

## Research Article

# Modular Analysis of Sequential Solution Methods for Almost Block Diagonal Systems of Equations

**Tarek M. A. El-Mistikawy**

*Department of Engineering Mathematics and Physics, Cairo University, Giza 12211, Egypt*

Correspondence should be addressed to Tarek M. A. El-Mistikawy; [t.mistik@gmail.com](mailto:t.mistik@gmail.com)

Received 18 May 2012; Accepted 21 August 2012

Academic Editor: Michele Benzi

Copyright © 2013 Tarek M. A. El-Mistikawy. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Almost block diagonal linear systems of equations can be exemplified by two modules. This makes it possible to construct all sequential forms of band and/or block elimination methods. It also allows easy assessment of the methods on the basis of their operation counts, storage needs, and admissibility of partial pivoting. The outcome of the analysis and implementation is to discover new methods that outperform a well-known method, a modification of which is, therefore, advocated.

## 1. Introduction

Systems of equations with almost block diagonal (ABD) matrix of coefficients are frequently encountered in numerical solutions of sets of ordinary or partial differential equations. Several such situations were described by Amodio et al. [1], who also reviewed sequential and parallel solvers to ABD systems and came to the conclusion that sequential solution methods needed little further study.

Traditionally, sequential solution methods of ABD systems performed LU decompositions of the matrix of coefficients  $\mathbb{G}$  through either band (scalar) elimination or block tridiagonal elimination. The famous COLROW algorithm [2], which is highly regarded for its performance, was incorporated in several applications [3–7]. It utilizes Lam's alternating column/row pivoting [8] and Varah's correspondingly alternating column/row scalar elimination [9]. The efficient block tridiagonal methods included Keller's Block Tridiagonal Row (BTDR) elimination method [10, Section 5, case i] and El-Mistikawy's Block Tridiagonal Column (BTDC) elimination method [11]. Both methods could apply a suitable form of Keller's mixed pivoting strategy [10], which is more expensive than Lam's.

The present paper is intended to explore other variants of the LU decomposition of  $\mathbb{G}$ . It does not follow the traditional approaches of treating the matrix of coefficients as a banded matrix or casting it into a block tridiagonal form. It, rather,

adopts a new approach, *modular analysis*, which offers a simple and unified way of expressing and assessing solution methods for ABD systems.

The matrix of coefficients  $\mathbb{G}$  (or, more specifically, its significant part containing the nonzero blocks) is disassembled into an ordered set of modules. (In fact, two different sets of modules are identified.) Each module  $\Gamma$  is an entity that has a head and a tail. By arranging the modules in such a way that the head of a module is added to the tail of the next, the significant part of  $\mathbb{G}$  can be reassembled. The module exemplifies the matrix, but is much easier to analyze.

All possible methods of LU decomposition of  $\mathbb{G}$  could be formulated as decompositions of  $\Gamma$ . This led to the discovery of two new promising methods: Block Column/Block Row (BCBR) Elimination and Block Column/Scalar Row (BCSR) Elimination.

The validity and stability of the elimination methods are of primary concern to both numerical analysts and algorithm users. Validity means that division by a zero is never encountered, whereas stability guards against round-off-error growth. To insure validity and achieve stability, pivoting is called for [12]. Full pivoting is computationally expensive requiring full two-dimensional search for the pivots. Moreover, it destroys the banded form of the matrix of coefficients. Partial pivoting strategies, though potentially less stable, are considerably less expensive. Unidirectional (row



to the following components: the stem  $\sigma_j$ , the head  $\eta_j^+ \equiv [\eta_j \vdots \theta_j]$ , and the fins  $\psi_j$  and  $\phi_j^+ \equiv [\phi_j \vdots \rho_j]$ .

Each module has a tail  $\tau_j^+ \equiv [\tau_j \vdots \lambda_j]$ . For  $\Gamma_j^{+A}$ ,  $\tau_j^{+A} = [Cm_j^\# \ Dm_j^\# \ gm_j^\#]$  which is defined through the head-tail relation

$$\eta_{j-1}^{+A} + \tau_j^{+A} = [Cm_j \ Dm_j \ gm_j]. \quad (6)$$

For  $\Gamma_j^{+D}$ ,  $\tau_j^{+D} = \begin{bmatrix} Bn_j^\# & gn_j^\# \\ Bm_j^\# & gm_{j+1}^\# \end{bmatrix}$  which is, likewise, defined through the head-tail relation

$$\eta_{j-1}^{+D} + \tau_j^{+D} = \begin{bmatrix} Bn_j & gn_j \\ Bm_j & gm_{j+1} \end{bmatrix}. \quad (7)$$

This makes it possible to construct the significant part of  $\mathbb{G}^+$  by arranging each set of modules in such a way that the tail of  $\Gamma_j^+$  adds to the head of  $\Gamma_{j-1}^+$ , for  $j = 0 \rightarrow J$ . Minor adjustments need only to be invoked at both ends of  $\mathbb{G}^+$ . Specifically, we define the truncated modules

$$\begin{aligned} \Gamma_0^{+A} &\equiv \eta_0^{+A} = 0, \\ \Gamma_J^{+A} &= \begin{bmatrix} Cm_J^\# & Dm_J^\# & gm_J^\# \\ An_J & Bn_J & gn_J \end{bmatrix}, \\ \Gamma_0^{+D} &= \begin{bmatrix} Cm_1 & Dm_1 & gm_1 \\ An_1 & Bn_1^\rightarrow & gn_1^\rightarrow \\ Am_1 & Bm_1^\rightarrow & gm_2^\rightarrow \end{bmatrix}, \\ \Gamma_{J-1}^{+D} &= \begin{bmatrix} Bn_{J-1}^\# & Cn_J & Dn_J & gn_{J-1}^\# \\ Bm_{J-1}^\# & Cm_J & Dm_J & gm_J^\# \\ An_J & Bn_J^\rightarrow & gn_J^\rightarrow \end{bmatrix}, \\ \Gamma_J^{+D} &= [Bn_J^\# \ gm_J^\#]. \end{aligned} \quad (8)$$

The head of the module  $\Gamma^+$  is yet to be defined. It is taken to be related to the other components of  $\Gamma^+$  by

$$\eta^+ = \psi\sigma^{-1}\phi^+, \quad (9)$$

in order to allow for decompositions of  $\Gamma^+$  having the form

$$\Gamma^+ = \mu\nu^+ = \begin{bmatrix} M \\ -\Psi \end{bmatrix} [N \vdots \Phi^+]. \quad (10)$$

The generic relations  $\sigma = MN$ ,  $\psi = \Psi N$ ,  $\phi^+ = M\Phi^+$ , and  $\eta^+ = \Psi\Phi^+$  then hold, leading to  $\eta^+ = \Psi\Phi^+ = (\psi N^{-1})(M^{-1}\phi^+) = \psi(N^{-1}M^{-1})\phi^+ = \psi\sigma^{-1}\phi^+$  as defined in (9).

**3.1. Elimination Methods.** All elimination methods can be expressed in terms of decompositions of the stem  $\sigma_j$ . Only those worthy methods that allow alternating column/row pivoting and elimination are presented here. Several inflections of the blocks of  $\mathbb{G}$  are involved and are defined in the appendix section. The sequence in which the blocks are manipulated for: decomposing the stem, processing the fins, and handling the head (evaluating the head and applying the head-tail relation to determine the tail of the succeeding module  $\Gamma_{j+1}$ ), is mentioned along with the equations (from the appendix section) involved. The correctness of the decompositions may be checked by carrying out the matrix multiplications, using the equalities of the appendix section, and comparing with the un-decomposed form of the module.

The following three methods can be generated from either module. They will be given in terms of the aligned module.

**3.1.1. Scalar Column/Scalar Row (SCSR) Elimination.** This is the method implemented by the COLROW algorithm. It performs scalar decomposition of the stem  $\sigma$ . The triangular matrices  $L$  and  $U$  appear explicitly. If unit diagonal, they are marked with a circumflex  $\hat{\phantom{x}}$ :

$$\Gamma^A = \begin{bmatrix} Lm & & & \\ An' & \hat{Ln} & & \\ Am' & Bm'' & & \end{bmatrix} \begin{bmatrix} \hat{U}m & Dm'' \\ Un & Cn' & Dn' \end{bmatrix}. \quad (11)$$

The following sequence of manipulations applies:

Stem:  $Lm\hat{U}m(A7)$ ,  $Dm''(A16)$ ,  $An'(A9)$ ,  $Bn''(A3b)$ ,  $\hat{L}nUn(A6)$ .

Fins:  $Am'(A8)$ ,  $Bm''(A2b)$ ,  $Bm''(A14)$ ,  $Cn'(A10)$ ,  $Dn'(A11)$ .

Head:  $Cm''(A4b)$ ,  $Dm''(A5b)$ .

**3.1.2. Block Column/Block Row (BCBR) Elimination.** The method performs block decomposition of the stem  $\sigma$ , in which the decomposed pivotal blocks  $\check{C}m^\# (\equiv Lm\hat{U}m)$  and  $\check{B}n^\# (\equiv \hat{L}nUn)$  appear:

$$\Gamma^A = \begin{bmatrix} \check{C}m^\# & & \\ An & In & \\ Am & Bm^* & \end{bmatrix} \begin{bmatrix} Im & Dm^* \\ \check{B}n^\# & Cn & Dn \end{bmatrix}. \quad (12)$$

The following sequence of manipulations applies:

Stem:  $\check{C}m^\#(A7)$ ,  $Dm''(A16)$ ,  $Dm^*(A17)$ ,  $Bn''(A3a)$ ,  $\check{B}n^\#(A6)$ .

Fins:  $Bm''(A2a)$ ,  $Bm''(A14)$ ,  $Bm^*(A15)$ .

Head:  $Cm''(A4a)$ ,  $Dm''(A5a)$ .

3.1.3. *Block Column/Scalar Row (BCSR) Elimination.* The method has the decomposition

$$\Gamma^A = \begin{bmatrix} \check{C}m^\# \\ An & \hat{L}n \\ \hline Am & Bm'' \end{bmatrix} \left[ \begin{array}{cc|cc} Im & Dm^* & & \\ & Un & Cn' & Dn' \end{array} \right]. \quad (13)$$

The following sequence of manipulations applies:

Stem:  $\check{C}m^\#(A7)$ ,  $Dm''(A16)$ ,  $Dm^*(A17)$ ,  $Bn^\#(A3a)$ ,  $\hat{L}nUn(A6)$ .

Fins:  $Bm^\#(A2a)$ ,  $Bm''(A14)$ ,  $Cn'(A10)$ ,  $Dn'(A11)$ .

Head:  $Cm^\#(A4b)$ ,  $Dm^\#(A5b)$ .

3.1.4. *Block-Tridiagonal Row (BTDR) Elimination.* This method can be generated from the aligned module only. It performs the identity decomposition  $\sigma = Ip \check{\sigma}$ , leading to the decomposition

$$\Gamma^A = \left[ \begin{array}{c|c} \hline \check{\sigma}^A & \\ \hline Am^* & Bm^* \end{array} \right] \left[ \begin{array}{cc|c} Im & & \\ & & Cn \\ \hline & & Dn \end{array} \right]. \quad (14)$$

In [10, Section 5, case (i)], scalar row elimination was used to obtain the decomposed stem  $\check{\sigma}$ . However,  $\check{\sigma}$  can, by now, be obtained by any of the nonidentity (scalar and/or block) decomposition methods given in Sections 3.1.1–3.1.3.

Using SCSR elimination, the following sequence of manipulations applies:

Stem:  $Lm\hat{U}m(A7)$ ,  $Dm''(A16)$ ,  $An'(A9)$ ,  $Bn^\#(A3b)$ ,  $\hat{L}nUn(A6)$ .

Fins:  $Am'(A8)$ ,  $Bm^\#(A2b)$ ,  $Bm''(A14)$ ,  $Bm^*(A15)$ ,  $Am''(A12)$ ,  $Am^*(A13)$ .

Head:  $Cm^\#(A4a)$ ,  $Dm^\#(A5a)$ .

3.1.5. *Block-Tridiagonal Column (BTDC) Elimination.* This method can be generated from the displaced module only. It performs the identity decomposition  $\sigma = \check{\sigma} Ip$ , leading to the decomposition

$$\Gamma^D = \left[ \begin{array}{c|c} \hline \check{\sigma}^D & \\ \hline In & An \\ & Am \end{array} \right] \left[ \begin{array}{c|c} Ip & \\ \hline & Dn^* \\ & Dm^* \end{array} \right]. \quad (15)$$

In [11],  $\check{\sigma}$  was obtained by scalar column elimination. As with BTDR elimination,  $\check{\sigma}$  can, by now, be obtained from any of the nonidentity decompositions given in Sections 3.1.1–3.1.3.

Using SCSR elimination, the following sequence of manipulations applies:

Stem:  $\hat{L}nUn(A6)$ ,  $Cn'(A10)$ ,  $Bm''(A14)$ ,  $Cm^\#(A4b)$ ,  $Lm\hat{U}m(A7)$ .

Fins:  $Dn'(A11)$ ,  $Dm^\#(A5b)$ ,  $Dm''(A16)$ ,  $Dm^*(A17)$ ,  $Dn''(A18)$ ,  $Dn^*(A19)$ .

Head:  $Bn^\#(A3a)$ ,  $Bm^\#(A2a)$ .

3.2. *Solution Procedure.* The procedure for solving the matrix equation  $\mathbb{G}^+ \mathbf{z} = \mathbf{g}$ , which can be described in terms of manipulation of the augmented matrix  $\mathbb{G}^+ \equiv [\mathbb{G} : \mathbf{g}]$ , can, similarly, be described in terms of manipulation of the augmented module  $\Gamma_j^+ \equiv [\Gamma_j : \gamma_j]$ . The manipulation of  $\mathbb{G}^+$  applies a forward sweep which corresponds to the decomposition  $\mathbb{G}^+ = \mathbb{L}\mathbb{U}^+ \equiv \mathbb{L}[\mathbb{U} : Z]$  that is followed by a backward sweep which corresponds to the decomposition  $\mathbb{U}^+ = \mathbb{U}[\mathbb{I} : \mathbf{z}]$ . Similarly, the manipulation of  $\Gamma_j^+$  applies a forward sweep involving two steps. The first step performs the decomposition  $\Gamma_j^+ = \mu_j \nu_j^+$ . The second step evaluates the head  $\eta_j^+ = \Psi_j \Phi_j^+$  then applies the head-tail relation to determine the tail of  $\Gamma_{j+1}^+$ . In a backward sweep, two steps are applied to  $\nu_j^+ \equiv [N_j \ \Phi_j : Z_j]$  leading to the solution module  $z_j$  ( $z_j^A = [zm_j^t \ zn_j^t]^t$ ,  $z_j^D = [zn_j^t \ zm_{j+1}^t]^t$ ). With  $z_{j+1}$  known, the first step uses  $\mathfrak{z}_{j+1}^A (\mathfrak{z}_{j+1}^A = z_{j+1}^A, \mathfrak{z}_{j+1}^D = zn_{j+1})$  in the back substitution relation  $Z_j - \Phi_j \mathfrak{z}_{j+1} = \zeta_j$  to contract  $\nu_j^+$  to  $N_j^+ \equiv [N_j : \zeta_j]$ . The second step solves  $N_j z_j = \zeta_j$  for  $z_j$  which is equivalent to the decomposition  $N_j^+ = N_j [Ip : z_j]$ .

3.3. *Operation Counts and Storage Needs.* The modules introduced previously allow easy evaluation of the elimination methods. The operation counts are measured by the number of multiplications (mul) with the understanding that the number of additions is comparable. The storage needs are measured by the number of locations (loc) required to store arrays calculated in the forward sweep for use in the backward sweep, provided that the elements of  $\mathbb{G}^+$  are not stored but are generated when needed, as is usually done in the numerical solution of a set of ODEs, for example.

Per module (i.e., per grid point), each method requires, for the manipulation of  $\mathbb{G}^+$ , as many operations as it requires to manipulate  $\Gamma^+$ . All methods require  $(p^3 - p)/3$  (mul) for decomposing the stem,  $p m n$  (mul) for evaluating the head, and  $2p^2 r$  (mul) to handle the right module  $\gamma$ . The methods differ only in the operation counts for processing the fins  $\psi$  and  $\phi$ , with BCBR elimination requiring the least count  $p m n$  (mul).

Per module, each method requires as many storage locations as it requires to store  $\nu^+ \equiv [\nu : Z]$ . All methods require  $p r$  (loc) to store  $Z$ . They differ in the number of locations needed for storing the  $\nu$ 's, with BTDC elimination requiring the least number  $p n$  (loc). Note that, in SCSR and BCSR eliminations, square blocks need to be reserved for storing the triangular blocks  $\hat{U}m$  and/or  $Un$ .

Table 1 contains these information, allowing for clear comparison among the methods. For example, when  $p \gg 1$ , BCBR elimination achieves savings in operations that are of leading order significance  $\sim p^3/8$  and  $\sim p^3/2$ , respectively, in the two distinguished limits  $m \sim n \sim p/2$  and  $(m, n) \sim (p, 1)$ , as compared to SCSR elimination.

3.4. *Pivoting Strategies.* Lam's alternating pivoting [8] applies column pivoting to  $\tau^A = [Cm^\# \ Dm^\#]$  in order to form

TABLE 1: Operation counts and storage needs.

Method	Operation counts (mul)	Storage needs (loc)
	$2p^2r + (p^3 - p)/3 + 2pmn +$	$pr + pn +$
BCBR	0	$pn$
SCSR	$(m^3 + n^3 - m^2 - n^2)/2$	$pn + m^2$
BCSR	$(n^3 - n^2)/2$	$pn$
BTDC	$pn^2$	0

TABLE 2: Execution times in seconds for the first system.

$m/n^a$	SCSR	BTDC	BCBR	BCSR
10/1	109	96	93	91
9/2	113	104	106	99
8/3	118	116	115	110
7/4	120	126	122	117
6/5	123	133	133	125

<sup>a</sup> $m + n = 11$ .

TABLE 3: Execution times in seconds for the second system.

$m/n^a$	SCSR	BTDC	BCBR	BCSR
20/1	67.9	52.0	51.8	51.4
18/3	70.6	59.9	60.0	55.0
16/5	72.0	68.3	73.8	66.3
14/7	73.0	74.2	80.8	73.8
12/9	73.1	78.3	81.4	74.8
11/10	73.6	81.7	82.3	75.9

<sup>a</sup> $m + n = 21$ .

and decompose a nonsingular pivotal block  $Cm^\#$  and applies row pivoting to  $\tau^D = [Bn^{\#t} Bm^{\#t}]^t$  in order to form and decompose a nonsingular pivotal block  $Bn^\#$ . These are valid processes since  $\tau^A$  is of rank  $m$  and  $\tau^D$  is of rank  $n$ , as can be shown following the reasoning of Keller [10, Section 5, Theorem].

To enhance stability further we introduce the Local pivoting strategy which applies full pivoting (maximum pivot strategy) to the segments  $\tau^A$  and  $\tau^D$ . Note that Keller's mixed pivoting [10], if interpreted as applying full pivoting to  $\tau^A$  and row pivoting to  $\tau^D$ , is midway between Lam's and local pivoting.

Lam's, Keller's, and local pivoting apply to all elimination methods of Section 3.1. Moreover, in any given problem, each pivoting strategy would produce the same sequence of pivots, regardless of the elimination method in use.

## 4. Implementation

The COLROW algorithm, which is based on SCSR elimination, is modified to perform BTDC, BCBR, or BCSR elimination, instead. As an illustration, the four methods are applied to three systems of equations having the augmented matrix given in (1), with  $J = 11$  and  $r = 1$ . The solution procedures are repeated  $k$  times, so that reasonable execution times can be recorded and compared. All calculations are

carried out in double precision via Compaq Visual Fortran (version 6.6) that is optimized for speed, on a Pentium 4 CPU rated at 2 GHz with 750 MB of RAM.

The first system has  $p = 11$  and  $k = 10^6$ . Different combinations of  $m/n = 10/1, 9/2, 8/3, 7/4$ , and  $6/5$  are considered. The execution times, without pivoting, are given in Table 2. All entries include  $\approx 22$  seconds that are required to read, write, and run empty Fortran-Do-Loops. Pivoting requires additional  $\approx 20$  seconds in all methods.

The second (third) system has  $p = 21$  (51) and  $k = 10^5$  ( $10^4$ ). The execution times, with Lam's pivoting, are given in Table 3 (4), for different combinations of  $m/n$ .

Although the modified COLROW algorithms may not produce the best performances of BTDC, BCBR, and BCSR eliminations, Tables 2, 3, and 4 clearly indicate that they, in some cases (when  $m \gg n$ ), outperform the COLROW algorithm that is designed to give SCSR elimination its best performance.

## 5. Conclusion

Using the novel approach of modular analysis, we have analyzed the sequential solution methods for almost block diagonal systems of equations. Two modules have been identified and have made it possible to express and assess all possible band and block elimination methods. On the

TABLE 4: Execution times in seconds for the third system.

$m/n^a$	SCSR	BTDC	BCBR	BCSR
50/1	60.51	38.05	39.19	38.04
46/5	62.71	48.30	57.55	49.15
41/10	65.56	58.97	76.25	61.57
36/15	68.12	65.86	87.53	68.35
31/20	69.09	76.38	110.61	85.24
26/25	69.86	83.95	115.95	94.00

<sup>a</sup> $m + n = 51$ .

basis of the operation counts, storage needs, and admissibility of partial pivoting, we have determined four distinguished methods: Block Column/Block Row (BCBR) Elimination (having the least operation count), Block Tridiagonal Column (BTDC) Elimination (having the least storage need), Block Column/Scalar Row (BCSR) elimination, and Scalar Column/Scalar Row (SCSR) elimination (implemented in the well-known COLROW algorithm). Application of these methods within the COLROW algorithm shows that they outperform SCSR elimination, in cases of large top-block/bottom-block row ratio. In such cases, BCSR elimination is advocated as an effective modification to the COLROW algorithm.

## Appendix

In Section 3, two modules,  $\Gamma^A$  and  $\Gamma^D$ , of the matrix of coefficients  $\mathbb{G}$  are introduced and decomposed to generate the elimination methods. The process involves inflections of the blocks of  $\mathbb{G}$ , which proceed for a block  $E$ , say, according to the following scheme:

$$\begin{array}{ccc} E & \longrightarrow & E^\# \xrightarrow{\text{if pivotal}} \check{E}^\# \\ \downarrow & & \downarrow \\ E' & \longrightarrow & E'' \longrightarrow E^* \end{array} \quad (\text{A.1})$$

The following equalities are to be used to determine the blocks with underscored leading character. The number of multiplications involved is given between braces:

$E^\#$ -Blocks

$$Bm - \underline{Bm}^\# \stackrel{a}{=} AmDm^* \stackrel{b}{=} Am'Dm'' \quad \{m^2n\}, \quad (\text{A.2})$$

$$Bn - \underline{Bn}^\# \stackrel{a}{=} AnDm^* \stackrel{b}{=} An'Dm'' \quad \{mn^2\}, \quad (\text{A.3})$$

$$Cm - \underline{Cm}^\# \stackrel{a}{=} Bm^*Cn \stackrel{b}{=} Bm''Cn' \quad \{m^2n\}, \quad (\text{A.4})$$

$$Dm - \underline{Dm}^\# \stackrel{a}{=} Bm^*Dn \stackrel{b}{=} Bm''Dn' \quad \{mn^2\} \quad (\text{A.5})$$

$\check{E}^\#$ -Blocks (decomposed pivotal blocks)

$$Bn^\# = \check{Bn}^\# (\equiv \underline{LnUn}) \quad \left\{ \frac{n^3 - n}{3} \right\} \quad (\text{A.6})$$

$$Cm^\# = \check{Cm}^\# (\equiv \underline{LmUm}) \quad \left\{ \frac{m^3 - m}{3} \right\}, \quad (\text{A.7})$$

$E'$ -Blocks

$$\underline{Am}'\widehat{Um} = Am \quad \left\{ \frac{m^2 - m}{2}m \right\}, \quad (\text{A.8})$$

$$\underline{An}'\widehat{Um} = An \quad \left\{ \frac{m^2 - m}{2}n \right\}, \quad (\text{A.9})$$

$$\widehat{Ln}\underline{Cn}' = Cn \quad \left\{ \frac{n^2 - n}{2}m \right\} \quad (\text{A.10})$$

$$\widehat{Ln}\underline{Dn}' = Dn \quad \left\{ \frac{n^2 - n}{2}n \right\}, \quad (\text{A.11})$$

$E''$  and  $E^*$ -Blocks

$$\underline{Am}'' = Am' - Bm^*An' \quad \{m^2n\}, \quad (\text{A.12})$$

$$\underline{Am}^*Lm = Am'' \quad \left\{ \frac{m^2 + m}{2}m \right\} \quad (\text{A.13})$$

$$\underline{Bm}''Un = Bm^\# \quad \left\{ \frac{n^2 + n}{2}m \right\}, \quad (\text{A.14})$$

$$\underline{Bm}^*\widehat{Ln} = Bm'' \quad \left\{ \frac{n^2 - n}{2}m \right\}, \quad (\text{A.15})$$

$$Lm\underline{Dm}'' = Dm^\# \quad \left\{ \frac{m^2 + m}{2}n \right\}, \quad (\text{A.16})$$

$$\widehat{Um}\underline{Dm}^* = Dm'' \quad \left\{ \frac{m^2 - m}{2}n \right\} \quad (\text{A.17})$$

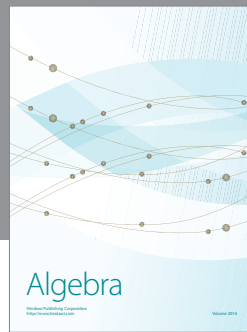
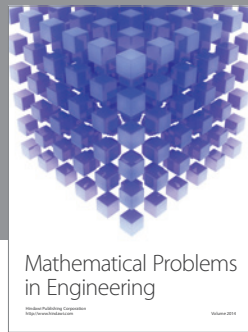
$$\underline{Dn}'' = Dn' - Cn'Dm^* \quad \{mn^2\}, \quad (\text{A.18})$$

$$Un\underline{Dn}^* = Dn'' \quad \left\{ \frac{n^2 - n}{2}n \right\}. \quad (\text{A.19})$$

## References

- [1] P. Amodio, J. R. Cash, G. Roussos et al., "Almost block diagonal linear systems: sequential and parallel solution techniques, and applications," *Numerical Linear Algebra with Applications*, vol. 7, no. 5, pp. 275–317, 2000.
- [2] J. C. Díaz, G. Fairweather, and P. Keast, "FORTRAN packages for solving certain almost block diagonal linear systems by

- modified alternate row and column elimination,” *ACM Transactions on Mathematical Software*, vol. 9, no. 3, pp. 358–375, 1983.
- [3] J. R. Cash and M. H. Wright, “A deferred correction method for nonlinear two-point boundary value problems: implementation and numerical evaluation,” *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 4, pp. 971–989, 1991.
  - [4] J. R. Cash, G. Moore, and R. W. Wright, “An automatic continuation strategy for the solution of singularly perturbed linear two-point boundary value problems,” *Journal of Computational Physics*, vol. 122, no. 2, pp. 266–279, 1995.
  - [5] W. H. Enright and P. H. Muir, “Runge-Kutta software with defect control for boundary value ODEs,” *SIAM Journal on Scientific Computing*, vol. 17, no. 2, pp. 479–497, 1996.
  - [6] P. Keast and P. H. Muir, “Algorithm 688: EPDCOL: a more efficient PDECOL code,” *ACM Transactions on Mathematical Software*, vol. 17, no. 2, pp. 153–166, 1991.
  - [7] R. W. Wright, J. R. Cash, and G. Moore, “Mesh selection for stiff two-point boundary value problems,” *Numerical Algorithms*, vol. 7, no. 2–4, pp. 205–224, 1994.
  - [8] D. C. Lam, *Implementation of the box scheme and model analysis of diffusion-convection equations. [Ph.D. thesis]*, University of Waterloo: Waterloo, Ontario, Canada, 1974.
  - [9] J. M. Varah, “Alternate row and column elimination for solving certain linear systems,” *SIAM Journal on Numerical Analysis*, vol. 13, no. 1, pp. 71–75, 1976.
  - [10] H. B. Keller, “Accurate difference methods for nonlinear two-point boundary value problems,” *SIAM Journal on Numerical Analysis*, vol. 11, pp. 305–320, 1974.
  - [11] T. M. A. El-Mistikawy, “Solution of Keller’s box equations for direct and inverse boundary-layer problems,” *AIAA journal*, vol. 32, no. 7, pp. 1538–1541, 1994.
  - [12] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, UK, 1986.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

