MAREs arising from Markov-modulated fluid queues. In this paper, we extend Nguyen's and Poloni's work to all MAREs by also devising a way to construct the triplet representations cancellation-free. Our construction, however, is not a straightforward extension of theirs. It is made possible by an introduction of novel recursively computable auxiliary nonnegative vectors. As the second contribution, we

✉ Ren-Cang Li
rcli@uta.edu

Jungong Xue
xuej@fudan.edu.cn

[1] School of Mathematical Science, Fudan University, Shanghai 200433, People's Republic of China

[2] Department of Mathematics, University of Texas at Arlington, P.O. Box 19408, Arlington, TX 76019-0408, USA

propose an entrywise relative residual for an approximate solution. The residual has an appealing feature of being able to reveal the entrywise relative accuracies of all entries, large and small, of the approximation. This is in marked contrast to the usual legacy normalized residual which reflects relative accuracies of large entries well but not so much those of very tiny entries. Numerical examples are presented to demonstrate and confirm our claims.

Mathematics Subject Classification 15A24 · 65F30 · 65H10

1 Introduction

An *M-Matrix Algebraic Riccati Equation* (MARE) is the matrix equation

$$XDX - AX - XB + C = 0, \tag{1.1}$$

for which A, B, C, D are matrices whose sizes are determined by the partitioning

$$W = \begin{array}{c} \\ m \\ n \end{array} \begin{array}{c} \begin{array}{cc} m & n \end{array} \\ \begin{bmatrix} B & -D \\ -C & A \end{bmatrix} \end{array}, \tag{1.2}$$

and W is a nonsingular or an irreducible singular M-matrix. This kind of Riccati equations arises in applied probability and transportation theory and has been extensively studied. See [11,12,14,15,17,18,22] and the references therein. It is shown in [12,14] that (1.1) has a unique minimal nonnegative solution Φ, i.e.,

$$\Phi \leq X \quad \text{for any other nonnegative solution } X \text{ of (1.1).}$$

Since (1.1) is a nonlinear equation, it admits more than one solution [19], but it is this minimal nonnegative solution Φ that is of the most practical importance.

Componentwise perturbation analysis [26] shows that small relative perturbations to all entries of A, B, C and D introduces small relative changes to the entries of Φ. Thus smaller entries of Φ do not suffer bigger relative errors than its larger entries. It is desirable to compute each entry of Φ, no matter how tiny it is, to high relative accuracy as the input data warrant. Indeed, there are several algorithms that, if implemented correctly, can compute such Φ to the accuracy as the theory predicts. They include several fix-point iterations [26,27] and the doubling algorithms [6,15,24]. The fix-point iterations [26] do not present much of implementation challenges as oppose to the doubling algorithms whose implementations involve inverting many nonsingular M-matrices, which, if not done carefully, leads to loss of accuracy in the tiny entries of the minimal nonnegative solution.

Thanks to Alfa et al. [1,2], it is now well-known that the inverse of an M-matrix is well-behaved under (certain structural) perturbations. They also devised a variation of Gaussian elimination without pivoting, called the GTH-like algorithm, to compute the inverse with almost full relative accuracy to all entries, no matter how tiny smaller

entries may be. But all of these require reparameterizing the M-matrix by the so-called *triple representation* and that each number in the representation is known correctly up to almost very last digits.

In [26, Remark 4.1], we outlined a very sketchy implementation detail for the *structure-preserving doubling algorithm* (SDA) [15]. The sketch applies to the newer doubling algorithms: SDA-ss [6] and ADDA [24], too, and usually work well but the sketch is not completely satisfactory as cancellations persist in calculating triple representations of the involved M-matrices in the doubling iteration kernel. Recently, Nguyen and Poloni [20] discovered a very robust implementation of the doubling algorithms for the special case where W satisfies

$$W\mathbf{1}_{m+n} = 0, \tag{1.3}$$

where $\mathbf{1}_{m+n}$ is the vector of all ones of dimension $m+n$ by discovering cancellation-free constructions of triple representations of the involved M-matrices. The main purpose of this paper is to extend the idea to cover all MAREs, i.e., when

W defined by (1.2) is a nonsingular M-matrix or an irreducible singular M-matrix. (1.4)

Our cancellation-free construction is, however, made possible by an introduction of novel recursively computable auxiliary nonnegative vectors that were not needed in the case of (1.3).

MARE (1.1) is said to be *in the critical case* if W is an irreducible singular M-matrix and $x_1^{\mathrm{T}} y_1 - x_2^{\mathrm{T}} y_2 = 0$, where $x_1, y_1 \in \mathbb{R}^m$ and $x_2, y_2 \in \mathbb{R}^n$ are positive vectors such that

$$W\boldsymbol{x} \equiv W \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0, \quad \boldsymbol{y}^{\mathrm{T}} W \equiv \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}^{\mathrm{T}} W = 0. \tag{1.5}$$

Among all MAREs, those in the critical case are the ones on which a doubling algorithm exhibits only linear convergence, whereas it will converge quadratically on all other MAREs.

Associated with MARE (1.1) is the complementary MARE

$$D - YA - BY + YCY = 0 \tag{1.6}$$

which can be symbolically obtained by multiplying both sides of (1.1) by X^{-1} and then setting $Y = X^{-1}$. This action is only symbolic because X may not be square, not to mention being possibly singular. This complementary equation is also an MARE and thus also has a unique minimal nonnegative solution which we will denote by Ψ.

The rest of paper is organized as follows. In Sect. 2, we summarize what one can expect from the GTH-like algorithm. Section 3 present our major contribution in this paper—constructing the needed triplet representations without cancellation for all nonsingular M-matrices arising the doubling algorithms for MARE (1.1). In Sect. 4, we propose a new entrywise relative residual to replace the commonly used legacy

normalized residual. With all preparations in the previous sections, we give details of our new highly accurate doubling algorithms in Sect. 5. Numerical examples are presented in Sect. 6. Conclusions are drawn in Sect. 7. Finally, in Appendix A we investigate the sparsity pattern in approximations by the doubling algorithms. The results are used to support the analysis in Sects. 3 and 4.

Notation $\mathbb{R}^{m\times m}$ is the set of all $m \times m$ real matrices, $\mathbb{R}^n = \mathbb{R}^{n\times 1}$, and $\mathbb{R} = \mathbb{R}^1$. I_n (or simply I if its dimension is clear from the context) is the $n \times n$ identity matrix. The superscript "$\cdot^{\mathrm{T}}$" takes transpose. For $X \in \mathbb{R}^{m\times m}$, $X_{(i,j)}$ refers to its (i, j)th entry, $|X|$ is in $\mathbb{R}^{m\times m}$ with its (i, j)th entry $|X_{(i,j)}|$. Inequality $X \le Y$ means $X_{(i,j)} \le Y_{(i,j)}$ for all (i, j), and similarly for $X < Y$, $X \ge Y$, and $X > Y$. In particular, $X \ge 0$ means that X is entrywise nonnegative. For a square matrix X, denote by $\rho(X)$ its spectral radius. $\mathbf{1}_n \in \mathbb{R}^n$ is the n-vector of all ones and $\mathbf{1}_{m\times n} \in \mathbb{R}^{m\times n}$ is the $m \times n$ matrix of all ones. The symbol $\mathfrak{u}$ is the unit machine roundoff.

2 The GTH-like algorithm for inverting a M-matrix

For a given nonsingular M-matrix A, Alfa et al. [1,2] proposed an alternative representation, the so-called *triplet representation* of A, and investigated its numerical advantages. In particular, the representation determines the smallest eigenvalue of A and A^{-1} entrywise to high relative accuracy. The reader is referred to [27, section 2] for a brief survey.

An M-matrix A can have infinite many triplet representations, but for the purpose of computation, any one is just as good as any other. A triplet representation $\{N_A, \boldsymbol{u}, \boldsymbol{v}\}$ of $A \in \mathbb{R}^{n\times n}$ consists of

$$N_A = \mathrm{diag}(A) - A, \quad 0 < \boldsymbol{u} \in \mathbb{R}^n, \quad \text{and} \quad \boldsymbol{v} = A\boldsymbol{u} \ge 0,$$

where $\mathrm{diag}(A)$ is the diagonal matrix obtained from extracting the diagonal part of A. For convenience, we will not distinguish A from its triplet representation and write $A = \{N_A, \boldsymbol{u}, \boldsymbol{v}\}$.

The main theoretical contribution in [2] is that if all entries of N_A, $\boldsymbol{u}$, and $\boldsymbol{v}$ are known to high relative accuracy, then all entries of A^{-1} are determined to a comparable high relative accuracy, or equivalently the solution x to $Ax = b \ge 0$ is determined to a comparable entrywise high relative accuracy. Numerically, Alfa et al. [1] presented the GTH-like algorithm to compute x to the claimed accuracy.

To save space, we will not give the detail of the GTH-like algorithm here, but explain the accuracy in the computed $\boldsymbol{x}$. To this end, we assume that we have an approximate triplet representation $\widehat{A} = \{N_{\widehat{A}}, \boldsymbol{u}, \hat{\boldsymbol{v}}\}$ to $A = \{N_A, \boldsymbol{u}, \boldsymbol{v}\}$ such that all entries of $N_{\widehat{A}}, \boldsymbol{u}, \hat{\boldsymbol{v}}$ are floating numbers and

$$\boldsymbol{u} > 0, \quad |\boldsymbol{v} - \hat{\boldsymbol{v}}| \le \epsilon\, \boldsymbol{v}, \quad |N_A - N_{\widehat{A}}| \le \epsilon N_A, \tag{2.1}$$

for some $0 < \epsilon < 1$. Consider now $A\boldsymbol{x} = \boldsymbol{b} \ge 0$ which is only known approximately as $\widehat{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$ with $\widehat{A} = \{N_{\widehat{A}}, \boldsymbol{u}, \hat{\boldsymbol{v}}\}$, and the latter will be solved by the GTH-like algorithm [1] to get an approximate solution $\widetilde{\boldsymbol{x}}$. According to [1,2,27], we have

$$|\widetilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}| \le [\phi(n)\mathfrak{u} + O(\mathfrak{u}^2)]\hat{\boldsymbol{x}}, \quad \frac{(1-\epsilon)^{n-1}}{(1+\epsilon)^n} A^{-1} \le \widehat{A}^{-1} \le \frac{(1+\epsilon)^{n-1}}{(1-\epsilon)^n} A^{-1},$$

where $\mathfrak{u}$ is the unit machine roundoff and $\phi(n) = 2(n+2)(n+3)(2n+5)/3$. Consequently,

$$\begin{aligned} |\widetilde{\boldsymbol{x}} - \boldsymbol{x}| &\le |\widetilde{\boldsymbol{x}} - \hat{\boldsymbol{x}}| + |\hat{\boldsymbol{x}} - \boldsymbol{x}| \\ &\le \left[\phi(n)\mathfrak{u} + O(\mathfrak{u}^2)\right]\hat{\boldsymbol{x}} + |\widehat{A}^{-1} - A^{-1}|b \\ &\le \left[\phi(n)\mathfrak{u} + O(\mathfrak{u}^2)\right]\frac{(1+\epsilon)^{n-1}}{(1-\epsilon)^n}\boldsymbol{x} + \left[\frac{(1+\epsilon)^{n-1}}{(1-\epsilon)^n} - 1\right]\boldsymbol{x} \\ &= \left[\phi(n)\mathfrak{u} + (2n-1)\epsilon + O(\mathfrak{u}^2 + \epsilon^2)\right]\boldsymbol{x}. \end{aligned} \tag{2.2}$$

We comment that $\phi(n) \approx 4n^3/3$ is a very pessimistic overestimate. In practice, it can be replaced by something much smaller, likely $O(n)$. It is what it is due to more the artifact of the proof [1] than anything else. In fact, such a phenomenon of overestimating error in a computational result much more than the actual error is is not uncommon, e.g., in the error analysis for using Gaussian elimination with pivoting to solve a linear system an $O(n^3)$ factor also appears [8]. It is now taken for granted that the most important thing in error bounds such as the one in (2.2) is not the polynomial factor $\phi(n)$ but rather the revelation of the leading error term $O(\mathfrak{u})$.

The critical foundation of the analysis above is the approximate triplet representation $\widehat{A} = \{N_{\widehat{A}}, \boldsymbol{u}, \hat{\boldsymbol{v}}\}$ that satisfies the inequalities in (2.1). The reader may notice that $\boldsymbol{u}$ is not perturbed. This is for a good reason. In practice, $\boldsymbol{u}$ either is known explicitly or has to be computed. In the latter, we referred the reader to the discussion in [27, subsection 2.2]. Another possible way to compute $\boldsymbol{u}$ is to use the self-corrective algorithms in [10]. In any case, we can always take $\boldsymbol{u}$ as being exact in the working floating point environment, $N_{\widehat{A}}$ as the rounded off-diagonal part of A and then let $\boldsymbol{v} = A\boldsymbol{u}$ and obtain its approximation $\hat{\boldsymbol{v}}$ by evaluating $\widehat{A}\boldsymbol{u}$ with care to ensure $|\boldsymbol{v} - \hat{\boldsymbol{v}}| \le \epsilon\, \boldsymbol{v}$, where $\widehat{A}$ is the rounded A in the working floating environment. This may be hard to do when some of the entries $\boldsymbol{v}_{(i)}$ involve catastrophic cancellations which are quite easy to detect actually. A quick error analysis shows

$$\frac{|\hat{\boldsymbol{v}}_{(i)} - \boldsymbol{v}_{(i)}|}{\boldsymbol{v}_{(i)}} \lesssim 3n \cdot \frac{\widehat{A}_{(i,i)}\boldsymbol{u}_{(i)} + \sum_{j\ne i} |\widehat{A}_{(i,j)}|\boldsymbol{u}_{(j)}}{\boldsymbol{v}_{(i)}} \cdot \mathfrak{u}, \tag{2.3}$$

where using "$\lesssim$" is for "less than or equal to" up to the first order. Assume $\hat{\boldsymbol{v}}_{(i)} \ge 0$; otherwise we know immediately either $\boldsymbol{u}$ is not good enough or A is unlikely an M-matrix. For all practical purpose, it is safe to estimate the fraction at the right-hand side by

$$\frac{\widehat{A}_{(i,i)}\boldsymbol{u}_{(i)} + \sum_{j\ne i} |\widehat{A}_{(i,j)}|\boldsymbol{u}_{(j)}}{\hat{\boldsymbol{v}}_{(i)}}.$$

If this ratio is unacceptably large, then a catastrophic cancellation has occurred and $\boldsymbol{v}_{(i)}$ has to be recomputed by a more accurate way via, e.g., some high precision package such as ARPREC [4]. Doing so carries additional cost, but it is marginal, compared to the computation of $\boldsymbol{u}$ and those afterwards.

3 Doubling algorithms

The original *structure-preserving doubling algorithm* (SDA) was proposed by Guo et al. [15] in 2006 when W is nonsingular. Then its two variants, SDA-ss [6] and ADDA [24], were discovered in 2010 and 2012, respectively. They are the most well-known and efficient algorithms for computing Φ, with ADDA being the fastest. These algorithms share the same doubling iteration kernel: for $k \geq 0$, iterate

$$E_{k+1} = E_k(I - Y_k X_k)^{-1} E_k, \tag{3.1a}$$

$$F_{k+1} = F_k(I - X_k Y_k)^{-1} F_k, \tag{3.1b}$$

$$Y_{k+1} = Y_k + E_k(I - Y_k X_k)^{-1} Y_k F_k, \tag{3.1c}$$

$$X_{k+1} = X_k + F_k(I - X_k Y_k)^{-1} X_k E_k, \tag{3.1d}$$

but differ only in their initial setups: constructing E_0, F_0, X_0 and Y_0. For this reason, we will call them collectively *structure-preserving doubling algorithms* (SDAs).

If these initial matrices are properly selected, as we will explain in the next subsection, it is guaranteed in theory that

1. all E_k, F_k, X_k, and Y_k are nonnegative[1] for $k \geq 0$,
2. all $I - X_k Y_k$ and $I - Y_k X_k$ are nonsingular M-matrices for $k \geq 0$, and
3. the sequences $\{X_k\}$ and $\{Y_k\}$ converge increasingly and quadratically to Φ and Ψ, respectively, except in the critical case. More details are in Theorem 3.1 below.

Numerically, by examining the expressions in (3.1), we see that the most delicate part lies in inverting the nonsingular M-matrices $I - Y_k X_k$ and $I - X_k Y_k$, for which we can use the GTH-like algorithm, provided their triplet representations are available. But in general they are not available. In [26, Remark 4.1], we suggested to use the idea in [27, subsection 2.2] to compute the desired triplet representations first before the GTH-like algorithm can be applied. Unfortunately there is no guarantee that a computed triplet representation this way would satisfy the perturbation model given in (2.1) with ϵ comparable to the machine roundoff, because there are cancellations in calculating the diagonal entries. Consequently, the accuracy in the computed $(I - Y_k X_k)^{-1}$ and $(I - X_k Y_k)^{-1}$ may not achieve the best possible or come to anywhere close to that. As pointed out in [20], such accuracy loss may not be reparable and thus can prevent doubling algorithms from converging in the componentwise sense.

[1] Previously in [24], E_0 and F_0 are nonpositive, but that is due to their assignments there and can be rectified through reassigning each of them to their corresponding opposites as we will do later in this paper for uniformity.

For the special case where W in (1.2) is an irreducible singular M-matrix with

$$W\mathbf{1}_{m+n} = 0, \tag{3.2}$$

it is discovered in [20] that explicit triplet representations of $I - X_k Y_k$ and $I - Y_k X_k$ can be constructed without cancellation. Consequently, the doubling algorithms can compute Φ to high componentwise relative accuracy, independent of the conditioning in inverting all $I - X_k Y_k$ and $I - Y_k X_k$.

In general, an irreducible singular M-matrix W may not necessarily have $\mathbf{1}_{m+n}$ as its (right) null vector. In the case when it is nonsingular, W has no (right) null vector at all. So the assumption (3.2) limits the applicability of the accurate implementation of SDAs in [20].

In this paper, our consideration is for all MAREs (1.1) with (1.4). Basically, we will establish a cancellation-free way to construct triplet representations for $I - X_k Y_k$ and $I - Y_k X_k$, provided a triplet representation $\{N_W, \boldsymbol{u}, \boldsymbol{v}\}$ of W:

$$\boldsymbol{u} > 0, \quad \boldsymbol{v} = W\boldsymbol{u} \geq 0 \tag{3.3}$$

is given, where $N_W = \mathrm{diag}(W) - W$ is the off-diagonal part of W. With the triplet representations and the GTH-like algorithm, the doubling algorithms will compute Φ as accurately as the input data, $W = \{N_W, \boldsymbol{u}, \boldsymbol{v}\}$, warrants. In practice, the triplet representation (3.3) either is known explicitly as in (3.2) or has to be computed [27, subsection 2.2]. Also, we won't have $\boldsymbol{v}$ but an approximation $\hat{\boldsymbol{v}}$ from evaluating $W\boldsymbol{u}$, accurately enough to ensure $|\hat{\boldsymbol{v}} - \boldsymbol{v}| = O(\epsilon)\boldsymbol{v}$ for some ϵ that is hopefully comparable to $\mathfrak{u}$. In what follows for clarity of presentation, we simply won't distinguish $\boldsymbol{v}$ from its numerical counterpart $\hat{\boldsymbol{v}}$ but assume $|\hat{\boldsymbol{v}} - \boldsymbol{v}| = O(\epsilon)\boldsymbol{v}$.

Our method of construction of the triplet representations for $I - X_k Y_k$ and $I - Y_k X_k$ relies on introducing recursively computable auxiliary nonnegative vectors which are far from obvious from the construction of Nguyen and Poloni [20].

3.1 Initialization

Poloni and Reis [20] reformulated the initialization in such a way that it elegantly unifies all initializations for the three doubling algorithms: SDA, SDA-ss, and ADDA. In what follows, we will outline their unification which also pleasantly paves the way for us to find triplet representations for $I - X_k Y_k$ and $I - Y_k X_k$ in a cancellation-free way.

We start by the original initialization of ADDA [24]. Select

$$\hat{\alpha} \geq \max_{1\leq i\leq n} A_{(i,i)}, \quad \hat{\beta} \geq \max_{1\leq j\leq m} B_{(j,j)},$$

and set

$$A_{\hat{\beta}} = A + \hat{\beta} I_n, \quad B_{\hat{\alpha}} = B + \hat{\alpha} I_m,$$
$$U_{\hat{\alpha}\hat{\beta}} = A_{\hat{\beta}} - C B_{\hat{\alpha}}^{-1} D, \quad V_{\hat{\alpha}\hat{\beta}} = B_{\hat{\alpha}} - D A_{\hat{\beta}}^{-1} C,$$

and[2]

$$\widehat{E}_0 = -I_m + (\hat{\alpha}+\hat{\beta})V_{\hat{\alpha}\hat{\beta}}^{-1}, \quad \widehat{F}_0 = -I_n + (\hat{\alpha}+\hat{\beta})U_{\hat{\alpha}\hat{\beta}}^{-1}, \tag{3.4a}$$

$$\widehat{Y}_0 = (\hat{\alpha}+\hat{\beta})B_{\hat{\alpha}}^{-1}DU_{\hat{\alpha}\hat{\beta}}^{-1}, \quad \widehat{X}_0 = (\hat{\alpha}+\hat{\beta})U_{\hat{\alpha}\hat{\beta}}^{-1}CB_{\hat{\alpha}}^{-1}. \tag{3.4b}$$

Without hats, $\hat{\alpha}$ and $\hat{\beta}$ are the same as the ones in [24]. We make such changes, adopted from [20, section 3], in order to leave the ones without hats for later use as, in fact, their reciprocals. Using [21, Theorem 5.1] (or [20, (15)]), we can combine (3.4a) and (3.4b) into one:

$$\begin{bmatrix} \widehat{E}_0 & \widehat{Y}_0 \\ \widehat{X}_0 & \widehat{F}_0 \end{bmatrix} = \begin{bmatrix} B+\hat{\alpha}I_m & -D \\ -C & A+\hat{\beta}I_n \end{bmatrix}^{-1} \begin{bmatrix} \hat{\beta}I_m - B & D \\ C & v\hat{\alpha}I_n - A \end{bmatrix}. \tag{3.5}$$

Now apply the doubling iteration kernel (3.1) with these $(\widehat{E}_0, \widehat{F}_0, \widehat{X}_0, \widehat{Y}_0)$ as the starting (E_0, F_0, X_0, Y_0) to produce the outputs E_k, F_k, X_k, and Y_k which we will rename as $\widehat{E}_k$, $\widehat{F}_k$, $\widehat{X}_k$, and $\widehat{Y}_k$ for $k \geq 1$.

Set $\alpha = \hat{\alpha}^{-1}$, $\beta = \hat{\beta}^{-1}$ and let

$$\begin{bmatrix} E_0 & Y_0 \\ X_0 & F_0 \end{bmatrix} = \begin{bmatrix} \hat{\alpha}I_m & \\ & \hat{\beta}I_n \end{bmatrix} \begin{bmatrix} \widehat{E}_0 & \widehat{Y}_0 \\ \widehat{X}_0 & \widehat{F}_0 \end{bmatrix} \begin{bmatrix} \beta I_m & \\ & \alpha I_n \end{bmatrix}.$$

It can be verified that

$$E_0 = \frac{\beta}{\alpha}\widehat{E}_0, \quad F_0 = \frac{\alpha}{\beta}\widehat{F}_0, \quad X_0 = \widehat{X}_0, \quad Y_0 = \widehat{Y}_0, \tag{3.6a}$$

and, by (3.5),

$$\begin{bmatrix} E_0 & Y_0 \\ X_0 & F_0 \end{bmatrix} = \begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix}^{-1} \begin{bmatrix} I_m - \beta B & \alpha D \\ \beta C & I_n - \alpha A \end{bmatrix}. \tag{3.6b}$$

Now apply the doubling iteration kernel (3.1) with this (E_0, F_0, X_0, Y_0) as the starting (E_0, F_0, X_0, Y_0) to produce the outputs E_k, F_k, X_k, and Y_k for $k \geq 1$. Note their notational differences from the ones previously generated with (3.4). Inductively, it can be shown that

$$E_k = \left(\frac{\beta}{\alpha}\right)^{2^k} \widehat{E}_k, \quad F_k = \left(\frac{\alpha}{\beta}\right)^{2^k} \widehat{F}_k, \quad \widehat{X}_k = X_k, \quad \widehat{Y}_k = Y_k.$$

In summary, the doubling iteration kernel (3.1) will generate the same X- and Y-sequences $\{X_k\}$ and $\{Y_k\}$, regardless of the initial input (E_0, F_0, X_0, Y_0) or $(\widehat{E}_0, \widehat{F}_0, \widehat{X}_0, \widehat{Y}_0)$.

[2] E_0 and F_0 are opposite in sign to the ones in [24] to make them entrywise nonnegative.

Previously in [24], the doubling iteration starts with the initialization given by (3.5). But using (E_0, F_0, X_0, Y_0) as given by (3.6b) can conveniently unify three main doubling algorithms in their implementation: SDA [15] (upon setting $\alpha = \beta$), SDA-ss [6] (uppon setting $\alpha = 0$ or $\beta = 0$), and ADDA [24] in general. More importantly, it is guaranteed that with (3.6b) E_k and F_k are uniformly bounded with respect to k (see Theorem 3.1 below) whereas one of $\widehat{E}_k$ and $\widehat{F}_k$ can be unbounded in the original implementation of ADDA (see [24, subsection 3.3]). In view of this, we adopt the initialization (3.6b) as suggested in [20], and ensure

$$0 \le \alpha \le \left[\max_{1\le i\le n} A_{(i,i)}\right]^{-1}, \quad 0 \le \beta \le \left[\max_{1\le j\le m} B_{(j,j)}\right]^{-1}, \quad \max\{\alpha, \beta\} \ne 0. \tag{3.7}$$

Introduce the *generalized Cayley transformation*

$$\mathscr{C}(M; \alpha, \beta) := (\alpha M - I)(\beta M + I)^{-1} \tag{3.8}$$

of a square matrix M. We purposely make this definition slightly differ from the one with the same notation in [24, (2.1)] in order to simplify the presentation in what follows. It can be proved along the lines of arguments in [15,24] that

$$E_k = (I - Y_k\Phi) \quad [\mathscr{C}(R; \beta, \alpha)]^{2^k}, \tag{3.9a}$$

$$\Phi - X_k = F_k\Phi \quad [\mathscr{C}(R; \beta, \alpha)]^{2^k}, \tag{3.9b}$$

$$\Psi - Y_k = E_k\Psi \quad [\mathscr{C}(S; \alpha, \beta)]^{2^k}, \tag{3.9c}$$

$$F_k = (I - X_k\Psi) \quad [\mathscr{C}(S; \alpha, \beta)]^{2^k}, \tag{3.9d}$$

where

$$R = B - D\Phi, \quad S = A - C\Psi. \tag{3.10}$$

We have the following theorem on the convergence of the doubling algorithms.

Theorem 3.1 *For MARE* (1.1)*, let* E_0, F_0, X_0, Y_0 *be as in* (3.6b) *with* (3.7)*,* R *and* S *as in* (3.10)*, and let the sequence* $\{(E_k, F_k, X_k, Y_k)\}$ *be generated by the doubling iteration kernel* (3.1)*. Then the following statements are true.*

1. *All* $E_k \ge 0$*,* $F_k \ge 0$*, and are uniformly bounded with respect to* k*.*
2. *All* $I - X_kY_k$ *and* $I - Y_kX_k$ *are nonsingular* M*-matrices.*
3. *All* X_k *have the same entrywise nonzero pattern*[3] *as* Φ*, and all* Y_k *have the same entrywise nonzero pattern as* Ψ*.*
4. $0 \le X_k \le X_{k+1} \le \Phi$*,* $0 \le Y_k \le Y_{k+1} \le \Psi$ *for all* $k \ge 0$*, and*

$$\limsup_{k\to\infty} \|\Phi - X_k\|^{1/2^k} \le \rho(\mathscr{C}(S; \alpha, \beta)) \cdot \rho(\mathscr{C}(R; \beta, \alpha)), \tag{3.11a}$$

$$\limsup_{k\to\infty} \|\Psi - Y_k\|^{1/2^k} \le \rho(\mathscr{C}(R; \beta, \alpha)) \cdot \rho(\mathscr{C}(S; \alpha, \beta)), \tag{3.11b}$$

[3] Two matrices X and Y are said to have the same entrywise nonzero pattern if $X_{(i,j)} = 0 \Leftrightarrow Y_{(i,j)} = 0$.

where $\|\cdot\|$ *is any matrix norm. Moreover, if* W *is also irreducible, then* $0 \le X_{k-1} < X_k < \Phi$, $0 \le Y_{k-1} < Y_k < \Psi$ *for* $k \ge 1$.

5. *In the critical case,* $\rho(\mathscr{C}(R;\beta,\alpha)) = \rho(\mathscr{C}(S;\alpha,\beta)) = 1$, *and* (X_k, Y_k) *converges to* (Φ, Ψ) *linearly with the linear rate* $1/2$.

Proof That all E_k and F_k are uniformly bounded is a corollary of Theorem 3.2 below. We leave the proof of item 3 to Appendix A. The rest of claims are the results of previous researches in [6,7,15,24], and thus will not be reproduced here. □

In [24], it is shown that $\rho(\mathscr{C}(S;\alpha,\beta)) \cdot \rho(\mathscr{C}(R;\beta,\alpha)) < 1$ if MARE (1.1) is not in the critical case, and it is 1 otherwise. The smallest $\rho(\mathscr{C}(S;\alpha,\beta)) \cdot \rho(\mathscr{C}(R;\beta,\alpha))$ subject to (3.7) is attained at

$$\alpha = \alpha_{\text{opt}} := \left[\max_{1\le i\le n} A_{(i,i)}\right]^{-1}, \quad \beta = \beta_{\text{opt}} := \left[\max_{1\le j\le m} B_{(j,j)}\right]^{-1}.$$

Therefore, the convergence of (X_k, Y_k) to (Φ, Ψ) is at least quadratic if MARE (1.1) is not in the critical case. For the critical case, however, the inequalities in (3.11) have no use in telling the convergence speed but item 5 says that (X_k, Y_k) goes to (Φ, Ψ) linearly with the linear rate $1/2$. It turns out that in the critical case, 0 is an eigenvalue of

$$\begin{bmatrix} I_m & \\ & -I_n \end{bmatrix} W = \begin{bmatrix} B & -D \\ C & -A \end{bmatrix}$$

of multiplicity 2 and associated with a 2×2 Jordan block. This particular eigenvalue is the contributing factor to the linear convergence. There are three existing methods proposed to speed up the convergence. Guo et al. [13] proposed to shift away the eigenvalue 0 to a properly chosen positive number before applying SDA. Later, Wang et al. [25] proposed to deflate out the eigenvalue 0 before applying ADDA. Theoretically, both approaches provably converge quadratically, provided no breakdown (i.e., all involved inverses exist) occurs. Basing on a detailed convergence analysis, Huang et al. [16] devised a two phase approach that effectively eliminates the linear convergence phase. All three methods achieve faster convergence, but cannot guarantee that all entries of Φ are computed to high relative accuracy when the entries of Φ vary widely in magnitude.

3.2 Triplet representations of $I - X_kY_k$ and $I - Y_kX_k$

In this section we discuss how to find triplet representations for $I - X_kY_k$ and $I - Y_kX_k$ in a cancellation-free way. Suppose that W in (1.2) is a nonsingular or irreducible singular M-matrix, and we also have

$$\boldsymbol{u} \equiv \begin{matrix} m \\ n \end{matrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} > 0, \quad \boldsymbol{v} \equiv \begin{matrix} m \\ n \end{matrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = W\boldsymbol{u} \ge 0. \tag{3.12}$$

Lemma 3.1 *Let $\alpha \geq 0$, $\beta \geq 0$ and $\max\{\alpha, \beta\} > 0$. Then*

$$\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix} \tag{3.13}$$

is a nonsingular M-matrix.

Proof No proof is necessary if both $\alpha = 0$ and $\beta = 0$. If $\alpha = 0$ but $\beta > 0$, then the matrix is

$$\begin{bmatrix} I_m & -\beta D \\ 0 & \beta A + I_n \end{bmatrix} = \begin{bmatrix} I_m & -D \\ 0 & A + \beta^{-1} I_n \end{bmatrix} \begin{bmatrix} I_m & 0 \\ 0 & \beta I_n \end{bmatrix}$$

which is a nonsingular M-matrix because its first matrix factor is a nonsingular M-matrix and the second matrix factor is diagonal with positive diagonal entries. Similarly, it can be showed that the matrix in (3.13) is a nonsingular M-matrix if $\alpha > 0$ but $\beta = 0$.

Suppose both $\alpha > 0$ and $\beta > 0$. Then

$$\begin{bmatrix} B + \alpha^{-1} I_m & -D \\ -C & A + \beta^{-1} I_n \end{bmatrix} = W + \begin{bmatrix} \alpha^{-1} I_m & \\ & \beta^{-1} I_n \end{bmatrix}$$

is a nonsingular M-matrix, since W is a nonsingular or irreducible singular M-matrix. Finally use

$$\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix} = \begin{bmatrix} B + \alpha^{-1} I_m & -D \\ -C & A + \beta^{-1} I_n \end{bmatrix} \begin{bmatrix} \alpha I_m & 0 \\ 0 & \beta I_n \end{bmatrix}$$

to conclude the proof. □

One advantage of the initialization (3.6b) with (3.7) is that E_k, F_k, X_k, Y_k are bounded for all k, as implied by the following theorem.

Theorem 3.2 *Let E_0, F_0, Y_0, X_0 be as in (3.6b) with (3.7). Then*

$$\begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{for all} \quad k \geq 0. \tag{3.14}$$

In particular, if $\boldsymbol{v} = 0$, then

$$\begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{for all} \quad k \geq 0. \tag{3.15}$$

Proof We only prove the inequality (3.14). The proof for the equality (3.15) can be obtained from that of (3.14) by changing all the inequalities into equalities. We use mathematical induction. For $k = 0$, we have by (3.12)

$$\begin{bmatrix} I_m - \beta B & \alpha D \\ \beta C & I_n - \alpha A \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - (\alpha + \beta) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$
$$\leq \begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{3.16}$$

By Lemma 3.1, $\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix}$ is a nonsingular M-matrix. Pre-multiply both sides of (3.16) by the nonnegative matrix $\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix}^{-1}$ and use (3.6b) to prove (3.14) for $k = 0$.

Suppose (3.14) holds for $k = j$. We need to show that it also holds for $k = j + 1$. By Theorem 3.1, $I - X_j Y_j$ and $I - Y_j X_j$ are nonsingular M-matrices, which implies that $\begin{bmatrix} I_m & -Y_j \\ -X_j & I_n \end{bmatrix}$ is also a nonsingualr M-matrix because

$$\begin{bmatrix} I_m & -Y_j \\ -X_j & I_n \end{bmatrix}^{-1} = \begin{bmatrix} (I - Y_j X_j)^{-1} & Y_j (I - X_j Y_j)^{-1} \\ (I - X_j Y_j)^{-1} X_j & (I - X_j Y_j)^{-1} \end{bmatrix} \geq 0. \tag{3.17}$$

By the induction hypothesis, we have

$$\begin{bmatrix} E_j & Y_j \\ X_j & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \left(I - \begin{bmatrix} 0 & Y_j \\ X_j & 0 \end{bmatrix} \right) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{3.18}$$

Pre-multiply both sides of the last inequality by the nonnegative matrix in (3.17) to get

$$\begin{bmatrix} I_m & -Y_j \\ -X_j & I_n \end{bmatrix}^{-1} \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{3.19}$$

Combining all the equations in (3.1) gives

$$\begin{bmatrix} E_{j+1} & Y_{j+1} \\ X_{j+1} & F_{j+1} \end{bmatrix} = \begin{bmatrix} 0 & Y_j \\ X_j & 0 \end{bmatrix} + \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} I_m & -Y_j \\ -X_j & I_n \end{bmatrix}^{-1} \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix}. \tag{3.20}$$

Now using (3.19), we have

$$\begin{aligned} \begin{bmatrix} E_{j+1} & Y_{j+1} \\ X_{j+1} & F_{j+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} &= \begin{bmatrix} 0 & Y_j \\ X_j & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} I_m & -Y_j \\ -X_j & I_n \end{bmatrix}^{-1} \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &\leq \begin{bmatrix} 0 & Y_j \\ X_j & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} E_j & 0 \\ 0 & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= \begin{bmatrix} E_j & Y_j \\ X_j & F_j \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \end{aligned}$$

as expected. □

Remark 3.1 A couple of comments are in order.

1. As a corollary, for the special case where W in (1.2) is a irreducible and singular M-matrix with $W\mathbf{1}_{m+n} = 0$, we deduce from this theorem that

$$\begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \begin{bmatrix} \mathbf{1}_m \\ \mathbf{1}_n \end{bmatrix} = \begin{bmatrix} \mathbf{1}_m \\ \mathbf{1}_n \end{bmatrix},$$

   a fact that was first proved in [20] by a probabilistic argument.
2. The initialization (3.6b) ensures that the matrices E_k, F_k, X_k, Y_k produced by the doubling iteration kernel (3.1) stay bounded in the sense of (3.14), whereas in the original implementation of ADDA [24, p. 182], $\widehat{E}_k$ or $\widehat{F}_k$ may go unbounded.

Now we are in a position to devise triplet representations for $I - X_k Y_k$ and $I - Y_k X_k$. We do this by tracking the differences

$$\begin{bmatrix} w_1^{(k)} \\ w_2^{(k)} \end{bmatrix} := \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geq 0, \tag{3.21}$$

where the nonnegativity is guaranteed by Theorem 3.2. However, we use this definition here only for derivation, not for actual computations, because if $w_i^{(k)}$ is evaluated as defined, the subtractions can prevent $w_i^{(k)}$ from being calculated with high relative accuracy and even its nonnegativity could be violated. Block-componentwise, (3.21) gives

$$w_1^{(k)} = u_1 - E_k u_1 - Y_k u_2, \tag{3.22a}$$
$$w_2^{(k)} = u_2 - X_k u_1 - F_k u_2. \tag{3.22b}$$

Solve (3.22a) for $Y_k u_2$ and (3.22b) for $u_2 - X_k u_1$ and use them in the following derivation:

$$\begin{aligned} (I_n - X_k Y_k)u_2 &= u_2 - X_k \left[u_1 - E_k u_1 - w_1^{(k)} \right] \\ &= \underbrace{u_2 - X_k u_1} + X_k E_k u_1 + X_k w_1^{(k)} \\ &= \underbrace{w_2^{(k)} + F_k u_2 + X_k (E_k u_1 + w_1^{(k)})}_{=: v_2^{(k)}} \geq 0, \end{aligned} \tag{3.23a}$$

and similarly

$$(I_m - Y_k X_k)u_1 = \underbrace{w_1^{(k)} + E_k u_1 + Y_k (F_k u_2 + w_2^{(k)})}_{=: v_1^{(k)}} \geq 0. \tag{3.23b}$$

With $v_i^{(k)}$ for $i = 1, 2$ defined in (3.23), the triplet representations for $I_n - X_k Y_k$ and $I_m - Y_k X_k$ can now be read off immediately from (3.23) as

$$I_m - Y_k X_k = \left\{ N_{I_m - Y_k X_k}, u_1, v_1^{(k)} \right\}, \tag{3.24a}$$

$$I_n - X_k Y_k = \left\{ N_{I_n - X_k Y_k}, u_2, v_2^{(k)} \right\}, \tag{3.24b}$$

without any cancellation, provided $w_1^{(k)}$ and $w_2^{(k)}$ are known. But they cannot be computed straightforwardly as defined in (3.21), otherwise cancellations could render inaccurate $w_i^{(k)}$ and even possibly violate their nonnegativity. So we need to find an alternative cancellation-free way to compute $w_1^{(k)}$ and $w_2^{(k)}$.

For the special case $\boldsymbol{v} = 0$, we have all $w_i^{(k)} = 0$ by Theorem 3.2 and thus by (3.23b)

$$v_1^{(k)} = E_k u_1 + Y_k F_k u_2, \quad v_2^{(k)} = F_k u_2 + X_k E_k u_1, \tag{3.25}$$

where no cancellation is possible. This is due to [20] for the case: $\boldsymbol{u} = \mathbf{1}$ and $\boldsymbol{v} = 0$.

Now we consider the most general case: $\boldsymbol{v} \geq 0$. For $k = 0$, we have by (3.6b)

$$\begin{bmatrix} w_1^{(0)} \\ w_2^{(0)} \end{bmatrix} = (\alpha + \beta) \begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix}^{-1} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{3.26}$$

The following lemma divides the case $\max\{\alpha, \beta\} \neq 0$ into three subcases to allow the GTH-like algorithm be applied to compute $w_i^{(0)}$ without any subtraction.

Lemma 3.2 *Suppose* (3.7) *holds.*

1. *If* $\alpha = 0$ *but* $\beta > 0$, *then*

$$\begin{bmatrix} I_m & -\beta D \\ 0 & I_n + \beta A \end{bmatrix}^{-1} = \begin{bmatrix} I_m & \beta D (I_n + \beta A)^{-1} \\ 0 & (I_n + \beta A)^{-1} \end{bmatrix},$$

and a triplet representation for $I_n + \beta A$ *can be read off from*

$$(I_n + \beta A) u_2 = u_2 + \beta (C u_1 + v_2).$$

2. *If* $\alpha > 0$ *but* $\beta = 0$, *then*

$$\begin{bmatrix} \alpha B + I_m & 0 \\ -\alpha C & I_n \end{bmatrix}^{-1} = \begin{bmatrix} (\alpha B + I_m)^{-1} & 0 \\ \alpha C (\alpha B + I_m)^{-1} & I_n \end{bmatrix},$$

and a triplet representation for $I_n + \beta A$ *can be read off from*

$$(\alpha B + I_m) u_1 = \alpha (v_1 + D u_2) + u_1.$$

3. If $\alpha > 0$ and $\beta > 0$, then

$$\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix} \begin{bmatrix} u_1/\alpha \\ u_2/\beta \end{bmatrix} = \begin{bmatrix} v_1 + u_1/\alpha \\ v_2 + u_2/\beta \end{bmatrix},$$

which yields a triplet representation for $\begin{bmatrix} \alpha B + I_m & -\beta D \\ -\alpha C & \beta A + I_n \end{bmatrix}$.

Finally we present a recursive formula to compute $w_i^{(k)}$ in a cancellation-free manner for $k \geq 1$, given $w_i^{(0)}$ by (3.26).

Theorem 3.3 *Let $w_i^{(k)}$ be defined by (3.21). Then*

$$\begin{aligned} w_1^{(k+1)} &= w_1^{(k)} + E_k(I_m - Y_k X_k)^{-1}\left[w_1^{(k)} + Y_k w_2^{(k)}\right], \\ w_2^{(k+1)} &= w_2^{(k)} + F_k(I_n - X_k Y_k)^{-1}\left[X_k w_1^{(k)} + w_2^{(k)}\right]. \end{aligned}$$

Proof The relation (3.20) can be rewritten as

$$\begin{aligned} \begin{bmatrix} E_{k+1} & Y_{k+1} \\ X_{k+1} & F_{k+1} \end{bmatrix} &= \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} - \begin{bmatrix} E_k & 0 \\ 0 & F_k \end{bmatrix} + \begin{bmatrix} E_k & 0 \\ 0 & F_k \end{bmatrix} \begin{bmatrix} I_m & -Y_k \\ -X_k & I_n \end{bmatrix}^{-1} \begin{bmatrix} E_k & \\ & F_k \end{bmatrix} \\ &= \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} - \begin{bmatrix} E_k & 0 \\ 0 & F_k \end{bmatrix} \begin{bmatrix} I_m & -Y_k \\ -X_k & I_n \end{bmatrix}^{-1} \left(I - \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \right). \end{aligned} \tag{3.27}$$

Post-multiplying both sides of (3.27) by $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ gives

$$\begin{bmatrix} E_{k+1} & Y_{k+1} \\ X_{k+1} & F_{k+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} E_k & 0 \\ 0 & F_k \end{bmatrix} \begin{bmatrix} I_m & -Y_k \\ -X_k & I_n \end{bmatrix}^{-1} \begin{bmatrix} w_1^{(k)} \\ w_2^{(k)} \end{bmatrix},$$

where we have used (3.21). Then,

$$\begin{bmatrix} w_1^{(k+1)} \\ w_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} w_1^{(k)} \\ w_2^{(k)} \end{bmatrix} + \begin{bmatrix} E_k & 0 \\ 0 & F_k \end{bmatrix} \begin{bmatrix} I_m & -Y_k \\ -X_k & I_n \end{bmatrix}^{-1} \begin{bmatrix} w_1^{(k)} \\ w_2^{(k)} \end{bmatrix},$$

which, combining with (3.17), prove the results. □

Remark 3.2 As $E_k(I_m - Y_k X_k)^{-1}$ and $F_k(I_n - X_k Y_k)^{-1}$ have been calculated during the doubling iteration kernel (3.1) without subtraction, updating $v_i^{(k)}$ can be done without subtraction at a negligible cost, comparing to that of the doubling iteration kernel.

4 Entrywise relative residual

Given an approximation $\widetilde{\Phi} \approx \Phi$, the following *normalized residual* in norm (NRes):

$$\mathrm{NRes}(\widetilde{\Phi}) = \frac{\|\widetilde{\Phi}D\widetilde{\Phi} - A\widetilde{\Phi} - \widetilde{\Phi}B + C\|}{\|\widetilde{\Phi}\|(\|\widetilde{\Phi}\|\|D\| + \|A\| + \|B\|) + \|C\|}, \tag{4.1}$$

is the commonly used legacy measure to gauge how accurate $\widetilde{\Phi}$ may be, because of its computational availablity, where $\|\cdot\|$ is some matrix norm, such as the ℓ_1 operator norm $\|\cdot\|_1$ [8] which is the one we will use later for its computational convenience. With conditions such as that NRes is sufficiently tiny, some multiple of it by a constant factor, called the *condition number*, can serve, up to the first order, as an upper bound on the *relative error* in norm (RErr):

$$\mathrm{RErr}(\widetilde{\Phi}) := \frac{\|\widetilde{\Phi} - \Phi\|}{\|\Phi\|}. \tag{4.2}$$

But such a bound in terms of the norm is good only in telling relative errors in the larger entries of $\widetilde{\Phi}$, those of $O(\|\widetilde{\Phi}\|_1)$ in magnitude. In the case where there is a wide variation in magnitude among the entries of Φ the bound is not able to yield any meaningful information on how accurate the smaller entries are. For example, suppose the error bound is $O(\epsilon)$ and suppose the ratio of the smallest entry over the largest entry in Φ is $O(\epsilon)$ or smaller, then the bound is useless in assessing the accuracy in the smallest entry and all other entries of comparable magnitudes. This is bad news since tiny entries also carry significant and useful probabilistic information.

In this section, we introduce a new residual, called the *entrywise relative residual* (ERRes) for MARE (1.1). This new ERRes reflects the non-negativeness property of the equation and is born out of the first order asymptotic error analysis in [26] for MARE (1.1).

Split A and B as

$$A = D_A - N_A, \quad D_A = \mathrm{diag}(A), \tag{4.3a}$$
$$B = D_B - N_B, \quad D_B = \mathrm{diag}(B). \tag{4.3b}$$

Rearrange MARE (1.1) with $X = \Phi$, the minimal nonnegative solution, to get

$$\mathscr{R}_L(\Phi) := \Phi D\Phi + N_A\Phi + \Phi N_B + C = D_A\Phi + \Phi D_B =: \mathscr{R}_R(\Phi). \tag{4.4}$$

An important outcome of such an arrangement is that there is no subtraction in evaluating both $\mathscr{R}_L(\Phi)$ and $\mathscr{R}_R(\Phi)$. Consider now nonnegative $\widetilde{\Phi}$ as an approximation to Φ. Instead of the equality in (4.4), now $\mathscr{R}_L(\widetilde{\Phi}) \neq \mathscr{R}_R(\widetilde{\Phi})$. In fact,

$$\mathscr{R}_L(\widetilde{\Phi}) = \mathscr{R}_R(\widetilde{\Phi}) + E, \tag{4.5a}$$

where $E = \widetilde{\Phi}D\widetilde{\Phi} - A\widetilde{\Phi} - \widetilde{\Phi}B + C$ which is the usual residual. Ideally we would like to have $E = 0$, but that's unlikely in practice. We define ERRes by

$$\mathrm{ERRes}(\widetilde{\Phi}) = \max_{\mathrm{i,j}} \frac{|\mathscr{R}_{\mathrm{L}}(\widetilde{\Phi}) - \mathscr{R}_{\mathrm{R}}(\widetilde{\Phi})|_{(\mathrm{i,j})}}{[\mathscr{R}_{\mathrm{R}}(\widetilde{\Phi})]_{(\mathrm{i,j})}}, \tag{4.5b}$$

where, as a convention, $0/0$ is treated as 0. By item 3 of Theorem 3.1, for approximations $\widetilde{\Phi}$ by the doubling algorithms, $\mathrm{ERRes}(\widetilde{\Phi}) < \infty$ always in exact arithmetic. Although the possibility that $\mathrm{ERRes}(\widetilde{\Phi}) = \infty$ exists numerically, it is extremely unlikely because each entry of $\widetilde{\Phi}$ is computed from 0 and then keeping adding a nonnegative number to the previous approximation as the doubling iteration progresses. So unless all these adding numbers are so tiny in magnitude (less than 10^{-308} in the IEEE double precision [3]) that they are rounded to 0, the entry of $\widetilde{\Phi}$ will be bigger than 0 whenever the corresponding entry of Φ is bigger than 0.

Suppose that $\widetilde{\Phi} \approx \Phi$ with high entrywise relative accuracy as we strive to achieve with doubling algorithms. Specifically, assume the *entrywise relative error* (ERErr) satisfies

$$\mathrm{ERErr}(\widetilde{\Phi}) := \max_{i,j} \frac{|(\widetilde{\Phi} - \Phi)_{(i,j)}|}{\Phi_{(i,j)}} \le \delta, \tag{4.6}$$

i.e.,

$$\widetilde{\Phi}_{(i,j)} = \Phi_{(i,j)}(1 + \epsilon_{ij}) \quad \text{with} \quad |\epsilon_{ij}| \le \delta < 1 \quad \text{for all} \quad i, j. \tag{4.7}$$

This ensures, in particular, $\widetilde{\Phi}$ and Φ have the same entrywise nonzero pattern. We would like to perform an error analysis to see how tiny E in (4.5a) could possibly get. To this end, we adopt the following floating point arithmetic model

$$\mathrm{fl}(\alpha \odot \beta) = (\alpha \odot \beta)(1 + \epsilon), \quad |\epsilon| \le \mathfrak{u} \quad \text{for} \quad \odot \in \{+, -, \times, \div\}, \tag{4.8}$$

where $\mathrm{fl}(\cdot)$ is the computed result of an expression. All today's commercially significant machines run the IEEE floating point arithmetic [3,9] and thus conform to (4.8).

Theorem 4.1 *Suppose that all entries of the coefficient matrices A, B, C, and D of MARE (1.1) are floating point numbers and suppose (4.7) holds. Then we have*

$$\mathrm{fl}(\mathscr{R}_L(\widetilde{\Phi})) = \mathrm{fl}(\mathscr{R}_R(\widetilde{\Phi})) + \widetilde{E},$$

with $\widetilde{E}$ satisfying

$$|\widetilde{E}| \le \Big[3mn\mathfrak{u} + 2mn\delta + O(\mathfrak{u}^2 + \delta^2 + \mathfrak{u}\delta)\Big] \cdot \mathscr{R}_R(\widetilde{\Phi}).$$

Proof Since both $\mathrm{fl}(\mathscr{R}_L(\widetilde{\Phi}))$ and $\mathrm{fl}(\mathscr{R}_R(\widetilde{\Phi}))$ involve no subtraction, we have

$$\mathrm{fl}(\mathscr{R}_L(\widetilde{\Phi})) = \mathscr{R}_L(\widetilde{\Phi}) + E_L, \quad |E_L| \le \Big[(3mn + 2m + 2n + 3)\mathfrak{u} + O(\mathfrak{u}^2)\Big]\mathscr{R}_L(\widetilde{\Phi}),$$
$$\mathrm{fl}(\mathscr{R}_R(\widetilde{\Phi})) = \mathscr{R}_R(\widetilde{\Phi}) + E_R, \quad |E_R| \le \Big[3\mathfrak{u} + O(\mathfrak{u}^2)\Big]\mathscr{R}_R(\widetilde{\Phi}).$$

Also we notice

$$\mathscr{R}_L(\widetilde{\Phi}) = \mathscr{R}_L(\Phi) + \widehat{E}_L, \quad |\widehat{E}_L| \le \left[(2mn + m + n)\delta + O(\delta^2)\right]\mathscr{R}_L(\Phi),$$
$$\mathscr{R}_R(\widetilde{\Phi}) = \mathscr{R}_R(\Phi) + \widehat{E}_R, \quad |\widehat{E}_R| \le \left[2\delta + O(\delta^2)\right]\mathscr{R}_R(\Phi).$$

Finally, use $\widetilde{E} = \mathrm{fl}(\mathscr{R}_L(\widetilde{\Phi})) - \mathrm{fl}(\mathscr{R}_R(\widetilde{\Phi})) = \widehat{E}_L + E_L - \widehat{E}_R - E_R$ and $\mathscr{R}_L(\Phi) = \mathscr{R}_R(\Phi)$ to conclude the proof. □

The next theorem says that if $\mathrm{ERRes}(\widetilde{\Phi})$ is sufficiently tiny, then some multiple of it by a constant factor, also called the *condition number* but in the entrywise sense, can tell entrywise relative accuracy in $\widetilde{\Phi}$ as an approximation to Φ, much like the role played by NRes in telling the relative error (4.2) of $\widetilde{\Phi}$ in norm.

Theorem 4.2 *Let $\widetilde{\Phi} \approx \Phi$ such that $\widetilde{\Phi}$ and Φ share the same entrywise nonzero pattern. Suppose MARE (1.1) is not in the critical case. If* $\mathrm{ERRes}(\widetilde{\Phi}) \le \epsilon$ *and if ϵ is sufficiently tiny, then*

$$|(\Phi - \widetilde{\Phi}) \oslash \Phi| \le \epsilon \Upsilon \oslash \Phi + O\left(\epsilon^2\right) \tag{4.9}$$
$$\le \gamma\epsilon\, \mathbf{1}_{n\times m} + O\left(\epsilon^2\right), \tag{4.10}$$

where $\oslash$ denotes the entrywise division, Υ and γ are defined by

$$(A - \Phi D)\Upsilon + \Upsilon(B - D\Phi) = D_A\Phi + \Phi D_B, \quad \gamma = \max_{i,j}(\Upsilon \oslash \Phi)_{(i,j)}. \tag{4.11}$$

Proof Write $\Delta\Phi = \Phi - \widetilde{\Phi}$ and subtract (4.5a) from (4.4), after rearrangement, to get

$$(A - \Phi D)(\Delta\Phi) + (\Delta\Phi)(B - D\Phi) = E + (\Delta\Phi)D(\Delta\Phi). \tag{4.12}$$

Define the following linear operator

$$\mathscr{L}_\Phi : X \to (A - \Phi D)X + X(B - D\Phi) \tag{4.13}$$

which is invertible and $\mathscr{L}_\Phi^{-1}$ is nonnegative in the sense that it maps nonnegative matrices into nonnegative ones [26, p. 675]. Following Stewart's argument [23, p. 242], we can see that for sufficiently tiny ϵ, (4.12) has a unique solution $\Delta\Phi = O(\epsilon)$. Therefore

$$|\Delta\Phi| \le \mathscr{L}_\Phi^{-1}\big(|E| + O(\epsilon^2)\big) = \epsilon\, \mathscr{L}_\Phi^{-1}\big(D_A\Phi + \Phi D_B\big) + O(\epsilon^2)$$

which yields (4.10) since Φ and $\widetilde{\Phi}$ have the same entrywise nonzero pattern. □

The quantity Υ and γ, introduced in [26], emerge from the first order error analysis of MARE (1.1) there. Naturally γ can be considered as the condition number of Φ as the solution to MARE (1.1). In general γ is not available, but can be estimated [26, Remark 2.1].

Example 4.1 Here we will use [20, Example 5.1] to numerically illustrate the superiority of ERRes over NRes in revealing ERErr. In this example, $m = n = 3$, and

$$A = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 15+\delta & -5 \\ 0 & -5 & 15 \end{bmatrix}, \quad B = \frac{1}{1.001}\begin{bmatrix} 15 & -5 & 0 \\ -5 & 15 & 0 \\ 0 & 0 & 5 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 0 & 4 \\ 5 & 5 & \delta \\ 5 & 5 & 0 \end{bmatrix}, \quad D = \frac{1}{1.001}\begin{bmatrix} 0 & 5 & 5 \\ 0 & 5 & 5 \\ 4 & 1 & 0 \end{bmatrix}, \tag{4.14}$$

where $\delta = 10^{-8}$. We compute the "exact" solution Φ by the computerized algebra system *Maple* with 100 decimal digits and we find that

$$1.7258 \cdot 10^{-9} \le \Phi_{(i,j)} \le 6.0999 \cdot 10^{-1}.$$

We purposely perturb this Φ to get $\widetilde{\Phi}$ such that 1) $\|\widetilde{\Phi} - \Phi\|_1$ is always about $O(\mathrm{u})$ and 2) ERErr varies from $O(\mathrm{u})$ to about 10^{-8} which is about what $\widetilde{\Phi}$ has if $\|\widetilde{\Phi} - \Phi\|_1 = O(\mathrm{u})$ in general. Specifically, we let in MATLAB

$$\widetilde{\Phi} = \Phi + \texttt{min(Phimax} * \texttt{rand(n, m)} * \texttt{eps, (Phi} * \texttt{eta)}. * \texttt{rand(n, m))},$$

where $\texttt{Phimax} = 6.0999 \cdot 10^{-1}$ is the largest entry in Φ and $\texttt{eta}$ varies from 10^{-14} to 10^{-8} so as to make ERErr vary from 10^{-15} to 10^{-8}. Figure 1 shows how NRes, RErr, ERRes, and ERErr change as $\texttt{eta}$ varies.

It clearly shows that ERRes and ERErr move in sync, whereas NRes and RErr remain "constant". This numerically demonstrates the capability of ERRes in revealing the entrywise relative accuracy in an approximation $\widetilde{\Phi}$, in addition to the theoretical justification we have in Theorem 4.2. In practice, since ERErr is not available because exact Φ is not known, ERRes is the perfect candidate to use because it is easily computable, just as we commonly use NRes and some comparable quantities in various numerical linear problems. ◊

According to Theorem A.1, the approximate solutions by the doubling iteration (3.1) with proper initialization always have the same entrywise nonzero pattern and thus with after enough iterations, the conditions of Theorem 4.2 should be satisfiable for ϵ about as tiny as $O(mn\mathrm{u})$, which in turn guarantees that the computed approximation X_k differs from Φ with an entrywise relative error about $O(mn\gamma\mathrm{u})$ for MARE not in the critical case. We point out that, as in most error analysis, the factor mn here likely overestimates the true effect of the matrix dimensions on the rounding errors. In practice, mn could be replaced by, e.g., $\max\{m, n\}$.

Theorem 4.2 only covers MARE (1.1) that is not in the critical case. For MARE (1.1) that is in the critical case, both $A - \Phi D$ and $B - D\Phi$ are singular M-matrices [25, Theorem 2.1] and consequently, it is unlikely for (4.12) to have a solution $\Delta\Phi = O(\epsilon)$ in general. In fact, when both $A - \Phi D$ and $B - D\Phi$ are singular, there exist nonzero vectors $\boldsymbol{w}$ and $\boldsymbol{z}$ such that

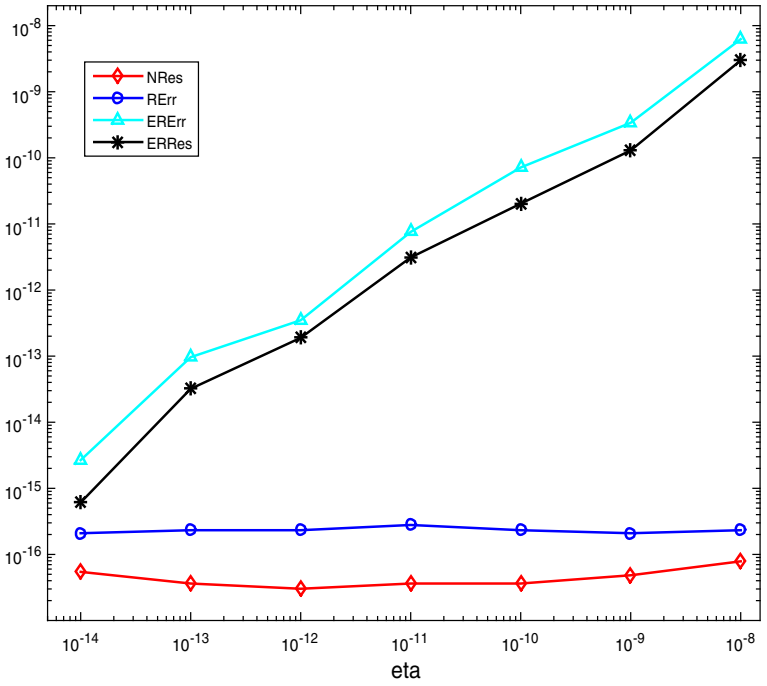

Fig. 1 Example 4.1. ERRes and ERErr move in sync, whereas NRes and RErr remain "constant"

$$\boldsymbol{w}^{\mathrm{T}}(A-\Phi D)=0,\quad (B-D\Phi)\boldsymbol{z}=0.$$

Normalize $\boldsymbol{w}$ and $\boldsymbol{z}$ to have unit norm, i.e., $\|\boldsymbol{w}\|_2=\|\boldsymbol{z}\|_2=1$, where $\|\cdot\|_2$ is the vector Euclidian norm or the matrix spectral norm. Pre- and post-multiply (4.12) by $\boldsymbol{w}^{\mathrm{T}}$ and $\boldsymbol{z}$, respectively, to get

$$\boldsymbol{w}^{\mathrm{T}}(\Delta\Phi)D(\Delta\Phi)\boldsymbol{z}=-\boldsymbol{w}^{\mathrm{T}}E\boldsymbol{z}\quad\Rightarrow\quad\|\Delta\Phi\|_2^2\geq\frac{|\boldsymbol{w}^{\mathrm{T}}(\Delta\Phi)D(\Delta\Phi)\boldsymbol{z}|}{\|D\|_2}=\frac{|\boldsymbol{w}^{\mathrm{T}}E\boldsymbol{z}|}{\|D\|_2},$$

suggesting that $\Delta\Phi$ be at least $O(\sqrt{\|E\|_2})$, unless $|\boldsymbol{w}^{\mathrm{T}}E\boldsymbol{z}|=o(\|E\|_2)$. Indeed, we will present a simple 2×2 MARE in Example 6.3 in the next section for which ERRes is $O(\mathrm{u})$ but $|\Delta\Phi|\oslash\Phi$ entrywise is $O(\sqrt{\mathrm{u}})$.

5 Highly accurate ADDA

With all the preparations, we are now ready to outline our highly accurate ADDA using the GTH-like algorithm to invert all nonsingular M-matrices along the way. This is done in Algorithm 5.1.

Three comments relevant to its implementation are in order:

1. Selecting α and β as in line 1 is to optimize the speed of convergence. In fact, they make $\rho(\mathscr{C}(S;\alpha,\beta))\cdot\rho(\mathscr{C}(R;\beta,\alpha))$ (cf. Theorem 3.1) the smallest, subject

Algorithm 5.1 Highly Accurate ADDA (accADDA)

Input: coefficient W in (1.2) of MARE (1.1), vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ that satisfy (3.12);
Output: the minimal nonnegative solutions Φ(, and Ψ if needed).

1: $\alpha = \left[\max_{1\le i\le n} A_{(i,i)}\right]^{-1}$, $\beta = \left[\max_{1\le j\le m} B_{(j,j)}\right]^{-1}$, $k = -1$;
2: compute E_0, F_0, X_0 and Y_0 by (3.6b), with the help of the GTH-like algorithm made possible by Lemma 3.2;
3: compute $w_1^{(0)}$ and $w_2^{(0)}$ by (3.26), with the help of the GTH-like algorithm made possible by Lemma 3.2;
4: **repeat**
5: $k = k + 1$;
6: compute $v_1^{(k)}$ and $v_2^{(k)}$ as defined in (3.23) and generate the triplet representations for $I - Y_k X_k$ and $I - X_k Y_k$ as in (3.24);
7: compute E_{k+1}, F_{k+1}, X_{k+1} and Y_{k+1} by (3.1) with the help of the GTH-like algorithm made possible by the triplet representations in (3.24);
8: compute $w_1^{(k+1)}$ and $w_2^{(k+1)}$ by the formulas in Theorem 3.3 (reuse $E_k(I_m - Y_k X_k)^{-1}$ and $F_k(I_n - X_k Y_k)^{-1}$ that appear in implementing line 7 to reduce work);
9: **until** convergence;
10: **return** the last X_k and. Y_k as approximations to Φ and Ψ, respectively.

to (3.7) [24, Theorem 2.3]. There is one possible bad consequence of the choice, however: it may incite cancellations in calculating some of the diagonal entries of $I_m - \beta B$ and $I_n - \alpha A$ in (3.6b). Such cancellations may potentially cause the initialization (E_0, F_0, X_0, Y_0) not accurate enough entrywise. One quick fix is to let α and β not be the optimal ones, but slightly less than that, i.e.,

$$\alpha = .9 \times \left[\max_{1\le i\le n} A_{(i,i)}\right]^{-1}, \quad \beta = .9 \times \left[\max_{1\le j\le m} B_{(j,j)}\right]^{-1},$$

instead of the ones in line 1. This strategy is similar to what we previously proposed in [26, p. 692].

2. Also at line 1, setting $\alpha = \beta$ to some proper value leads to the original SDA [15], and setting $\alpha = 0$ but β to some proper value or α to some proper value but $\beta = 0$ leads to SDA-ss [6].
3. We have to decide when to stop at line 9. There are three options:

$$|X_{k+1} - X_k| \le \epsilon \cdot X_{k+1}, \tag{5.1a}$$

$$\mathrm{ERRes}(\mathrm{X}_{\mathrm{k+1}}) \le \epsilon, \tag{5.1b}$$

$$\frac{(X_{k+1} - X_k)^2_{(i,j)}}{(X_k - X_{k-1})_{(i,j)} - (X_{k+1} - X_k)_{(i,j)}} \le \epsilon \cdot (X_{k+1})_{(i,j)} \quad \text{for all} \quad i \quad \text{and} \quad j, \tag{5.1c}$$

where ϵ is a pre-selected tolerance. The first one (5.1a) is the simplest one and the cheapest one to use, too, the second one (5.1b) is based on our newly proposed entrywise relative residual (4.5b), and the third one (5.1c) is Kahan's stopping criterion, previously in [24,26,27]. Both the simple (5.1a) and Kahan's stopping criterion (5.1c) can be too conservative in the case of a monotonically quadrati-

cally convergent sequence in the sense that they stop iterations unnecessarily late, wasting the last one or two iterations, and with the same ϵ, (5.1a) is even more conservative than (5.1c) because, in the phase of quadratic convergence,

$$(X_k - X_{k-1})_{(i,j)} - (X_{k+1} - X_k)_{(i,j)} \approx (X_k - X_{k-1})_{(i,j)} \gg (X_{k+1} - X_k)_{(i,j)},$$

and when $X_{k+1} - X_k = O([X_k - X_{k-1}]^2)$, the left hand side of (5.1c) is $O([(X_{k+1} - X_k)_{(i,j)}]^{3/2})$ which is much tinier than $(X_{k+1} - X_k)_{(i,j)}$. Another shortcoming for both is a possible pitfall: false-convergence in the sense that the iteration may be stopped due to a period of very slow moving X_k or just one particular iteration. The second stopping criterion (5.1b) is most expensive to use among the three, especially ERRes(X_{k+1}) is not needed in the doubling iteration kernel. But it does not have the pitfall mentioned above, and also when it is satisfied, the entrywise relative error in X_{k+1} is about $\gamma\epsilon$, where γ is defined in Theorem 4.2. In view of this discussion, we propose to use Kahan's stopping criterion (5.1c) with a safeguard, in the sense that when Kahan's stopping criterion is satisfied we check if (5.1b) (probably with a different ϵ) is also satisfied to avoid possible false-convergence. After numerous numerical experiments, we find that in the non-critical case ϵ about 10^{-10} to 10^{-12} works the best for computed solution to achieve its deserved entrywise relative accuracy about $O(10^{-15})$ without wasting the last iteration step (although not guaranteed; see the second MARE in Example 6.2 below). But in the critical case, ϵ should be set to about 10^{-14} to 10^{-16} because of the linear convergence.

In the case of $\boldsymbol{u} = \mathbf{1}_{m+n}$ and $\boldsymbol{v} = 0$, Algorithm 5.1 is basically the same as [20, Algorithm 2] except in the use of different stopping criterion at line 9 here. In [20, Algorithm 2], it is (5.1a) that was used.

6 Numerical examples

We will present several numerical examples to illustrate the superior performance of Algorithm 5.1 in delivering entrywise accuracy in computed $\widetilde{\Phi}$. In each example, we will plot history curves for entrywise relative residual ERRes defined by (4.5b), normalized residual NRes defined by (4.1), and the *entrywise relative error* ERErr defined by (4.6).

NRes is the commonly used legacy measure because it is readily available. Note that our new ERRes is equally readily available. In Sect. 4, we argued that in the case of MARE, ERRes should be preferred to NRes since ERRes and ERErr are in concert when it comes to measure the entrywise relative accuracy in $\widetilde{\Phi}$. All tests were done by MATLAB with $\mathfrak{u} = 2^{-53} \approx 1.1 \times 10^{-16}$. Kahan's stopping criterion (5.1c) is used with $\epsilon = 10^{-10}$ for MAREs not in the critical case and 10^{-15} for MAREs in the critical case. Previously, we recommend using Kahan's stopping criterion (5.1c) with a safeguard by checking the associated ERRes. But to better understand what is going on, in our numerical results below we report ERRes at each iteration step. Also for all examples, exact solutions Φ are either known explicitly or computed by

the computerized algebra system *Maple* with 100 decimal digits for testing purpose. Three variations of ADDA [24] are tested:

1. Algorithm 5.1 itself referred to as accADDA;
2. the *plain* ADDA which simply uses the usually Gaussian elimination with partial pivoting, such as MATLAB's operators "\" and "/", to carry out all the inversions in (3.6b) and (3.1)
3. the lite version of Algorithm 5.1, referred to as accADDA-lite, which simply sets, in view of (3.24),

$$v_1^{(k)} = (I - Y_k X_k)u_1, \quad v_2^{(k)} = (I - X_k Y_k)u_2 \tag{6.1}$$

to replace line 6 there.

The use of (6.1) carries a risk, i.e., some entries of computed $v_i^{(k)}$ may turn out to be negative, albeit tiny, due to roundoff errors, especially in the critical case where $(I - Y_k X_k)u_1$ and $(I - X_k Y_k)u_2$ converge to 0. Since in theory $v_i^{(k)} \geq 0$, if there are negative entries in computed $v_i^{(k)}$ by (6.1) then the value of such an entry must be comparable to the roundoff error in evaluating it. For this reason, as a safe-guard, we reset all negative entries, if any, in computed $v_i^{(k)}$ to 0.

In all of our tests, including many not reported here, the *plain* ADDA performs no better, often worse, than accADDA-lite which does no better than accADDA. To save space, we omit reporting numerical results by the plain ADDA.

Example 6.1 This is essentially the example of a positive recurrent Markov chain with nonsquare coefficients, originally from [5]. Here

$$A = 18 \cdot I_2, \quad B = 180002 \cdot I_{18} - 10^4 \cdot \mathbf{1}_{18\times 18}, \quad C = \mathbf{1}_{2\times 18}, \quad D = C^{\mathrm{T}}.$$

It is known $\Phi = \frac{1}{18} \cdot \mathbf{1}_{2\times 18} = \Psi^{\mathrm{T}}$, and $\boldsymbol{x} = \boldsymbol{y} = \mathbf{1}_{20}$ (see (1.5)). So $x_1^{\mathrm{T}} y_1 = 18 > x_2^{\mathrm{T}} y_2 = 2$ and thus the corresponding MARE is not in the critical case, doubling algorithms converge quadratically. This MARE falls into the class of MAREs dealt with in [20]. Iterative history curves are plotted in Figure 2. For this example, accADDA and accADDA-lite are indistinguishable. ◊

Example 6.2 This is a modification to the similar ones [15, Example 6.2], [26, Example 5.1], [24, Example 7.2], and [25, Example 7.1]. Here we like to create problems in the critical case and near the critical case. There are three MAREs to test, and each of them has a point to make. The first two MAREs come from

$$B = \begin{bmatrix} 3 & -1 & & \\ & 3 & \ddots & \\ & & \ddots & -1 \\ -1 & & & 3 \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad C = 2I_n, \quad A = \xi B, \quad D = \xi C, \tag{6.2}$$

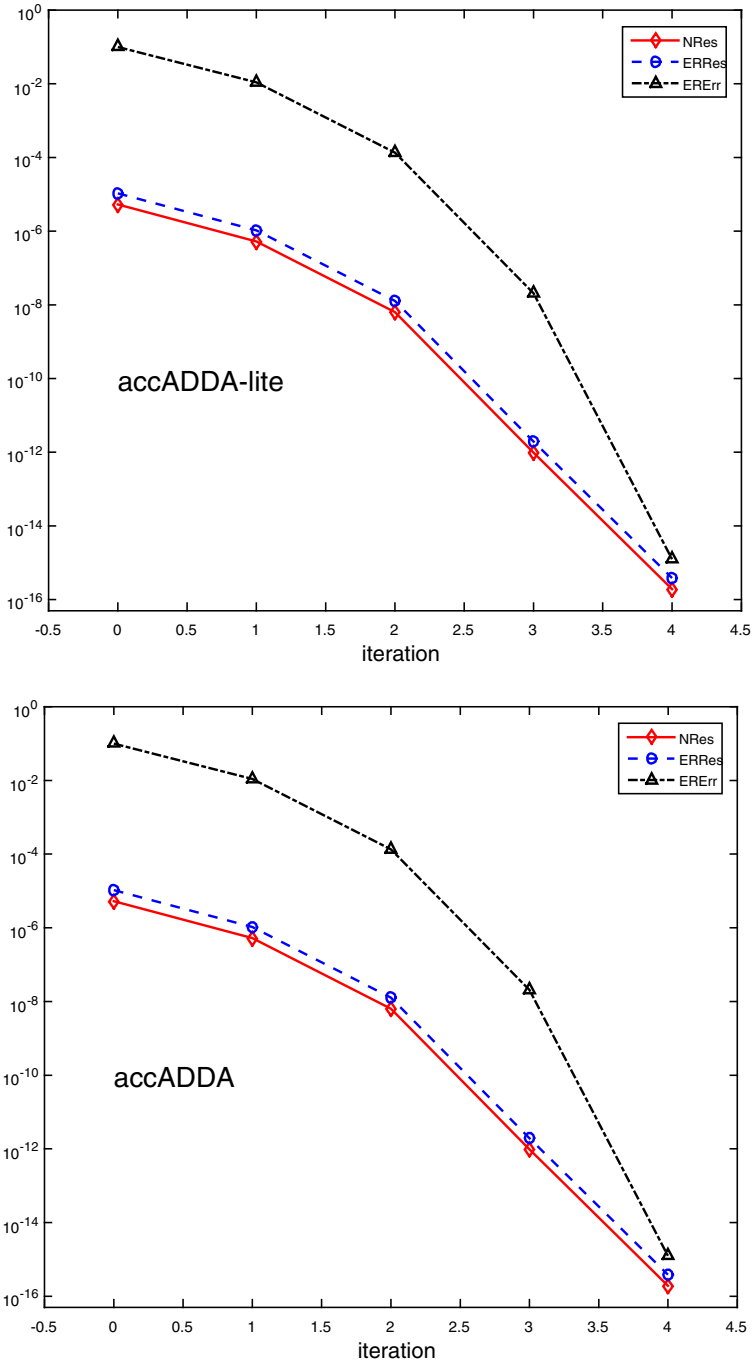

Fig. 2 Example 6.1. accADDA and accADDA-lite are indistinguishable.

where $\xi > 0$ is a parameter. W is an irreducible singular M-matrix:

$$W \begin{bmatrix} \mathbf{1}_n \\ \xi^{-1}\mathbf{1}_n \end{bmatrix} = 0, \quad \mathbf{1}_{2n}^{\mathrm{T}} W = 0.$$

So it is in the critical case if $\xi = 1$ and not in the critical case otherwise. We show in Fig. 3 the numerical results for $n = 100$ with $\xi = 1$ and 2^4, respectively. The "exact" solutions Φ is computed by the computerized algebra system *Maple* and

$$7.4339 \cdot 10^{-4} \leq \Phi_{(i,j)} \leq 3.8270 \cdot 10^{-1}, \quad \text{for} \quad \xi = 1, \tag{6.3}$$

$$1.3336 \cdot 10^{-35} \leq \Phi_{(i,j)} \leq 4.0231 \cdot 10^{-2}, \quad \text{for} \quad \xi = 2^4. \tag{6.4}$$

For $\xi = 1$, numerical results by accADDA-lite and the plain ADDA are nearly indistinguishable while for $\xi = 2^4$, all three implementations produce indistinguishable history curves.

In Fig. 3, the top two plots are for $\xi = 1$, the critical case. The convergence is clearly linear. While the curves for ERRes and NRes can reach the level about 10^{-15}, the ones for ERErr tell a different story. ERErr for accADDA-lite can only achieve 10^{-6}, but accADDA goes all the way to slightly better than 10^{-14}. It is interesting to notice that for accADDA-lite all three curves reach their respective lowest points at the same iterative step 26 while for accADDA the curve for ERErr steadily decreases to 1.1×10^{-14} even after both ERRes and NRes reach to $O(10^{-15})$ also at the iteration 26 and stay at that level ever after. This suggests that accADDA can even achieve nearly full entrywise relative accuracy, going against the analysis we had at the end of Sect. 4. The only possible explanation is that our accurately computed triplet representations must have structurally skewed subsequent rounding errors favorably in the way their contaminating effect on the solution is somehow damped. Note that the analysis did not take triplet representations into consideration. Turning to the case $\xi = 2^4$, we note from (6.4) there is a wide variation in magnitudes in Φ's entries. Such a variation prominently exposes the advantage of ERRes over NRes when it comes to correctly reveal the level of entrywise relative accuracies in computed X_k. In fact, the bottom two plots in Fig. 3 clearly show the curves for ERRes and ERErr match very well in shape.

The third MARE is an MARE, similar to (6.2), but with a nonsingular W and also near the critical case:

$$B = \begin{bmatrix} 3+\delta & -1 & & \\ & 3+\delta & \ddots & \\ & & \ddots & -1 \\ -1 & & & 3+\delta \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad C = 2I_n, \quad A = B, \quad D = C, \tag{6.5}$$

where $\delta > 0$ is a parameter. W is an irreducible nonsingular M-matrix:

$$W\mathbf{1}_{2n} = \delta \mathbf{1}_{2n}, \quad \mathbf{1}_{2n}^{\mathrm{T}} W = \delta \mathbf{1}_{2n}^{\mathrm{T}}.$$

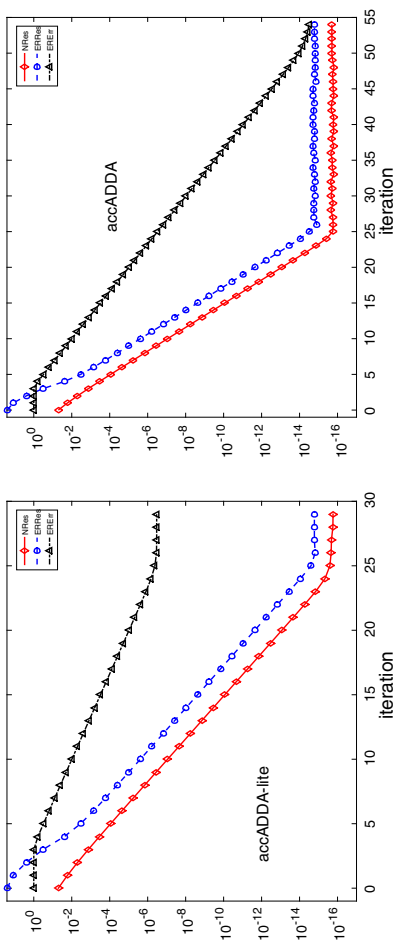

Fig. 3 For MARE (6.2) of Example 6.2. The *top* two plots are for $\xi = 1$, the critical case. The *bottom* two plots are for $\xi = 2^4$, a near critical case. The *curves* for ERRes and ERErr move in concert, demonstrating the capability of ERRes in correctly revealing the entrywise relative accuracies in the computed approximations. Kahan's stopping criterion with $\epsilon = 10^{-10}$ seems to cause one iteration too late since the last iteration does not bring any improvement in ERErr

The numerical results for $\delta = 2^{-24}$ are shown in Fig. 4. For this MARE, the plots for accADDA-lite and the plain ADDA are nearly indistinguishable. Clearly accADDA yields much more accurate $\widetilde{\Phi}$ than accADDA-lite does. In fact, accADDA gives best ERErr $= 2.1 \times 10^{-15}$ at iteration step 16 while accADDA-lite only gets its best ERErr $= 2.3 \times 10^{-12}$ also at iteration step 16. This suggests that it is worthwhile to go through all the troubles to compute $v_i^{(k)}$ recursively as accADDA demands than simply use (6.1).

MARE defined by (6.2) with $\xi \neq 1$ and that defined by (6.5) with $\delta \neq 0$ doesn't fall into the class of MAREs studied in [20]. ◊

Example 6.3 ([24, Example 7.1]) In this example, $m = n = 2$ and

$$B = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}, \quad D = \mathbf{1}_{2\times 2}, \quad A = B, \quad C = D.$$

Scaling W by 10^{-3} recovers a null recurrent case example in [5] (see also [13, Test 7.2]). It can be verified that

$$W\mathbf{1}_4 = 0, \quad \mathbf{1}_4^{\mathrm{T}} W = 0, \quad \Phi = \frac{1}{2}\mathbf{1}_{2\times 2}, \quad \Psi = \frac{1}{2}\mathbf{1}_{2\times 2}. \tag{6.6}$$

This example is small and special enough to allow us to explicitly construct $\widetilde{\Phi}$ that differ entrywise from Φ by $O(\sqrt{\mathrm{u}})$ while the corresponding NRes and ERRes are of $O(\mathrm{u})$. In fact, given $0 < \delta < 1/2$, let $\eta = 1/2 - \delta$ and $\widetilde{\Phi} = \eta\,\mathbf{1}_{2\times 2}$. We have

$$\mathscr{R}_L(\widetilde{\Phi}) = [4\eta^2 + 2\eta + 1]\mathbf{1}_{2\times 2}, \quad \mathscr{R}_R(\widetilde{\Phi}) = 6\eta\mathbf{1}_{2\times 2}.$$

Therefore $\mathscr{R}_L(\widetilde{\Phi}) - \mathscr{R}_R(\widetilde{\Phi}) = (2\eta - 1)^2\mathbf{1}_{2\times 2} = 4\delta^2\mathbf{1}_{2\times 2}$ and

$$\mathrm{NRes} = \frac{\delta^2}{(1-\delta)^2 + 1/2 - \delta}, \quad \mathrm{ERRes} = \frac{4\delta^2}{3 - 6\delta}, \quad \mathrm{ERErr} = 2\delta.$$

So if $\delta = O(\sqrt{\mathrm{u}})$, say 2^{-26}, then ERErr $= 2^{-25}$ only but already NRes $\approx 2^{-52.6}$ and ERRes $= 2^{-51.6}$. But numerically, iteration history curves for NRes, ERRes, and ERErr by the three variations of ADDA show similar behaviors as in the top two plots in Fig. 3 (that is why they are omitted). In particular, accADDA are still able to deliver X_k with ERRes being 10^{-15}, almost full entrywise relative accuracy. ◊

Example 6.4 ([20, Example 5.1]) This is the same one as in Example 4.1. Numerical results are shown in Fig. 5 which clearly says both accADDA and accADDA-lite reach their respective best in the same number of doubling iterations but their best entrywise relative accuracies differ: 4.3×10^{-16} by accADDA and 6.7×10^{-13} by accADDA-lite. The result by accADDA on this example should be the same as that by [20, Algorithm 2]. ◊

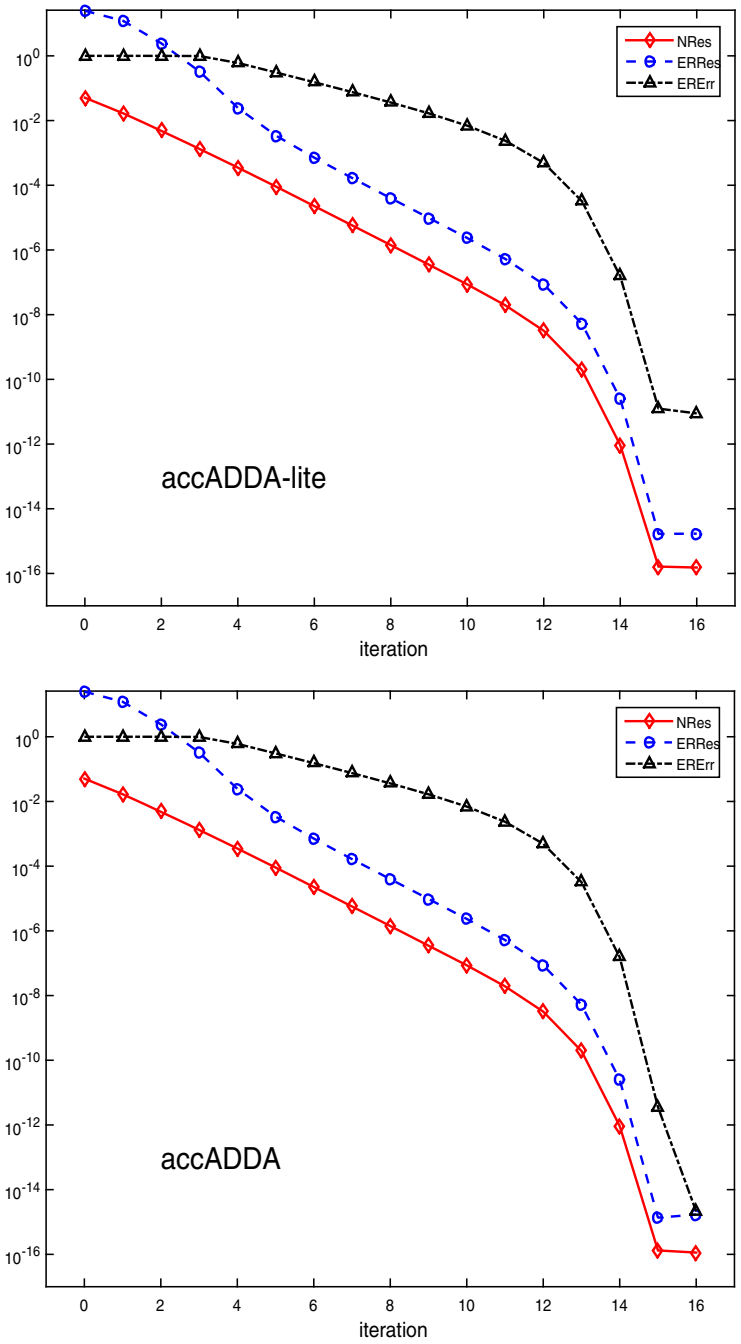

Fig. 4 For (6.5) of Example 6.2 with $\delta = 2^{-24}$. Three more accurate decimal digits in the approximation by accADDA than the one by accADDA-lite suggests that it is worthwhile to go through all the troubles to compute $v_i^{(k)}$ recursively than simply use (6.1). It is interesting to ntoice that between iteration steps 15 and 16 of accADDA, ERRes stays about the same but ERErr improves by about 10^{-3}

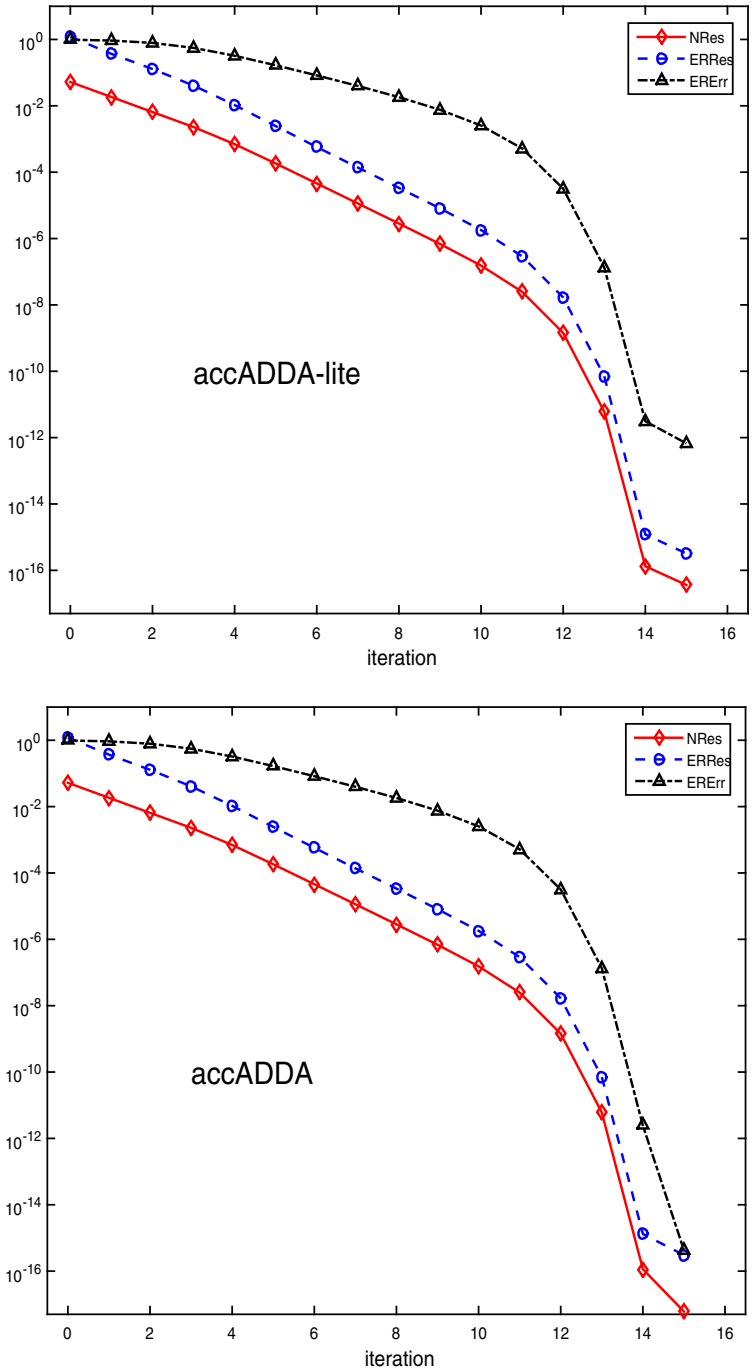

Fig. 5 Example 6.4. At convergence, ERErr $= 4.3 \times 10^{-16}$ by accADDA and 6.7×10^{-13} by accADDA-lite. It is interesting to ntoice that between iteration steps 14 and 15 of accADDA, ERRes improves about 10^{-1} but ERErr improves by about 10^{-3}

Table 1 ERErr and ERRes at convergence for all examples, and γ (cf. Theorem 4.2) for those not in the critical case

Eg.	AccADDA		AccADDA-lite		Plain ADDA		γ
	ERErr	ERRes	ERErr	ERRes	ERErr	ERRes	
1	1.2×10^{-15}	3.9×10^{-16}	1.2×10^{-15}	3.9×10^{-16}	4.5×10^{-13}	3.9×10^{-16}	1.0×10^{4}
2(1)	3.1×10^{-15}	1.5×10^{-15}	3.4×10^{-7}	1.5×10^{-15}	1.5×10^{-7}	1.5×10^{-15}	–
2(2)	8.6×10^{-15}	1.2×10^{-15}	1.9×10^{-14}	1.0×10^{-15}	2.1×10^{-14}	1.1×10^{-15}	7.4×10^{1}
2(3)	2.1×10^{-15}	1.7×10^{-15}	8.9×10^{-12}	1.7×10^{-15}	5.9×10^{-12}	1.7×10^{-14}	1.1×10^{3}
3	5.5×10^{-16}	1.5×10^{-16}	8.5×10^{-9}	1.5×10^{-16}	1.7×10^{-8}	1.5×10^{-16}	–
4	4.3×10^{-16}	3.1×10^{-16}	6.7×10^{-13}	3.2×10^{-16}	3.5×10^{-10}	4.5×10^{-11}	6.2×10^{2}

Eg. 2 (i) for $i = 1, 2, 3$ refers to the three MAREs in Example 6.2 with $\xi = 1, 2^4$, and $\delta = 2^{-24}$, respectively

Finally, we summarize in Table 1 ERErr and ERRes at convergence for all examples, and γ for those not in the critical case. It is interesting to note that accADDA delivers approximations with both ERErr and ERRes at the level of $O(\mathrm{u})$ at convergence for all examples, all ERRes for the approximations by accADDA-lite are also at the level of $O(\mathrm{u})$ but the associated ERErr varies, with $\gamma \times$ ERRes about $O(\mathrm{u})$, and the plain ADDA performs the worst (for Exampe 6.4 in particular). This observation makes us wonder if accADDA can always yield an approximation with almost full entrywise relative accuracy, regardless the underlying MARE being ill-conditioned (i.e., large γ) or well-conditioned or even in the critical case. A precise error analysis eludes us. Another conjecture we have is whether the minimal nonnegative solution of MARE (1.1) changes entrywise relatively by at most $\phi(m,n)\epsilon$ when the triplet representation $W = \{N_W, \boldsymbol{u}, \boldsymbol{v}\}$ is perturbed to $\widehat{W} = \{N_{\widehat{W}}, \boldsymbol{u}, \hat{\boldsymbol{v}}\}$ satisfying

$$\boldsymbol{u} > 0, \quad |\boldsymbol{v} - \hat{\boldsymbol{v}}| \le \epsilon\, \boldsymbol{v}, \quad |N_W - N_{\widehat{W}}| \le \epsilon N_W,$$

where $\phi(m,n)$ is some low degree polynomial of m and n.

In [20, Theorem 4.4], an estimate on $|X_k - \Phi|$ was established and it involves constants that, unfortunately, are not known a priori, and depend on the underlying MARE. Much of the convergence analysis [20] can be modified to work in our content here to give an estimate similar to Theorem 4.4 there, but detail is omitted here to save space. The estimate so established does not confirm, nor reject, the conjectures we just mentioned since it depends on the underlying MARE.

7 Conclusions

The structure-preserving doubling algorithms (SDAs) are the most efficient methods for computing the unique minimal nonnegative solution of MARE (1.1). They are globally and quadratically convergent, except in the critical case where the convergence is still globally but only linearly with the linear rate $1/2$. But their implementations involve inverting nonsingular M-matrices $I - X_kY_k$ and $I - Y_kX_k$ in the doubling iteration kernel. Finding triplet representations for these M-matrices in the most possibly accurate way holds the key to a correct implementation. For the special case $W\mathbf{1}_{m+n} = 0$, Nguyen and Poloni [20] cleverly show that triplet representations for $I - X_kY_k$ and $I - Y_kX_k$ can be constructed in a cancellation-free manner. Note that $W\mathbf{1}_{m+n} = 0$ yields a natural triplet representation of W.

In general when W is nonsingular or singular but $W\mathbf{1}_{m+n} \neq 0$, it is not so clear whether one can construct triplet representations for the nonsingular M-matrices $I - X_kY_k$ and $I - Y_kX_k$ in the doubling iteration kernel without actually computing them, even if a triplet representation for W is given. Our first and main contribution in this paper is to show that this indeed can be done and how to do it. We do so through introducing recursively computable auxiliary nonnegative vectors without any cancellation and consequently calculating triplet representations for $I - X_kY_k$ and $I - Y_kX_k$ without the need to calculate their diagonal entries which, if calculated directly, carries a risk of causing irreparable loss of accuracy later on.

Our second contribution is the proposal of the new entrywise relative residual (4.5b) whose magnitude reflects the entrywise relative error (4.6), in the similar spirit to the usually legacy normalized residual (4.1) that only reflects the relative error in norm (4.2) well.

Acknowledgements The authors are grateful to both reviewers for their constructive comments/suggestions that improve the presentation considerably. Xue is supported in part by NSFC Grant 11371105 and Laboratory of Mathematics for Nonlinear Science, Fudan University. Li is supported in part by NSF Grants DMS-1317330 and CCF-1527104, and NSFC Grant 11428104.

Appendix: Sparsity of E_k, F_k, X_k, Y_k

For convenience, we introduce a partial ordering on nonnegative matrices with respect to their entrywise nonzero patterns. For matrices $P \geq 0, Q \geq 0$ of the same size, we say that Q majorizes P in the entrywise nonzero pattern, written as $P \overset{0}{\preceq} Q$, if $Q_{(i,j)} = 0$ implies $P_{(i,j)} = 0$, and write $P \overset{0}{=} Q$ if $P \overset{0}{\preceq} Q$ and $Q \overset{0}{\preceq} P$. Evidently, $0 \leq P \leq Q$ implies $P \overset{0}{\preceq} Q$, but not the other way around.

Lemma A.1 *1. Given $0 \leq P_i \leq Q_i$ for $i = 1, 2$, all of the same size, we have*

$$P_1 + P_2 \overset{0}{\preceq} Q_1 + Q_2, \quad P_1 P_2 \overset{0}{\preceq} Q_1 Q_2.$$

2. Given $0 \leq P \overset{0}{\preceq} Q$, we have $P + Q \overset{0}{=} Q$.

Lemma A.2 *Let P, Q be nonsingular M-matrices, and split P and Q as*

$$P = D_P - N_P, \quad D_P = \mathrm{diag}(P), \tag{A.1a}$$
$$Q = D_Q - N_Q, \quad D_Q = \mathrm{diag}(Q). \tag{A.1b}$$

If $N_P \overset{0}{\preceq} N_Q$, then $P^{-1} \overset{0}{\preceq} Q^{-1}$. In particular, if $N_P \overset{0}{=} N_Q$, then $P^{-1} \overset{0}{=} Q^{-1}$.

Proof Since P, Q are nonsingular M-matrices, P^{-1} and Q^{-1} admit the following series respresentations

$$P^{-1} = D_P^{-1} \sum_{k=0}^{\infty} \left(N_P D_P^{-1} \right)^k, \quad Q^{-1} = D_Q^{-1} \sum_{k=0}^{\infty} \left(N_Q D_Q^{-1} \right)^k. \tag{A.2}$$

If $N_P \overset{0}{\preceq} N_Q$, then $(N_P D_P^{-1})^k \overset{0}{=} N_P^k \overset{0}{\preceq} N_Q^k \overset{0}{=} (N_Q D_Q^{-1})^k$ for all $k \geq 0$ and thus $P^{-1} \overset{0}{\preceq} Q^{-1}$. In the case when $N_P \overset{0}{=} N_Q$, we also have $N_Q \overset{0}{\preceq} N_P$ and thus $Q^{-1} \overset{0}{\preceq} P^{-1}$ at the same time. □

Lemma A.3 *For a nonsingular M-matrix P, $P^{-1} \overset{0}{=} P^{-k}$ for $k \geq 1$, and $(\alpha I - P^{-1})^{-1} \overset{0}{=} P^{-1}$ for $\alpha > \rho(P^{-1})$.*

Proof In this proof and that of Lemma A.4 later, the series $\sum_{i=0}^{\infty} N_P^i$ is used only symbolically for its entrywise nonzero pattern since the series itself may not even converge. Again we have (A.2) which yields that $P^{-1} \overset{0}{=} \sum_{i=0}^{\infty} N_P^i$, and that

$$P^{-k} = (P^{-1})^k \overset{0}{=} \left(\sum_{i=0}^{\infty} N_P^i\right)^k \overset{0}{=} \sum_{i=0}^{\infty} N_P^i \overset{0}{=} P^{-1}.$$

For $\alpha > \rho(P^{-1})$, we have $(\alpha I - P^{-1})^{-1} = \alpha^{-1} \sum_{i=0}^{\infty} (\alpha P)^{-i} \overset{0}{=} P^{-1}$, as expected. □

Lemma A.4 *Let P be a nonsingular M-matrix and $Q \geq 0$, and split P, Q as*

$$P = D_P - N_P, \quad D_P = \mathrm{diag}(P), \tag{A.3a}$$

$$Q = D_Q + \widetilde{N}_Q, \quad D_Q = \mathrm{diag}(Q). \tag{A.3b}$$

If all of the diagonal entries of D_Q are positive and $\widetilde{N}_Q \overset{0}{\preceq} N_P$, then

$$P^{-1}Q \overset{0}{=} P^{-1}.$$

Proof Since $P^{-1} \geq 0$ and $Q \geq 0$, $P^{-1} \overset{0}{=} P^{-1}D_Q \overset{0}{\preceq} P^{-1}D_Q + P^{-1}\widetilde{N}_Q = P^{-1}Q$. On the other hand,

$$P^{-1}Q \overset{0}{=} P^{-1} + P^{-1}\widetilde{N}_Q \overset{0}{\preceq} P^{-1} + P^{-1}N_P \overset{0}{=} \sum_{i=0}^{\infty} N_P^i \overset{0}{=} P^{-1},$$

as was to be shown. □

Theorem A.1 *Let E_0, F_0, X_0, Y_0 be as in (3.6b), and let E_k, F_k, X_k, Y_k be produced by the doubling iteration (3.1). Then for $k \geq 0$,*

$$\begin{bmatrix} E_{k+1} & Y_{k+1} \\ X_{k+1} & F_{k+1} \end{bmatrix} \overset{0}{\preceq} \begin{bmatrix} E_k & Y_k \\ X_k & F_k \end{bmatrix}, \tag{A.4}$$

$X_{k+1} \overset{0}{=} X_k \overset{0}{=} \Phi$, and $Y_{k+1} \overset{0}{=} Y_k \overset{0}{=} \Psi$.

Proof It is clear that $X_k \overset{0}{\preceq} X_{k+1}$ and $Y_k \overset{0}{\preceq} Y_{k+1}$ because of (3.1c) and (3.1d). So if (A.4) is proven true, then we will immediately have $X_{k+1} \overset{0}{=} X_k \overset{0}{=} \Phi$ and $Y_{k+1} \overset{0}{=} Y_k \overset{0}{=} \Psi$.

It remains to prove (A.4). Because of (3.6), it suffices to prove (A.4) with a hat over every symbol. Set $Q = \begin{bmatrix} N_B & D \\ C & N_A \end{bmatrix}$. We have

$$\begin{bmatrix} B + \hat{\alpha} I_m & -D \\ -C & A + \hat{\beta} I_n \end{bmatrix} = \begin{bmatrix} D_B + \hat{\alpha} I_m & 0 \\ 0 & D_A + \hat{\beta} I_n \end{bmatrix} - Q,$$

$$\begin{bmatrix} \hat{\beta} I_m - B & D \\ C & \hat{\alpha} I_n - A \end{bmatrix} = \begin{bmatrix} \hat{\beta} I_m - D_B & 0 \\ 0 & \hat{\alpha} I_n - D_A \end{bmatrix} + Q.$$

Noting (3.5) and using Lemmas A.2 and A.4, we see

$$\begin{bmatrix} \widehat{E}_0 & Y_0 \\ X_0 & \widehat{F}_0 \end{bmatrix} \overset{0}{=} (\gamma_1 I - Q)^{-1}$$

for $\gamma_1 > \rho(Q)$. By Lemmas A.2 and A.3,

$$\begin{bmatrix} I_m & -Y_0 \\ -X_0 & I_n \end{bmatrix}^{-1} \overset{0}{\preceq} \left(\begin{bmatrix} \gamma_2 I_m & 0 \\ 0 & \gamma_2 I_n \end{bmatrix} - \begin{bmatrix} \widehat{E}_0 & Y_0 \\ X_0 & \widehat{F}_0 \end{bmatrix} \right)^{-1} \overset{0}{=} (\gamma_1 I - Q)^{-1}$$

for γ_2 large enough. Using (3.20) with $k = 0$ and noting

$$\begin{bmatrix} 0 & Y_0 \\ X_0 & 0 \end{bmatrix} \overset{0}{\preceq} (\gamma_1 I - Q)^{-1}, \quad \begin{bmatrix} \widehat{E}_0 & 0 \\ 0 & \widehat{F}_0 \end{bmatrix} \overset{0}{\preceq} (\gamma_1 I - Q)^{-1},$$

we have

$$\begin{aligned} \begin{bmatrix} \widehat{E}_1 & Y_1 \\ X_1 & \widehat{F}_1 \end{bmatrix} &\overset{0}{\preceq} (\gamma_1 I - Q)^{-1} + (\gamma_1 I - Q)^{-1} (\gamma_1 I - Q)^{-1} (\gamma_1 I - Q)^{-1} \\ &\overset{0}{=} (\gamma_1 I - Q)^{-1} + (\gamma_1 I - Q)^{-1} \\ &\overset{0}{=} (\gamma_1 I - Q)^{-1} \\ &\overset{0}{=} \begin{bmatrix} \widehat{E}_0 & Y_0 \\ X_0 & \widehat{F}_0 \end{bmatrix}. \end{aligned}$$

This proves (A.4) for $k = 0$.

Now suppose (A.4) holds for $k = \ell$. We can show that it also holds for $k = \ell + 1$ by using the following majorizations in the entrywise nonzero pattern

$$\begin{bmatrix} \widehat{E}_{\ell+1} & 0 \\ 0 & \widehat{F}_{\ell+1} \end{bmatrix} \overset{0}{\preceq} \begin{bmatrix} \widehat{E}_\ell & 0 \\ 0 & \hat{F}_\ell \end{bmatrix}, \quad \begin{bmatrix} 0 & Y_{\ell+1} \\ X_{\ell+1} & 0 \end{bmatrix} \overset{0}{\preceq} \begin{bmatrix} 0 & Y_\ell \\ X_\ell & 0 \end{bmatrix}$$
$$\begin{bmatrix} I_m & -Y_{\ell+1} \\ -X_{\ell+1} & I_n \end{bmatrix}^{-1} \overset{0}{\preceq} \begin{bmatrix} I_m & -Y_\ell \\ -X_\ell & I_n \end{bmatrix}^{-1}.$$

By induction, (A.4) holds for all k. □

References

1. Alfa, A.S., Xue, J., Ye, Q.: Accurate computation of the smallest eigenvalue of a diagonally dominant M-matrix. Math. Comp. **71**, 217–236 (2002)
2. Alfa, A.S., Xue, J., Ye, Q.: Entrywise perturbation theory for diagonally dominant M-matrices with applications. Numer. Math. **90**(3), 401–414 (2002)

3. American National Standards Institute and Institute of Electrical and Electronic Engineers: IEEE standard for binary floating-point arithmetic. ANSI/IEEE Standard, Std 754–1985, New York (1985)
4. Bailey, D.H., Hida, Y., Li, X.S., Thompson, B.: ARPREC: an arbitrary precision computation package. Tech. rep., Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (2002). Available at http://crd-legacy.lbl.gov/~dhbailey/dhbpapers/arprec.pdf
5. Bean, N.G., O'Reilly, M.M., Taylor, P.G.: Algorithms for return probabilities for stochastic fluid flows. Stoch. Models **21**, 149–184 (2005)
6. Bini, D.A., Meini, B., Poloni, F.: Transforming algebraic Riccati equations into unilateral quadratic matrix equations. Numer. Math. **116**, 553–578 (2010)
7. Chiang, C.Y., Chu, E.K.W., Guo, C.H., Huang, T.M., Lin, W.W., Xu, S.F.: Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. SIAM J. Matrix Anal. Appl. **31**(2), 227–247 (2009)
8. Demmel, J.: Applied Numerical Linear Algebra. SIAM, Philadelphia, PA (1997)
9. Goldberg, D.: What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv. **23**(1), 5–47 (1991)
10. Guan, J., Lu, L., Li, R.-C., Shao, R.: Self-corrective algorithms for generalized diagonally dominant matrices. J. Comput. Appl. Math. **302**, 285–300 (2016)
11. Guo, C., Higham, N.: Iterative solution of a nonsymmetric algebraic Riccati equation. SIAM J. Matrix Anal. Appl. **29**, 396–412 (2007)
12. Guo, C.H.: Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for M-matrices. SIAM J. Matrix Anal. Appl. **23**, 225–242 (2001)
13. Guo, C.H., Iannazzo, B., Meini, B.: On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation. SIAM J. Matrix Anal. Appl. **29**(4), 1083–1100 (2007)
14. Guo, C.H., Laub, A.J.: On the iterative solution of a class of nonsymmetric algebraic Riccati equations. SIAM J. Matrix Anal. Appl. **22**, 376–391 (2000)
15. Guo, X., Lin, W., Xu, S.: A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation. Numer. Math. **103**, 393–412 (2006)
16. Huang, T.M., Huang, W.Q., Li, R.-C., Lin, W.W.: A new two-phase structure-preserving doubling algorithm for critically singular m-matrix algebraic riccati equations. Numer. Linear Algebra Appl. (2015). To appear
17. Juang, J.: Existence of algebraic matrix Riccati equations arising in transport theory. Linear Algebra Appl. **230**, 89–100 (1995)
18. Juang, J., Lin, W.W.: Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices. SIAM J. Matrix Anal. Appl. **20**(1), 228–243 (1998)
19. Lancaster, P., Rodman, L.: Algebraic Riccati Equations. Oxford University Press, New York (1995)
20. Nguyen, G.T., Poloni, F.: Componentwise accurate fluid queue computations using doubling algorithms. Numer. Math. **130**(4), 763–792 (2015)
21. Poloni, F., Reis, T.: The SDA Method for Numerical Solution of Lur'e Equations. arXiv:1101.1231 (2011)
22. Rogers, L.: Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains. Ann. Appl. Probab. **4**, 390–413 (1994)
23. Stewart, G.W., Sun, J.G.: Matrix Perturbation Theory. Academic Press, Boston (1990)
24. Wang, W.G., Wang, W.C., Li, R.-C.: Alternating-directional doubling algorithm for M-matrix algebraic Riccati equations. SIAM J. Matrix Anal. Appl. **33**(1), 170–194 (2012)
25. Wang, W.G., Wang, W.C., Li, R.-C.: Deflating irreducible singular M-matrix algebraic Riccati equations. Numer. Algebra Control Optim. **3**, 491–518 (2013)
26. Xue, J., Xu, S., Li, R.-C.: Accurate solutions of M-matrix algebraic Riccati equations. Numer. Math. **120**(4), 671–700 (2012)
27. Xue, J., Xu, S., Li, R.-C.: Accurate solutions of M-matrix Sylvester equations. Numer. Math. **120**(4), 639–670 (2012)