2007

TR-2007021: Row-Column Pivoting in Gaussian Elimination

Victor Y. Pan

Guoliang Qian

Ai-Long Zheng

Severin Ngnosse

Row–Column Pivoting in Gaussian Elimination *

Victor Y. Pan[1], Guoliang Qian[2,a],
Ai-Long Zheng[3], Severin Ngnosse[2,b]

[1] victor.pan@lehman.cuny.edu
Department of Mathematics and Computer Science
Lehman College, City University of New York
Bronx, NY 10468, USA
http://comet.lehman.cuny.edu/vpan/

[2] Ph.D. Program in Computer Science
The City University of New York
New York, NY 10036 USA

[3] azheng 1999@yahoo.com
Ph.D. Program in Mathematics
The City University of New York
New York, NY 10036 USA

[a]guoliangqian@yahoo.com
[b]sngnosse@msn.com

November 8, 2007

Abstract

We introduce a new policy of pivoting in Gaussian elimination and test its power versus partial and complete pivoting. According to our extensive tests, the policy is as stable numerically as complete pivoting and uses quadratic number of comparisons like partial pivoting. In particular this behavior was observed for the known classes of input matrices for which partial pivoting fails.

Key words: Gaussian elimination, pivoting

1 Introduction

Hereafter we write GEPP, GECP, and GERCP to denote Gaussian elimination with partial, complete, and row-column pivoting. GEPP and GPPP are classical algorithms [1], [2], [3], whereas GERCP is our novelty. Each of the three algorithms uses $(2/3)n^3 + O(n^2)$ flops to yield triangular factorization of an $n \times n$ matrix, but they differ in the number of comparisons involved, and GEPP has slightly weaker numerical stability. Namely, GECP guarantees numerical stability [4], whereas GEPP is statistically stable for most of the input instances in computational practice but fails for some rare but important classes of inputs [5], [6], [7]. Nevertheless GEPP is omnipresent in modern numerical matrix computations, whereas GECP is rarely used. The reason is simple: GEPP involves $(1/2)n^2 + O(n)$ comparisons versus $(1/3)n^3 + O(n^2)$ in GECP, that is the computational cost of pivoting is negligible versus arithmetic cost for GEPP but is substantial for GECP.

Our new algorithm GERCP combines the advantages of both GECP and GEPP. According to our extensive tests, GERCP is as stable numerically as GECP and uses about $2n^2$ comparisons (see Remark 4.1). Like GEPP and GECP, our algorithm can include initial scaling, which is the customary additional heuristic protection against instability and requires from about n^2 to about $2n^2$ comparisons and as many flops [1, Section 3.5.2], [2, Section 3.4.4], [3, Section 9.7], so that its overall computational cost is still strongly dominated by the elimination flops.

In the next section we specify a class of input matrices for which already the rounding errors at the first step of GEPP (but neither GECP nor

*Supported by PSC CUNY Award 69330–0038

GERCP) completely corrupt the output. In Section 3 we specify GERCP. In Section 4 we present the test results. The tests were designed by the first author of this paper and were implemented and performed by Dr. Xinmao Wang at the USTC, Hefei, Anhui, China (see Tables 1–4 in Section 4) and by the last three authors of the present paper (see Remark 4.1). Otherwise the paper is due to its first author.

2 A Hard Input Class for GEPP

Already the first step of Gaussian elimination tends to magnify the input errors wherever the pivot entry is absolutely smaller than some other entries in the same row and column. For example, represent an input matrix M as follows,

$$M = \begin{pmatrix} 1 & \mathbf{v}^T \\ \mathbf{u} & B \end{pmatrix} = (m_{ij})_{i,j=0}^{n-1}, \quad B = (m_{ij})_{i,j=1}^{n-1}, \tag{2.1}$$

let ϵ denote the machine epsilon (also called unit roundoff), and suppose that

$$\mathbf{u} = U\mathbf{e}, \quad \mathbf{v} = V\mathbf{e}, \quad \mathbf{e} = (1, 1, \ldots, 1)^T, \quad |m_{ij}| \leq 1 \text{ for } i, j > 0, \tag{2.2}$$

$$U < 2/\epsilon, \quad V = 1.$$

Then the first elimination step, performed error-free, produces an $(n-1) \times (n-1)$ matrix $B_U = B + U\mathbf{e}\mathbf{e}^T$, which turns into a rank-one matrix $\mathrm{fl}(U)\mathbf{e}\mathbf{e}^T$ in the result of rounding. Here and hereafter $\mathrm{fl}(A)$ denotes the matrix whose entries are the floating-point representations of the respective entries of a real matrix A.

Partial pivoting fixes the latter problem for this matrix but does not help against exactly the same problem where the input matrix M satisfies equations (2.1) and (2.2) and where

$$U = 1, \quad V > 2/\epsilon. \tag{2.3}$$

In this case the first elimination step, performed error-free, would produce the $(n-1) \times (n-1)$ matrix $B_U = B + V\mathbf{e}\mathbf{e}^T$. Rounding would turn it into the rank-one matrix $\mathrm{fl}(V)\mathbf{e}\mathbf{e}^T$.

We refer the reader to [5] and [6] (cf. also [7]) on some narrow but important classes of linear systems of equations coming from computational practice on which GEPP fails to produce correct output.

3 Gaussian Elimination with Row–Column Pivoting (GERCP)

Algorithm 3.1. Row–Column Pivoting.

INPUT: a nonsingular $n \times n$ matrix $M = (m_{ij})_{i,j=0}^{n-1}$.

OUTPUT: A pair of integers (g, h) such that

$$|m_{gh}| = \max\{\max\{|m_{gj}|, j = 0, \ldots, n-1\}, \max\{|m_{ih}|, i = 0, \ldots, n-1\}\}.$$

INITIALIZATION: $h \leftarrow 0$, $G \leftarrow \{0, 1, \ldots, n-1\}$, and $H \leftarrow \{0, 1, \ldots, n-1\}$.

COMPUTATIONS:

1. Compute an integer g such that $|m_{gh}| = \max_{i \in G}\{|m_{ih}|\}$.
2. Compute an integer k such that $|m_{gk}| = \max_{j \in H}\{|m_{gj}|\}$.

If $|m_{gh}| = |m_{gk}|$, output the pair (g, h) and stop.

3. Otherwise $h \leftarrow k$, and go to Stage 1.

One can apply a single step of Gaussian elimination with the pivot m_{gh} output by Algorithm 3.1 and then recursively alternate applications of this algorithm and the Gaussian elimination steps until triangular factorization of the matrix M is computed. This defines our GERCP.

Remark 3.2. Algorithm 3.1 includes redundant comparisons because at Stage 3 we keep the sets G and H unchanged. We can modify Stage 3 as follows: 3. Otherwise $G \leftarrow G - \{g\}$, $H \leftarrow H - \{h\}$, $h \leftarrow k$, and go to Stage 1. At the cost of such a set manipulation, we would save some comparisons at the next steps. We refer to GERCP based on the latter version of Row-Column Pivoting as GERCP1.

4 Experimental Results

Tables 1–4 show the results of testing GERCP (based on Algorithm 3.1) by Dr. Xinmao Wang at the Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China. He implemented the GERCP algorithm in C++ under the 64-bit Fedore Core 7 Linux with AMD

Athlon64 3200+ uniprocessor and 1 GB memory. In his implementation he used n comparisons for computing the maximum of n numbers. He tested the algorithm for $n \times n$ matrices M of the following seven classes.

1. Matrices with random integer entries in the range $(-10^l, 10^l)$.

2. Matrices $M = PLU$ for $n \times n$ permutation matrices P that define n interchanges of random pairs of rows and for lower unit triangular matrices L and U^T with random integer entries in the range $(-10^b, 10^b)$.

3. Matrices $M = S\Sigma T$ for random orthogonal matrices S and T (computed as the Q-factors in the QR factorization of matrices with random integer entries in the range $(-10^c, 10^c)$) and for the diagonal matrix $\Sigma = \mathrm{diag}(\sigma_i)_{i=1}^n$ where $\sigma_1 = \sigma_2 = \cdots = \sigma_{n-\rho} = 1$ and $\sigma_{n-\rho+1} = \sigma_n = 10^{-q}$.

4. Matrices M satisfying equations (2.1)–(2.3) where B denotes an $(n-1) \times (n-1)$ matrix from matrix class 1 above.

5. Matrices $M = \begin{pmatrix} I & O & & \dots & I \\ -M_1 & I & O & \dots & O \\ & -M_1 & I & & \vdots \\ & & \ddots & \ddots & O \\ & & & -M_1 & I \end{pmatrix}$ from [5, page 232], where $M_1 = \exp\begin{pmatrix} -0.05 & 0.3 \\ 0.3 & -0.05 \end{pmatrix} \approx \begin{pmatrix} 0.994357 & 0.289669 \\ 0.289669 & 0.994357 \end{pmatrix}$.

6. Matrices $M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & -1/C \\ -\frac{kh}{2} & 1-\frac{kh}{2} & 0 & \cdots & 0 & -1/C \\ -\frac{kh}{2} & -kh & 1-\frac{kh}{2} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & -1/C \\ -\frac{kh}{2} & -kh & \cdots & -kh & 1-\frac{kh}{2} & -1/C \\ -\frac{kh}{2} & -kh & \cdots & -kh & -kh & 1-1/C-\frac{kh}{2} \end{pmatrix}$ from [6, page 1360], where $kh = \frac{2}{3}, C = 6$.

7. Matrices $M = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & \vdots \\ -1 & -1 & \ddots & 0 & 1 \\ \vdots & \vdots & \ddots & 1 & 1 \\ -1 & -1 & \cdots & -1 & 1 \end{pmatrix}$ from [7, page 156].

$n = 128$	minimal	maximal	average
Class 1	31371	37287	34147
Class 2	35150	40904	38168
Class 3, $\rho = 1$	30189	36097	32995
Class 3, $\rho = 2$	30597	36561	32960
Class 3, $\rho = 3$	29938	35761	32967
Class 4	31342	36333	33648
Class 5	24318		
Class 6	32258		
Class 7	32764		

$n = 256$	minimal	maximal	average
Class 1	131692	146780	139419
Class 2	147123	161971	153559
Class 3, $\rho = 1$	127911	143706	136361
Class 3, $\rho = 2$	129228	144226	136427
Class 3, $\rho = 3$	129945	145882	136508
Class 4	131533	146014	138392
Class 5	97790		
Class 6	130050		
Class 7	131068		

Table 1: numbers of comparisons in GERCP.

$n = 128$	GEPP	GECP	GERCP
Class 1	13.8 ± 2.5	6.4 ± 0.4	8.4 ± 0.8
Class 2	2.5 ± 0.5	1.5 ± 0.2	1.8 ± 0.2
Class 3, $\rho = 1$	17.4 ± 4.0	8.7 ± 1.0	11.6 ± 1.8
Class 3, $\rho = 2$	15.6 ± 3.6	7.7 ± 0.8	10.2 ± 1.4
Class 3, $\rho = 3$	14.3 ± 3.5	7.0 ± 0.7	9.3 ± 1.3
Class 4	FAIL	1	1
Class 5	3.4e6	2	2
Class 6	6.6e36	1.33	1.33
Class 7	1.7e38	2	2

$n = 256$	GEPP	GECP	GERCP
Class 1	21.8 ± 3.8	9.5 ± 0.6	12.8 ± 1.3
Class 2	3.4 ± 0.6	1.9 ± 0.2	2.4 ± 0.3
Class 3, $\rho = 1$	32.2 ± 7.4	15.5 ± 1.7	20.6 ± 2.9
Class 3, $\rho = 2$	29.2 ± 6.7	13.8 ± 1.4	18.6 ± 2.9
Class 3, $\rho = 3$	27.0 ± 6.1	12.5 ± 1.3	16.7 ± 2.3
Class 4	FAIL	1	1
Class 5	3.1e13	2	2
Class 6	8.6e74	1.33	1.33
Class 7	5.8e76	2	2

Table 2: growth factor in GEPP/GECP/GERCP.

$n = 128$	GEPP	GECP	GERCP
Class 1	6.8e-13 ± 3.4e-12	5.2e-13 ± 2.8e-12	4.8e-13 ± 2.2e-12
Class 2	1.7e7 ± 2.6e8	8.7e5 ± 4.6e6	6.6e5 ± 3.7e6
Class 3, $\rho = 1$	1.1e-5 ± 8.4e-6	7.4e-6 ± 5.7e-6	8.7e-6 ± 6.7e-6
Class 3, $\rho = 2$	1.7e-5 ± 8.8e-6	1.2e-5 ± 6.1e-6	1.3e-5 ± 7.0e-6
Class 3, $\rho = 3$	2.1e-5 ± 9.2e-6	1.5e-5 ± 6.2e-6	1.7e-5 ± 7.5e-6
Class 4	FAIL	5.7e-13 ± 6.3e-12	5.7e-13 ± 3.5e-12
Class 5	1.0e-9	2.7e-15	2.7e-15
Class 6	3.1e3	2.7e-15	2.7e-15
Class 7	6.5	0.0	0.0

$n = 256$	GEPP	GECP	GERCP
Class 1	3.8e-12 ± 3.7e-11	2.8e-12 ± 4.0e-11	2.6e-12 ± 2.0e-11
Class 2	3.9e7 ± 5.0e8	1.1e6 ± 4.1e6	2.2e6 ± 1.3e7
Class 3, $\rho = 1$	2.0e-5 ± 1.5e-5	1.3e-5 ± 9.3e-6	1.5e-5 ± 1.1e-5
Class 3, $\rho = 2$	3.1e-5 ± 1.6e-5	2.0e-5 ± 1.1e-5	2.4e-5 ± 1.2e-5
Class 3, $\rho = 3$	3.9e-5 ± 1.7e-5	2.5e-5 ± 1.1e-5	2.9e-5 ± 1.2e-5
Class 4	FAIL	3.6e-12 ± 4.0e-11	3.6e-12 ± 2.5e-11
Class 5	1.4e-2	3.7e-15	3.7e-15
Class 6	7.2e57	3.6e-14	3.6e-14
Class 7	11.3	0.0	0.0

Table 3: norms of the error vectors in GEPP/GECP/GERCP.

$n = 128$	GEPP	GECP	GERCP
Class 1	1.6e-9 ± 3.0e-10	1.1e-9 ± 1.7e-10	1.2e-9 ± 2.1e-10
Class 2	2.2e-4 ± 1.6e-3	1.2e-4 ± 4.7e-4	1.1e-4 ± 6.3e-4
Class 3, $\rho = 1$	3.1e-14 ± 5.1e-15	2.0e-14 ± 2.9e-15	2.3e-14 ± 3.6e-15
Class 3, $\rho = 2$	3.0e-14 ± 5.0e-15	1.9e-14 ± 2.8e-15	2.3e-14 ± 3.6e-15
Class 3, $\rho = 3$	3.0e-14 ± 5.3e-15	1.9e-14 ± 2.8e-15	2.3e-14 ± 3.5e-15
Class 4	FAIL	3.3e2 ± 3.3e2	3.5e2 ± 3.3e2
Class 5	1.1e-9	1.9e-15	1.9e-15
Class 6	2.9e3	1.7e-14	1.7e-14
Class 7	14.5	0.0	0.0

$n = 256$	GEPP	GECP	GERCP
Class 1	7.1e-9 ± 1.1e-9	4.4e-9 ± 5.8e-10	5.2e-9 ± 7.2e-10
Class 2	2.1e-3 ± 3.7e-2	6.2e-4 ± 2.1e-3	1.5e-3 ± 1.6e-2
Class 3, $\rho = 1$	9.8e-14 ± 1.5e-14	5.7e-14 ± 6.8e-15	7.4e-14 ± 9.3e-15
Class 3, $\rho = 2$	9.7e-14 ± 1.4e-14	5.7e-14 ± 7.0e-15	7.1e-14 ± 9.2e-15
Class 3, $\rho = 3$	3.9e-5 ± 1.7e-5	5.7e-14 ± 6.9e-15	7.0e-14 ± 9.1e-15
Class 4	FAIL	6.7e2 ± 6.5e2	6.6e2 ± 6.3e2
Class 5	9.0e-3	2.6e-15	2.6e-15
Class 6	2.1e58	1.0e-13	1.0e-13
Class 7	41.1	0.0	0.0

Table 4: norms of the residual vectors in GEPP/GECP/GERCP.

For each matrix of classes 1–4 the tests were performed for $m = 1000$ input instances M for each of the two values $n = 128$ and $n = 256$, for $b = c = l = 4$, and for $q = 10$. For class 3 the tests were performed for each of the three values $\rho = 1, 2, 3$. Besides the results of these tests, Tables 1–4 also cover the test results for matrices M of classes 5–7 (from the papers [5], [6], and [7], respectively), for which GEPP produced corrupted outputs.

To every matrix GEPP, GECP, and GERCP were applied. As was expected, for matrix classes 1–3 numerical performance of GEPP, GECP, and GERCP was similar but for classes 4–7 GEPP either failed or lost many more correct input bits versus GECP and GERCP.

Table 1 shows the maximum, minimum and average numbers of comparisons used in GERCP for every input class of matrices.

Table 2 shows the average growth factor

$$\phi = \max{}_{i,j,k=0}^{n-1} \ |m_{ij}^{(k)}| / \max{}_{i,j=0}^{n-1} \ |m_{ij}|$$

(as well as its standard deviation from the average) where $M^{(k)} = (m_{i,j}(k))_{i,j=k}^{n-1}$ denotes the matrix computed in k steps of Gaussian elimination with the selected pivoting policy and $M = M^{(0)} = (m_{ij})_{i,j=0}^{n-1}$ denotes the input matrix.

Tables 3 and 4 show the average norms of the error and residual vectors, respectively, as well as the standard deviations from the average, where the linear systems $M\mathbf{y} = \mathbf{f}$ were solved by applying GECP, GEPP, and GERCP. The vectors $\mathbf{f}$ were defined according to the following rule: first generate vectors $\mathbf{y}$ with random components from the sets $\{-1, 0, 1\}$ or $\{-1, 1\}$, then save these vectors for computing the errors vectors, and finally compute the vectors $\mathbf{f} = M\mathbf{y}$.

Remark 4.1. Table 1 shows the results of testing GERCP where n comparisons were used for computing the maximum of n numbers. Extensive additional tests with random matrices (of class 1) for $n = 2^h$ and for h ranging from 5 to 10 were performed in the Graduate Center of the City University of New York. In these tests the modification GERCP1 was run (where the sets G and H were modified at Stage 3 of Algorithm 3.1 as we specified in Remark 3.2). Furthermore, the tests used $k - 1$ comparisons for computing the maximum of k numbers. The observed numbers of comparisons slightly decreased versus Table 1 and always stayed below $2n^2$.

Acknowledgement

We are happy to acknowledge valuable experimental support of our work by Dr. Xinmao Wang.

References

[1] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[2] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, Philadelphia, 1998.

[3] N. J. Higham, *Accuracy and Stability in Numerical Analysis*, SIAM, Philadelphia, 1996 (first edition), 2002 (second edition).

[4] J. H. Wilkinson, Error Analysis of Direct Methods of Matrix Inversion, *J. of ACM*, **8**, 281–330, 1961.

[5] S. J. Wright, A Collection of Problem for Which Gaussian Elimination with Partial Pivoting Is Unstable, *SIAM J. on Sci. Stat. Computing*, **14**, **1**, 231–238, 1993.

[6] L. V. Foster, Gaussian Elimination with Partial Pivoting Can Fail in Practice, *SIAM J. on Matrix Analisys and Applications*, **15**, **4**, 1354–1362, 1994.

[7] N. J. Higham and D. J. Higham, Large Growth Factors in Gaussian Elimination with Pivoting, *SIAM J. on Matrix Analisys and Applications*, **10**, **2**, 155–164, 1989.