Journal of Computational and Applied Mathematics 123 (2000) 353–369

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

The Rook's pivoting strategy

George Poole *, Larry Neal [1]

Mathematics Department, East Tennessee State University, Johnson City, TN 37614-0663, USA

Received 24 February 1999

Abstract

Based on the geometric analysis of Gaussian elimination (GE) found in Neal and Poole (Linear Algebra Appl. 173 (1992) 239–264) and Poole and Neal (Linear Algebra Appl. 149 (1991) 249–272; 162–164 (1992) 309–324), a new pivoting strategy, *Rook's pivoting* (RP), was introduced in Neal and Poole (Linear Algebra Appl. 173 (1992) 239–264) which encourages stability in the back-substitution phase of GE while controlling the growth of round-off error during the sweep-out. In fact, Foster (J. Comput. Appl. Math. 86 (1997) 177–194) has previously shown that RP, as with complete pivoting, cannot have exponential growth error. Empirical evidence presented in Neal and Poole (Linear Algebra Appl. 173 (1992) 239–264) showed that RP produces computed solutions with consistently greater accuracy than partial pivoting. That is, Rook's pivoting is, on average, more accurate than partial pivoting, with comparable costs. Moreover, the overhead to implement Rook's pivoting in a scalar or serial environment is only about three times the overhead to implement partial pivoting. The theoretical proof establishing this fact is presented here, and is empirically confirmed in this paper and supported in Foster (J. Comput. Appl. Math. 86 (1997) 177–194). © 2000 Elsevier Science B.V. All rights reserved.

MSC: 65F05; 65F35

Keywords: Gaussian elimination; Pivoting

1. Introduction

The geometric analysis of Gaussian elimination (GE) presented in [3,5,6] suggests that if the pivoting strategy used during the sweep-out phase (SWOP) of GE makes no attempt to control instability during the back-substitution phase (BSP), then the computed solution after back-substitution has been performed may bear little resemblance to the exact solution of the triangular system produced

* Corresponding author.
E-mail address: pooleg@etsu.edu (G. Poole).
[1] On October 16, 1995, Larry Neal died after a nine-year, fiercely fought battle with cancer. Without his technical expertise, creative mind, and incomparable friendship, this work could not have been completed. This work was sponsored in part by a grant from the East Tennessee State University Research Development Committee, #2-25417.

by the sweep-out. Rook's pivoting (RP) was introduced in [3] and is designed to simultaneously reduce round-off error during the SWOP and control instability that might arise during the BSP. Very simply, RP searches for pivots that are maximal in absolute value in both the row and column they reside. Partial pivoting (PP) abdicates any power to control BSP instability while complete pivoting (CP) does, in fact, exercise some control (during the SWOP) over the instability that might arise during the BSP. In fact, PP can admit exponential growth error while CP and RP do not [2]. In a minimal sense, RP resembles the pivoting strategy of Bunch and Kaufmann [1] applied to symmetric matrices. Sometimes referred to as "diagonal pivoting", the BK strategy, at each pivot selection step, searches two rows and columns in order to find a "pivoting element" (either of order one or order two) to maintain stability *and* symmetry throughout the SWOP. However, the similarity of the two strategies abruptly ends here.

Section 2 outlines the philosophy of the RP strategy. Section 3 contains an outline of a formal mathematical proof of the fact that the cost or overhead to implement RP in a scalar or serial environment is the same order of magnitude as the cost to implement PP. The empirical evidence supporting the theory in Section 3 is the subject of Section 4 (and corroborated in [2]). Section 5 contains empirical data to show that computed solutions based on RP are, on average, more nearly accurate than those based on PP. In Section 6, an example promised in an earlier paper [3], is presented to show a worst-case instability during the BSP of GE, in either CP or RP. This example underscores and clarifies the misunderstandings regarding the so-called "no problem phase" (i.e., BSP) of GE. Section 7 contains some conclusions and remarks. Finally, an amplified version with complete details is contained in [4].

2. The Rook's pivoting strategy

Suppose $\boldsymbol{A} = [a_{ij}]$ is a square matrix of order n and $\boldsymbol{Ax} = \boldsymbol{b}$ is a linear system to be solved by Gaussian elimination (GE). As with PP and CP, RP is a strategy based on the magnitude of elements of $\boldsymbol{A}$. In RP, the kth pivot is selected to have "dominion" (maximal absolute value) over *both the row and column* in which it lies. As does CP, RP confines its search for the kth pivot to elements in rows k through n and columns k through n of $\boldsymbol{A}^{(k)}$, the modified coefficient matrix *before* the kth step of the SWOP of GE is performed. Also, as with CP, RP selects for a pivot in the (k,k) position of the upper-triangular matrix $\boldsymbol{U}$ an entry whose magnitude is greater than or equal to the magnitudes of all other entries in both its column (to minimize round-off error in the SWOP) and in its row (to foster stability during the BSP). However, unlike CP, it is established theoretically in Section 3 (and confirmed empirically in Section 4 and [2]) that RP rarely requires a complete search of every entry in the unswept sub-matrix of order $(n-k+1)$ in the lower right corner of $\boldsymbol{A}^{(k)}$ in order to find a suitable kth pivot $(k=1,2,\ldots,n-1)$. To locate the kth pivot using RP, one performs a sequential search (column, row, column, etc.) of the remaining unsearched vectors until an element has been located whose magnitude in absolute value is not exceeded by any other element in either the row or column it resides. If $r>k$, rows r and k are interchanged. If $c>k$, columns c and k are interchanged. Then the kth step of the sweep-out phase of GE is performed on $\boldsymbol{A}^{(k)}$ to eliminate nonzero entries below the diagonal in column k, thus producing $\boldsymbol{A}^{(k+1)}$, $(k=1,2,\ldots,n-1)$. Consequently, the cost of implementing RP in a serial environment varies between twice the cost of PP and the full cost of CP. In fact, we shall *prove* that the *expected cost* of implementing RP in a serial environment is about three times the cost of PP. This is precisely the goal of the next section.

3. The cost of implementing Rook's pivoting

No matter what pivoting-for-size strategy is employed, the amount of arithmetic performed to complete the SWOP and BSP of GE is exactly the same. The difference in the costs between strategies lies in the number of compares required to determine the pivots, as well as any other costs that might arise from data access and data management (to be discussed shortly).

When unnecessary duplicate compares are avoided, the total cost in number of compares required to locate all $n-1$ pivots using RP varies between the cost of PP and the cost of CP, namely between $O(n^2/2)$ and $O(n^3/3)$. We shall prove that the *expected total number of compares* for RP is $O(3n^2/2)$ for all $n \geqslant 2$, even if duplicate compares are allowed. This compares favorably with the fixed cost of PP, namely $O(n^2/2)$ compares. As will be demonstrated in this section and the next, within the confines of a serial computing environment, *rarely* does the cost of RP approach the fixed cost of CP. Foster [2] empirically confirms, under independent testing, that RP is only slightly more costly than PP. Moreover, Foster has shown that RP does not have the exponential growth error that PP sometimes experiences.

Before outlining the proof regarding the expected costs of RP, there is one other issue we must discuss regarding the implementation of RP, namely the costs of data access. Unlike PP and CP in which data can be organized to take advantage of storage and retrieval in some architectures relative to some languages (i.e., by rows in a "C" environment or by columns in a FORTRAN environment), the RP strategy necessitates that one access data by both rows and columns. On the surface, one might question the use of RP for large linear systems since severe page faulting may arise throughout the process of searching for its pivots. The investigation into this issue depends on the size of the linear system and the size of machine memory. However, a careful but simple analysis of the entire process of GE (including data management, the pivot selection strategy, the SWOP and the BSP), one can easily show for a given fixed memory size (pages of memory available), that as the size n of the linear system increases, the percentage of time spent in page faults for the pivot-selection phase against the total time of the GE algorithm decreases towards zero. In other words, if page faulting becomes a problem for pivot selection, it becomes a problem for the entire algorithm, especially in view of the theory presented in this section and the empirical evidence presented in the next section. Besides, cache and register management is more of a concern today for large data sets than page faults. For example, LAPACK involves a block algorithm design to address good cache behavior. Finally, all the numerical experiments found in [2], and in this paper, confirm the fact that RP is only slightly more expensive than PP in a serial environment.

We begin to outline our proof with a theoretical demonstration. Suppose we are attempting to locate the first Rook's pivot, labeled p^*, in a linear system of order n. Here, $\boldsymbol{V}$ may denote either a row vector or column vector. Eventually, we wish to (1) compute P_i, the probabilities that p^* is in $\boldsymbol{V}_i$, the ith vector searched (and in no previous vector $\boldsymbol{V}_j$, $j < i$), and (2) compute $C(i)$, the total number of compares required to confirm that p^* is in the ith vector searched (and in no previous vectors) using the RP strategy. To perform these computations, we shall define a *random variable* C whose possible values depend on the size of the matrix and the (variable) number of vectors that must be searched in order to locate a suitable Rook's pivot. To this end, we offer a few preliminary observations and establish some basic notation.

We shall say that a component t of the vector $\boldsymbol{V}$ is *maximal in* $\boldsymbol{V}$ if it is the first element encountered (top-to-bottom of a column vector or left-to-right of a row vector) whose magnitude is greater than or equal to the magnitude of every component of $\boldsymbol{V}$. Also, we shall use $L(\boldsymbol{V})$ to denote the number of components in $\boldsymbol{V}$.

First, consider a coefficient matrix $\boldsymbol{A}$ of order n in which the first Rook's pivot, p^*, happens to be in column number one. This means that p^* is the first element in a top-to-bottom search of column one whose magnitude is greater than or equal to the magnitude of all other elements of column one, and whose magnitude (by chance) happens to be greater than or equal to the magnitudes of all other elements to its right in the same row. For convenience, it is initially assumed that p^* is in row r of $\boldsymbol{A}$. That is, a_{r1} is maximal in the first column vector of the matrix $\boldsymbol{A}$.

Let $\boldsymbol{V}_1 = [a_{11}\ a_{21}\ \cdots\ a_{n1}]^{\mathrm{t}}$ denote the n-component vector in *column* one of $\boldsymbol{A}$ and let $\boldsymbol{V}_2 = [a_{r2}\ a_{r3}\ a_{r4}\ \cdots\ a_{rn}]$ denote the $(n-1)$-component vector in *row* r of $\boldsymbol{A}$ which omits the column one component. So $|p^*|$ is at least as big as the magnitudes of every entry in $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$. We shall describe this situation by saying that the maximal entry p^* of $\boldsymbol{V}_1$ is *maximal through* $\boldsymbol{V}_2$.

Let P_k be the probability that a suitable Rook's pivot is located in the kth vector searched (and in no previous vectors). Then P_1 is the probability that p^* is in the first column of the matrix $\boldsymbol{A}$. That is, P_1 is the *conditional probability* that p^* is maximal through $\boldsymbol{V}_2$ given that p^* is maximal in $\boldsymbol{V}_1$. The probability that p^* is maximal in $\boldsymbol{V}_1$ is just $1/L(\boldsymbol{V}_1) = 1/n$. The probability that p^* is maximal through $\boldsymbol{V}_2$ is $1/[L(\boldsymbol{V}_1)+L(\boldsymbol{V}_2)] = 1/(n+n-1)$. So,

$$P_1 = \frac{1/(L(\boldsymbol{V}_1)+L(\boldsymbol{V}_2))}{1/L(\boldsymbol{V}_1)} = \frac{L(\boldsymbol{V}_1)}{L(\boldsymbol{V}_1)+L(\boldsymbol{V}_2)} = \frac{n}{2n-1}. \tag{3.3}$$

Moreover, the number of compares required to confirm that $p^* = a_{r1}$ is the first pivot is given by

$$C(1) = (L(\boldsymbol{V}_1)-1) + L(\boldsymbol{V}_2) = (n-1)+(n-1) = 2(n-1). \tag{3.4}$$

Before determining the remaining probabilities, P_2 through P_{2n-1}, it is important to clarify the assumptions used in computing P_1, which are also used in computing the remaining probabilities P_i. In the absence of knowing the initial matrix pattern (for example, symmetric, diagonally dominant, banded, etc.), we shall assume the entries of the coefficient matrix are randomly distributed among the positions of the matrix. Moreover, in the absence of knowing the initial distribution of the entries themselves (uniform, normal, symmetric, skewed, etc.), we shall assume the entries of the coefficient matrix form a uniformly distributed set.

The first assumption is not unreasonable. The more random the distribution of the matrix entries, the more time must be consumed to locate a suitable Rook's pivot. Likewise, the second assumption is not unreasonable because Trefethen and Schreiber [8] have observed that whatever the distribution of the entries in the initial coefficient matrix, the entries in the sequence $A^{(k)}$ of submatrices determined by PP *tend toward a normal distribution*. We believe the same property holds for RP and will assume this to be the case. With these assumptions we shall proceed with the computation of probabilities.

Now, suppose a_{r1} is maximal in $\boldsymbol{V}_1$, but a_{rs} $(s>1)$ is maximal in $\boldsymbol{V}_2$ so that $|a_{r1}| < |a_{rs}|$. Then a_{r1} is not the first Rook's pivot, and the magnitudes of the $n-1$ components of the column vector $\boldsymbol{V}_3 = [a_{1s}\ a_{2s}\ \cdots\ a_{r-1,s}\ a_{r+1,s}\ \cdots\ a_{n,s}]^{\mathrm{t}}$ must be compared to $|a_{rs}|$ to determine if $p^* = a_{rs}$. In what follows, it is important to understand how the sequence of vectors $\{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_k\}$ is determined and defined using the RP strategy to locate p^*.

In searching for p^* in any matrix $\boldsymbol{A}$ of order n, $\boldsymbol{V}_1$ is the first vector searched (column one of $\boldsymbol{A}$), $\boldsymbol{V}_2$ is the second vector searched (the partial row of $\boldsymbol{A}$ corresponding to the entry maximal in $\boldsymbol{V}_1$, omitting the component common to $\boldsymbol{V}_1$), and $\boldsymbol{V}_i$ is the ith vector searched, omitting any components in $\boldsymbol{V}_j$, for $j=1,\dots,i-1$. If $i>1$ is odd, $\boldsymbol{V}_i$ is a partial column vector of $\boldsymbol{A}$; otherwise, $\boldsymbol{V}_i$ is a partial row vector. Note that $L(\boldsymbol{V}_1)=n$, $L(\boldsymbol{V}_2)=L(\boldsymbol{V}_3)=(n-1)$, and $L(\boldsymbol{V}_4)=L(\boldsymbol{V}_5)=n-2$, and so forth. Furthermore, it is not difficult to show

$$L(\boldsymbol{V}_i)=n-[i/2] \quad \text{for } i=1,2,\dots,(2n-1), \tag{3.5}$$

where $[i/2]$ denotes the greatest integer function. Note that $L(\boldsymbol{V}_i)\geqslant L(\boldsymbol{V}_{i+1})$ for each $1\leqslant i\leqslant 2n-1$, and $L(\boldsymbol{V}_{2n-2})=L(\boldsymbol{V}_{2n-1})=n-(n-1)=1$. So $n\geqslant L(\boldsymbol{V}_i)\geqslant 1$ for each $1\leqslant i\leqslant 2n-1$. Also note that $L(\boldsymbol{V}_{2n})=0$.

Now suppose $\{\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}_3,\dots,\boldsymbol{V}_k,\boldsymbol{V}_{k+1}\}$ is the *exact* sequence of vectors searched to locate a suitable Rook's pivot p^* where $k\geqslant 1$. If $k<(2n-2)$, then k represents the smallest integer for which (1) p^* is maximal in $\boldsymbol{V}_k$, (2) p^* has magnitude greater than or equal to the magnitude of each component in $\boldsymbol{V}_{k+1}$, and (3) p^* has magnitude strictly greater than the magnitude of each component of the vectors in $\{\boldsymbol{V}_1,\boldsymbol{V}_2,\dots,\boldsymbol{V}_{k-1}\}$. In this case, p^* is in $\boldsymbol{V}_k$, the second to last vector searched and we shall say that p^* is *maximal up through* $\boldsymbol{V}_{k+1}$. In case $k=(2n-2)$, p^* could be in either $\boldsymbol{V}_{2n-2}$, the second-to-last vector searched or in $\boldsymbol{V}_{2n-1}$, the last vector searched (a column vector of length one).

Now P_2 is the probability that p^* is not in $\boldsymbol{V}_1$ multiplied by the conditional probability that p^* is maximal up through $\boldsymbol{V}_3$ given that p^* is maximal in $\boldsymbol{V}_2$. That is,

$$P_2=(1-P_1)\frac{1/(L(\boldsymbol{V}_1)+L(\boldsymbol{V}_2)+L(\boldsymbol{V}_3))}{1/(L(\boldsymbol{V}_1)+L(\boldsymbol{V}_2))}=\frac{n-1}{3n-1}. \tag{3.6}$$

Also, the number of compares required to confirm that p^* is in $\boldsymbol{V}_2$ is given by

$$C(2)=(L(\boldsymbol{V}_1)-1)+L(\boldsymbol{V}_2)+L(\boldsymbol{V}_3)=C(1)+L(\boldsymbol{V}_3)=3(n-1). \tag{3.7}$$

To determine P_k for $k=3,4,\dots,2n-1$, we list the following lemmas, corollaries, and theorems, but for the sake of space, proofs are omitted. However, some proofs are rather technical, but easy [4].

Lemma 3.1. *For all $n\geqslant 2$ and for each $k=2,3,\dots,2n-1$,*

$$P_k=\frac{(1-\sum_{i=1}^{k-1}P_i)(\sum_{i=1}^{k}L(\boldsymbol{V}_i))}{\sum_{i=1}^{k+1}L(\boldsymbol{V}_i)}.$$

Lemma 3.2. *For all $n\geqslant 2$ and for any $k=2,3,\dots,2n$,*

$$1-\sum_{i=1}^{k-1}P_i=\prod_{j=2}^{k}\left(\frac{L(\boldsymbol{V}_j)}{\sum_{i=1}^{j}L(\boldsymbol{V}_i)}\right).$$

Corollary 3.3. *For all $n\geqslant 2$, $\sum_{i=1}^{2n-1}P_i=1$.*

Lemma 3.4. *For any $n\geqslant 2$ and for all $k=2,3,\dots,2n-1$,*

$$P_k=\prod_{j=2}^{k-1}\left(\frac{L(\boldsymbol{V}_j)}{\sum_{i=1}^{j}L(\boldsymbol{V}_i)}\right)\left(\frac{L(\boldsymbol{V}_k)}{\sum_{i=1}^{k+1}L(\boldsymbol{V}_i)}\right).$$

Lemma 3.5. *For any integer* $j \geqslant 2$, $j[j/2] - \sum_{i=1}^{j} [i/2] > 0$.

Lemma 3.6. *For any* $n \geqslant 2$ *and for all* $k = 2, 3, \dots, 2n-1$,

$$0 < \left(\frac{L(\boldsymbol{V}_j)}{\sum_{i=1}^{j} L(\boldsymbol{V}_i)} \right) < \frac{1}{j}.$$

Lemma 3.7. *For any* $n \geqslant 2$ *and for all* $k = 2, 3, \dots, 2n-1$,

$$0 < \left(\frac{L(\boldsymbol{V}_k)}{\sum_{i=1}^{k+1} L(\boldsymbol{V}_i)} \right) < \frac{1}{k+1}.$$

Theorem 3.8. *For any* $n \geqslant 2$ *and for all* $k = 2, 3, \dots, 2n-1$,

$$0 < P_k < \frac{k}{(k+1)!}$$

Proof. Follows immediately from Lemmas 3.4, 3.6 and 3.7. □

Theorem 3.8 provides practical and useful upper bounds for P_k $(k = 2, 3, \dots, 2n-1)$ which are *independent of n*, the size of the square matrix being searched using the Rook's pivot strategy. These bounds show just how unlikely it is that the RP search strategy requires anything near an entire search of the matrix $\boldsymbol{A}$ before an entry is found whose magnitude is maximal in both its row and its column.

Eq. (3.3) provides a lower bound of $\frac{1}{2}$ for P_1 if $n \geqslant 2$. So a rather *important observation* about PP is that for any position, PP selects a pivot that is maximal in its row as well as in its column with a probability at least $\frac{1}{2}$. *This is precisely the reason PP is so successful in practice*. At least half the time PP makes the correct choice for a pivot.

Corollary 3.3 and the first inequality of Theorem 3.8 show that the P_i's, for $i = 1, 2, \dots, 2n-1$ satisfy the criteria to be a valid probability distribution. So the number of compares required to locate the first Rook's pivot p^* in a matrix of order n is indeed a *random variable C* with $2n-1$ integer values (one for each P_k). Now $C(k)$ denotes the number of compares required to confirm that the first pivot is found in $\boldsymbol{V}_k$, the kth vector searched, and not in any previous vector $\boldsymbol{V}_i$ for $i = 1, 2, \dots, k-1$ using the RP strategy. We have seen that $C(1) = 2(n-1)$ is the minimum value of this random variable (Eq. (3.4)).

Theorem 3.9. *For any* $n \geqslant 2$ *and for each* $k = 2, 3, \dots, 2n-1$,

$$C(k) = (k+1)(n-1) - \left[\frac{(k-1)^2}{4} \right].$$

Proof. Note that $C(k) = C(k-1) + L(\boldsymbol{V}_{k+1}) = C(k-1) + n - [(k+1)/2]$. Now use induction on k and Eqs. (3.4) and (3.7).

From Theorem 3.9,

$$C(2n-1) = C(2n-2) = n^2 - 1. \quad \square \tag{3.8}$$

Just before proving the main theorem, by considering the previous lemmas and theorems, notice what is involved to determine the expected number of compares E_2 to locate the first pivot for a system of order $n = 2$. That is, since $2n - 1 = 3$, $C(2) = C(3) = 3$ and we have

$$E_2 = \sum_{k=1}^{3} P_k C(k) = (\tfrac{2}{3})(2) + (\tfrac{1}{5})(3) + (\tfrac{2}{15})(3) = \tfrac{7}{3}. \tag{3.9}$$

We are now prepared to state and prove the main theorem in this paper which establishes that, on average and under the stated assumptions, RP requires approximately $3(n-1)$ compares to locate the first pivot of a matrix of order n. That is, RP is about three times more expensive than PP.

Theorem 3.10. *The expected number of compares E_n required to locate the first Rook's pivot p^* in a linear system of order $n \geqslant 2$ is* $\mathrm{O}(3(n-1))$.

Proof. By Eq. (3.9), $E_2 = \frac{7}{3} < 3(n-1)$ when $n = 2$. Now assume that $n > 2$ and note that $En = \sum_{k=1}^{2n-1} C(k)P_k$. Consequently, E_n satisfies the following equalities and inequalities:

$$\begin{aligned}
E_n &= \sum_{k=1}^{2n-1} \{(k+1)(n-1) - [(k-1)^2/4]\} P_k \\
&\leqslant \sum_{k=1}^{2n-1} (k+1)(n-1)P_k = (n-1)\sum_{k=1}^{2n-1}(k+1)P_k \\
&< (n-1)\left(2\left(\frac{n}{2n-1}\right) + \sum_{k=2}^{2n-1}\left(\frac{(k+1)k}{(k+1)!}\right)\right) \\
&= (n-1)\left(\left(\frac{2n}{2n-1}\right) + \sum_{k=2}^{2n-1}\left(\frac{1}{(k-1)!}\right)\right) \\
&< (n-1)\left(1 + \frac{1}{2n-1} + e - 1\right) \\
&= \left(\frac{n-1}{2n-1}\right) + e(n-1) \\
&< \tfrac{1}{2} + e(n-1) \\
&< 3(n-1) \quad \text{for all } n > 2.
\end{aligned}$$

Therefore, $\mathrm{O}(E_n) \leqslant \mathrm{O}[e(n-1)] \leqslant \mathrm{O}[3(n-1)] = \mathrm{O}(n-1)$. □

Note that the first inequality above abandons the assumption that unnecessary duplicate compares are avoided. By way of contrast, recall the number of compares required to locate the *first pivot* using PP and CP are, respectively, $(n-1)$ and (n^2-1). So, with $n \geqslant 2$, the expected cost of locating the *first pivot* using RP is approximately 3 times the cost of PP, even when unnecessary duplicate

Table 1
Cost in number of compares to locate all $n-1$ pivots of a system of order n using the PP, CP, and RP strategies

	Number of compares required to locate all $n-1$ pivots using			
n	PP $\sum_{i=2}^{n}(i-1)$	CP $\sum_{i=2}^{n}(i^2-1)$	RP (expected) $\sum_{i=1}^{2n-1}P(i)C(i)$	RP (expected) approximated $\frac{e(n-1)n}{2}$
2	1	3	2.333	2.7
5	10	50	25.125	27.2
10	45	375	117.041	122.3
50	1225	42 875	3296.626	3329.9
100	4950	338 250	13 386.490	13 455.5
10^3	499 500	333 832 500	1357063.656	1357781.8
10^4	4 999 500	$3.33383325 \times 10^{11}$	1.35893×10^8	1.3590×10^8
10^5	4.99995×10^9	$3.3333833325 \times 10^{14}$	1.35912×10^{10}	1.35913×10^{10}

compares are *not* avoided. As we shall observe, the expected cost to locate all $(n-1)$ RP pivots is about 3 times the cost to locate all PP pivots.

The nature of the random variable C, that is, the formulas that give the number of compares required to locate a suitable Rook's pivot (Theorem 3.9) and the probabilities associated with these number of compares (Lemma 3.4 and Theorem 3.8) are assumed to be the same for any square sub-matrix of the original square matrix of order n. Thus, one need only replace n with $n-q$ in Theorems 3.8, 3.9 and Lemma 3.4 to obtain the random variable C and its associated probability distribution for the number of compares to locate the $(q+1)$th Rook's pivot $(q=1,2,\dots,n-2)$.

Corollary 3.11. *The expected total number of compares E_T required to locate all $(n-1)$ Rook's pivots is* $O(3n^2/2)=O(n^2)$.

Proof. $E_T \sim \sum_{i=2}^{n} 3(i-1) = 3\sum_{i=2}^{n}(i-1) = 3\sum_{k=1}^{n-1} k = 3(n-1)n/2$.

By way of comparison, the total number of compares required to locate all $(n-1)$ pivots by PP (Eq. (3.1)) and CP (Eq. (3.2)) are, respectively, $(n-1)n/2$ and $(2n^3+3n^2-5n)/6$. □

Table 1 shows the cost in number of compares required to implement partial, Rook's, and complete pivoting on linear systems of order n. To insure accuracy, the third column of Table 1 was obtained using Lemma 3.4 and Theorem 3.9 and extended precision (19–20 digit floating point arithmetic).

Under the stated assumptions about the initial coefficient matrix and the subsequent, sweep-out induced submatrices $A^{(k)}$, it is clear that RP and PP are similar in costs and both are orders of magnitude less expensive than CP. Moreover, Section 5 will contain empirical evidence establishing that the accuracy expected from RP is superior to that expected from PP. This is due, in part, to two factors: First, RP cannot have exponential error growth as can PP [2]. Second, unlike PP, the favorably oriented hyperplanes (i.e., the measure of orthogonality between the rows of U in the LU decomposition) produced by RP fosters a high degree of stability during the back-substitution phase (see [3,5,6,9]).

Table 2
Ratio of number of compares using RP to number of compares using PP for randomly generated matrices with entries which represent either a uniform distribution or normal distribution

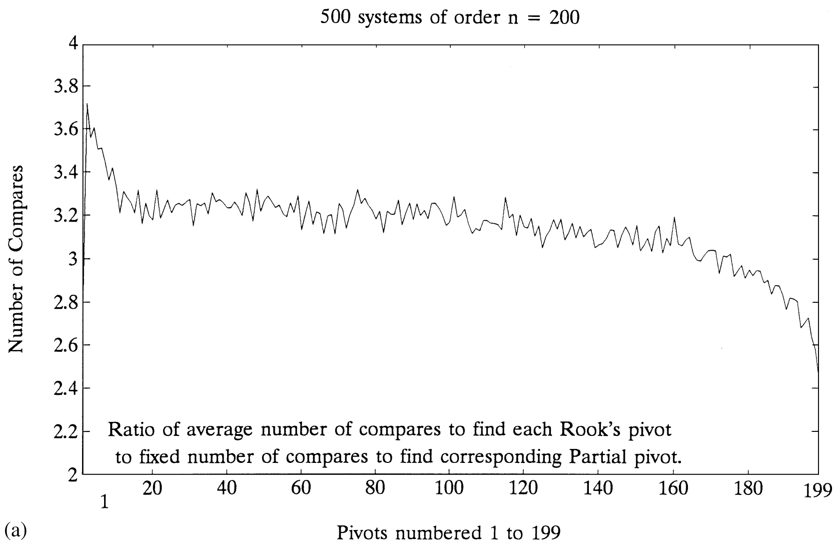
n matrix order	Number of systems (k)	Uniform dist A_n/W_n	Normal dist A_n/W_n
10	1000	2.719	2.658
15	1000	2.770	2.709
20	1000	2.819	2.753
25	1000	2.872	2.782
50	1000	2.989	2.894
75	1000	3.053	2.966
100	1000	3.107	3.021
125	1000	3.136	3.055
150	1000	3.163	3.096
200	500	3.2184	3.149
300	200	3.2787	3.215
400	150	3.3146	3.273
500	100	3.3432	3.305

4. Empirical evidence on cost of Rook's pivoting

This section contains empirical data supporting the conclusions of Theorem 3.10 regarding the cost of locating the first Rook's pivot. Moreover, we compare the cost of locating all of the Rook's pivots to the cost of locating all the pivots by PP and CP.

The power of MATLAB [11] and specially designed M-files were exploited to generate, massage, and provide appropriate graphs to empirically compare the cost of RP to PP. For certain values k and n, k coefficient matrices of order n were generated whose entries were *uniform* over the interval $[-10^4, 10^4]$, and *randomly distributed* by position. The actual number of compares required to implement the RP strategy was calculated for each system and the average A_n was determined over the k generated systems of order n. Then, to compare RP to PP, the quotient $A_n/[n(n-1)/2] = A_n/W_n$ was computed for each size n. We repeated this experiment using a *normal distribution* of the entries in the coefficient matrix which were *randomly distributed* across the positions of the matrix. The results are provided in Table 2.

In addition, for each matrix, the number of vectors searched required to locate each of the $n-1$ Rook's pivots was saved. From this information, the number of compares required to locate each of the pivots could be computed. Then, over the k generated matrices, the average number of compares was computed. The pivot which generated the largest average over the k matrices was also determined. From the data generated for each triple ($n=$ size, type $=$ uniform or normal, $k=$ How many matrices), a sequence of graphs was produced. The purpose of studying these sequences of graphs was to observe the behavior of RP as the size of the coefficient matrix increased, for either the uniform case or the normal case. Our analysis included systems of orders up to 1000 and was much more extensive than the results presented here. However, we will *limit our presentation* to matrices of order $n=200$ and provide a sample of the results. Figs. 1a and b represent data from



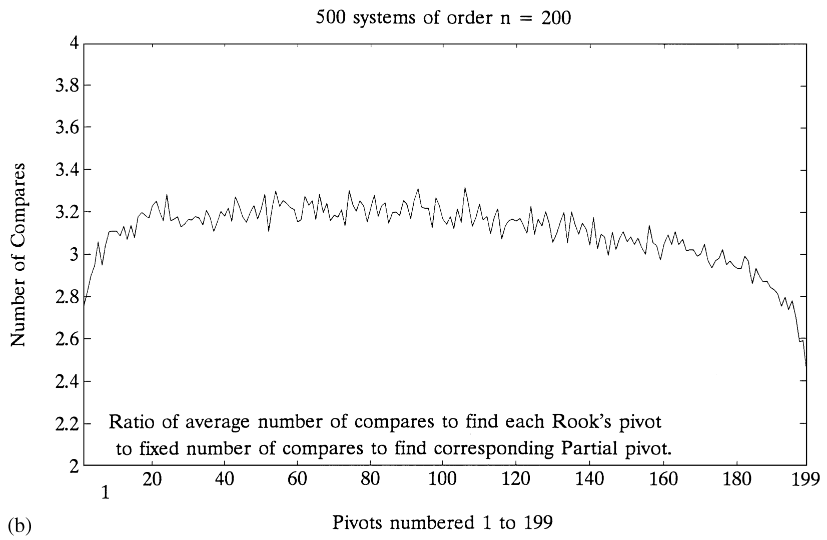


Fig. 1. (a) Uniform. (b) Normal.

500 uniform and 500 normal systems, each of order 500. Additional data and graphs are contained in [4].

There are three important observations regarding Figs. 1a and b. *First*, since the theory presented in Section 3 is based on uniform distributions and all 500 coefficient matrices were designed to have such an *initial* distribution, it is not surprising that the ratio of averages for the *first pivot* is approximately Napier's ***e***, as theorized and expected. In fact, by running a simulation over 600 matrices of orders 10–300, in which each submatrix throughout the sweep-out phase was designed to have a normal distribution, the average number of compares for RP was 2.7157 times the cost of

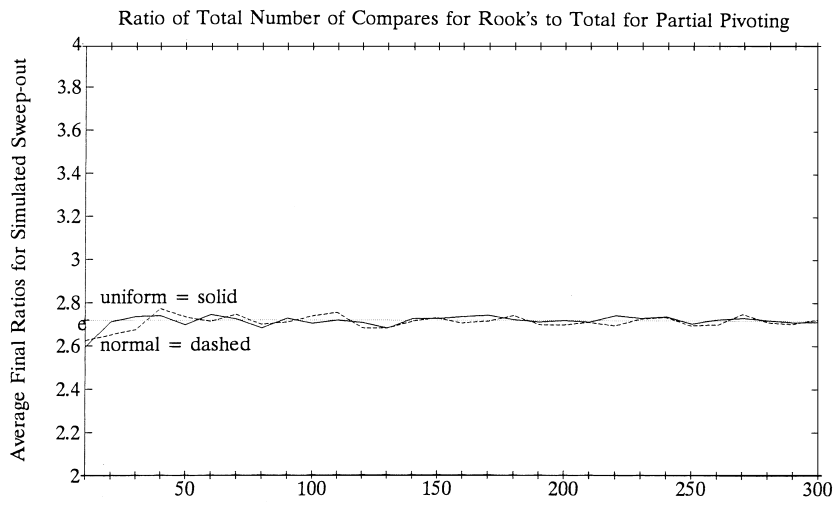


Fig. 2.

PP. By running an identical simulation for matrices with normal distributions, the factor was 2.7105. The graphs of this simulation are presented in Fig. 2. *Second*, since an initial uniform distribution challenges the RP strategy more than an initial normal one, the pivot requiring the largest number of compares, on average, is realized earlier in the uniform case than the normal one. *Third*, as noted in [8], the distribution of the elements in $A^{(k)}$ tends toward normal throughout the SWOP. This fact explains why the two graphs in Figs. 1a and b resemble each other as the SWOP progresses. Namely, the graphs tend toward the same number of vectors searched, or the same ratio of compares (Rook's to partial). *Fourth*, even though $A^{(k)}$ tends towards a normal distribution as k increases, "normal" does not have much meaning when $n < 6$ and the number of elements in the coefficient matrix is less than 30. This explains why the last portion of these two graphs drops off accordingly at the end. That is, when the SWOP of Gaussian elimination has reduced the search for the Rook's pivot from a given matrix A of order n to a matrix $A^{(k)}$ of order less than 6, the number of compares required to locate the Rook's pivots is significantly less than e.

5. Empirical evidence on accuracy of Rook's pivoting

The geometric analysis of both phases of GE presented in [3,5] clearly indicates that a pivoting-for-size strategy which selects as pivots those entries whose magnitudes are maximal in *both* their column and their row will, on average, produce more nearly accurate results than PP. CP is such a strategy, but it is cost prohibitive. As demonstrated in the previous section, RP is such a strategy whose *magnitude of cost* is the same as PP, far less than the cost for CP. In this section we compare the average total error between solutions of linear systems computed by PP, MATLAB's A\b Routine (ML), RP, and CP. These linear systems were designed to provide a range of examples with the following properties: (1) All entries and solutions are exactly machine representable, (2) all systems are solved in IEEE double-precision, (3) the entries of each matrix are either normally or uniformly distributed among themselves and the magnitudes of the entries in the coefficient matrices lie in the interval $[10^{-4}, 10^{4}]$, (4) the collection of systems and their condition numbers cover the spectrum from 10^{6} to 10^{18}, (5) systems of order 25–500 were considered in the analysis. However, we only discuss and present the graphs for the case $n = 100$. The results are comparable for systems of other orders [4].

The method by which these linear systems were crafted is explained in Section 3 of Neal and Poole [3]. Briefly, thousands of integer linear systems with integer solutions were randomly generated and then modified by the technique of Rice [7] to assume condition numbers within some designated interval, usually $[10^{6}, 10^{18}]$. *Note*: The reason for *not* including linear systems with condition numbers inside the interval $[10^{0}, 10^{6}]$ is that in double-precision arithmetic, the errors resulting from any one of the four methods (PP, ML, RP, CP) is insignificant. That is, double-precision is sufficient to minimize the effects of round-off error during the SWOP and to *mask* any instability during the BSP, even for moderately ill-conditioned systems.

To compare the accuracies of the double-precision-computed solutions between the four pivoting strategies, over 6000 linear systems of order 100 were generated whose coefficients represented a *Normal distribution*, and whose condition numbers spanned the interval $[10^{6}, 10^{18}]$. Each of these linear systems was solved by each of the four methods: PP, RP, CP and ML. Fourteen pieces of data were stored for each of the linear systems in a 6000 by 14 data file called *datanorm.ill*: L_2-norm condition number, the seed producing the system, the L_2-norm error from each of the four methods, L-infinity norm error from each of the four methods, and the back-substitution phase error multipliers (a single vector) for each of the four strategies. To summarize the process of graphically comparing the accuracies between PP, RP, CP and ML, we constructed an M-file, called *plotill.m*, designed to massage the *datanorm.ill* file in the following manner: For size = 100, *plotill* (size, increments, minlog, maxlog) (1) loads the data file *datanorm.ill* and accepts as input $k =$ increments, and $6 \leqslant \text{minlog} < \text{maxlog} \leqslant 18$, then sets $w = \text{maxlog} - \text{minlog}$; (2) partitions the interval $[\text{minlog}, \text{maxlog} + 1/k]$ into $w * k + 1$ subintervals or "bins" of equal width $1/k$; (3) scans the array *datanorm.ill* and locates all systems with condition numbers COND such that $\log_{10}(\text{COND})$ is contained in the interval $[\text{minlog}, \text{maxlog} + 1/k]$; (4) for each bin, the average L_2-norm error is computed for each of the four pivoting strategies over the systems in the bin. The resulting four plot points are assigned with first component being the *average of the condition numbers* over the systems in the bin. Any bin not containing a minimum of 30 systems is omitted; (5) plots four polygonal graphs, one for each of PP, RP, ML and CP, with horizontal axis ranging over [10minlog, 10maxlog] and vertical axis ranging over [0, maxerror +], automatically scaled by MATLAB [11].

The generation of data was repeated for linear systems of order 100 whose coefficients represented a *uniform distribution.* As long as $(\text{minlog} - \text{maxlog}) \geqslant 2$, the resulting graphs all had an appearance similar to that represented by the four graphs in Figs. 3–6.

Fig. 3 contains four graphs generated from the data bank *datanorm.ill* (containing 6000 normal systems) when the parameters entered for M-file *plotill* were minlog = 6, maxlog = 9, and increments = 2. The L_2-norm (rather than the L-infinity norm) was selected to measure the error. The code at the top of each figure matches the pivoting strategy to its graph type. The vertical scale is based on the maximum L_2-norm error. In Fig. 3, the error ranged from *very small* to approximately 4.5×10^{-8}. The text inside the figure indicates that 1668 of the 6000 systems had coefficient matrices with condition numbers in the interval $[10^{6}, 10^{9}]$.

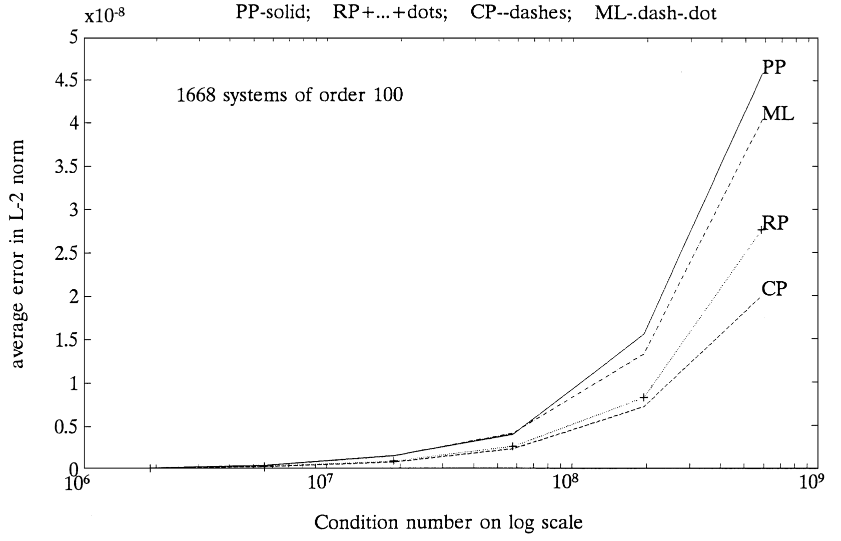


Fig. 3. Normal.

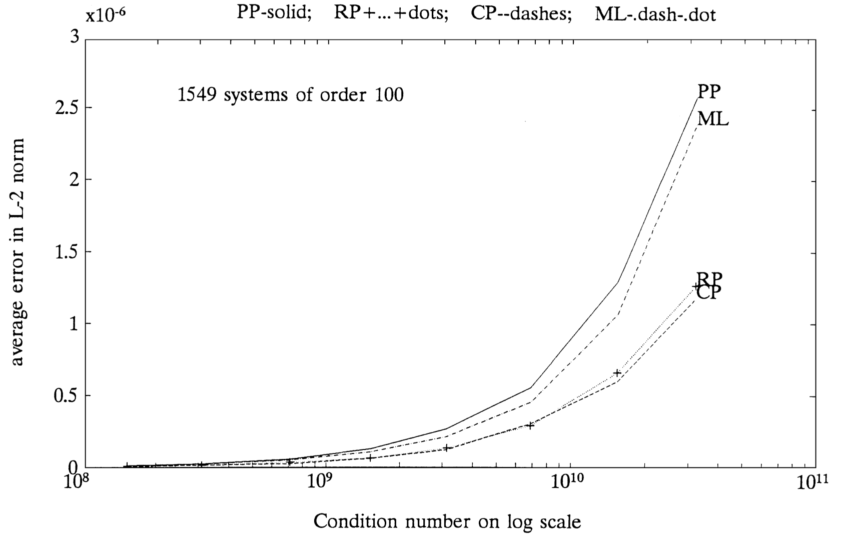


Fig. 4. Normal.

The parameters and data sets give in Table 3 were used to generate the next three sets of graphs in Figs. 4–6.

The magnitudes of the average L_2-norm errors in each bin were generally ordered (largest to smallest) as PP$\geqslant$ML$\geqslant$RP$\geqslant$CP. Moreover, RP tracked closely with CP, while MATLAB's method tracked closely with PP. It might be noted here that MATLAB (ML) uses the code that was historically attributed to Fortran's management of data, namely by columns and not rows. On the other hand, our code for PP uses the traditional, pedagogical management of data, namely by rows. It is

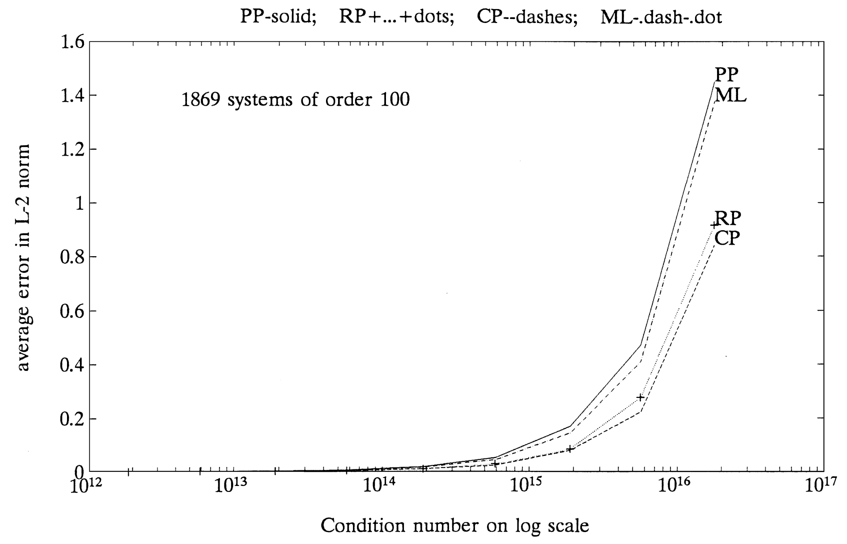


Fig. 5. Uniform.

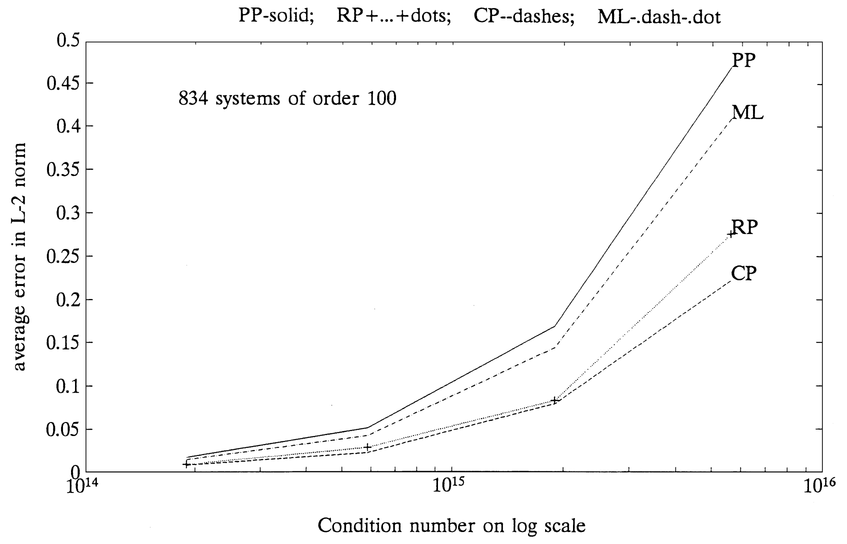


Fig. 6. Uniform.

Table 3

Figure	Data set	Condition	Increments	No. of sys.
5.2	Datanorm	$[10^8, 10^{10.5}]$	3	1549
5.3	Dataunif	$[10^{12}, 10^{16.5}]$	2	1869
5.4	Dataunif	$[10^{14}, 10^{16}]$	2	834

the difference in the order of arithmetic that explains the difference in computed solutions. However, the explanation of why ML does a better job than PP, on average, is not yet fully understood.

6. Worst-case instability in back substitution

As noted in [3], computing and reviewing the back-substitution phase error multipliers (BSP EMs) is an inexpensive way to check for instability in the back-substitution phase of GE. It is our experience that the magnitudes of all BSP EMs for either CP or RP rarely exceed 10 [3]. However, it can be shown by induction that in the worst case, CP and RP may produce upper-triangular systems for which $\|m\|_\infty = 2^{n-2}$ where n is the dimension of the system [3]. So even CP and RP might produce upper-triangular systems for which the BSP is unstable, a rarity under either strategy. We illustrate this fact with the following example which was promised in [3, p. 254].

Example 6.1. Recall from [3, p. 247] that the BSP error multipliers can be obtained by solving the triangular system $\boldsymbol{U}\boldsymbol{y} = \varepsilon$ where $\boldsymbol{U}$ is the triangular matrix derived from the sweep-out phase of GE and $\varepsilon = [0, 0, \ldots, 0, \boldsymbol{u}_{nn}]^t$. Consider the upper triangular matrix $\boldsymbol{U}$ of order n for which $u_{ii} = 1$, for $i = 1$ to n and $u_{ij} = -1$ for $j \geqslant i$. $\boldsymbol{U}$ could be obtained through the sweep-out phase using either CP or RP (or any other strategy which leaves $|u_{ii}| \geqslant |u_{ij}|$ for each i). Solving $\boldsymbol{U}\boldsymbol{y} = \varepsilon$, one finds that $\boldsymbol{m} = [2^{n-2}, 2^{n-3}, \ldots, 2, 1, 1]^t$ and hence, $\|m\|_\infty = 2^{n-2}$. Thus, apart from any round-off error that may occur during the back-substitution phase, if there is any error in the first computed component, no matter how small or seemingly insignificant, the error will be magnified (instability) as it is propagated through the back-substitution process (see example in [5, p. 257]). However, as noted earlier, such examples are very pathological and our experiences indicate that usually $\|m\|_\infty \leqslant 10$ with RP.

7. Conclusions

In addition to the many remarks found in the conclusions of Neal and Poole [3] and Poole and Neal [5,6] three additional practical conclusions are provided here, as well as one philosophical remark.

First, as with CP, RP usually produces an upper-triangular system whose hyperplanes are very well-oriented with respect to their corresponding coordinate axes [3,5,9]. Also, as with CP, RP usually produces BSP EMs whose magnitudes are much smaller than those produced by PP. That is, empirical evidence suggests that except for highly contrived pathological systems, RP and CP usually produce upper-triangular systems whose corresponding hyperplanes are well-oriented with respect to their coordinate axes, but more importantly, they are also well-oriented with respect to each other [3,9]. So, if the sweep-out phase of GE is performed in double-precision using the RP strategy, round-off error during the sweep-out phase is usually well controlled and the back-substitution phase is almost always numerically stable [3,9].

Second, if one insist on using PP with Gaussian elimination to solve a large linear system, one should calculate the back-substitution phase error multipliers [3]. If any are large compared to the precision used for the calculations, then iterative refinement should be used after the back-substitution phase has been completed, or RP should be used. See Foster [2] for a "partial Rook pivoting" strategy, one threshold in nature.

Three, the important point made in Section 3 is worth repeating. That is, PP is very successful in practice because at least half the time it selects a pivot which controls round-off error during the SWOP while fostering stability during the BSP.

Philosophical point: The future of computing technology and management of binary information is unknown. Today's literature contains new creative ways to manage GE that differ significantly from former or even contemporary techniques [10]. Many current philosophies and practices are reflections of old computer architecture, one-dimensional data structures, and deified programming languages. History is very important. But in the area of electronic manipulation of binary information, sharing knowledge and perspectives among those who aspire to compute is more important than controlling research based on yesteryear's beliefs. David Wise [10] says it better than we: "Such a trend (separating ourselves from the knowledge and perspectives of others) is not only scientifically, but politically foolish; we are all colleagues learning how to compute. We must better share problems, solutions, styles, techniques, and philosophy".

Acknowledgements

We appreciate the care and patience required of the referees who "walked" through this paper without many details available to them, but contained in [4]. Cleve Ashcraft read a version of this paper and suggested several good improvements. Moreover, we are grateful and indebted to the editorial staff of *CAM* for their support and encouragement, particularly Apostolos Hadjidimos. Finally, we are forever grateful to John R. Rice for his "underwing" encouragement throughout the last decade of the 20th century.

References

[1] J. Bunch, L. Kaufmann, Math. Comput. 31 (137) (1977) 163–179.

[2] L. Foster, The growth factor and efficiency of Gaussian elimination with rook pivoting, J. Comput. Appl. Math. 86 (1997) 177–194.

[3] L. Neal, G. Poole, A Geometric analysis of Gaussian elimination, II, Linear Algebra Appl. 173 (1992) 239–264.

[4] L. Neal, G. Poole, The Rook's pivoting strategy, Technical Report GP112595001, ETSU, 1995.

[5] G. Poole, L. Neal, A geometric analysis of Gaussian elimination, I, Linear Algebra Appl. 149 (1991) 249–272.

[6] G. Poole, L. Neal, Gaussian elimination: when is scaling beneficial?, Linear Algebra Appl. 162–164 (1992) 309–324.

[7] J.R. Rice, Numerical Methods, Software, and Analysis, McGraw-Hill, New York, 1983.

[8] L.N. Trefethen, R.S. Schreiber, Average-case stability of Gaussian elimination, SIAM J. Matrix Anal. Appl. 11 (1990) 335–360.

[9] E. van der Biest, A study of different pivoting strategies and error reducing algorithms for Gaussian elimination, M.S. Thesis, East Tennessee State University, 1991.

[10] D. Wise, Undulant-block elimination and integer-preserving matrix inversion, Sci. Comput. Programming 33 (1999) 29–85.

[11] MATLAB, A software product of Math Works Inc.

Further reading

1. L. Eldén, L. Wittmeyer-Koch, Numerical Analysis, An Introduction, Academic Press, San Diego, CA, 1990.
2. G. Forsythe, M. Malcolm, C.B. Moler, Computer Methods for Mathematical Computations, Prentice-Hall, Englewood Cliffs, NJ, 1977.
3. G.W. Stewart, Introduction to Matrix Computations, Academic Press, New York, 1973.