The growth factor and efficiency of Gaussian elimination with rook pivoting

Leslie V. Foster

Department of Mathematics and Computer Science, San Jose State University, San Jose, CA 95192

Abstract

Gaussian elimination is among the most widely used tools in scientific computing. Gaussian elimination with partial pivoting requires only $O(n^2)$ comparisons beyond the work required in Gaussian elimination with no pivoting but can, in principle, have error growth that is exponential in the matrix size n. Gaussian elimination with complete pivoting on the other hand can not have exponential error growth but requires $O(n^3)$ comparisons beyond the work required by Gaussian elimination with no pivoting. Rook pivoting is a pivoting technique that appears to be intermediate between partial pivoting and complete pivoting in terms of efficiency and accuracy. In this paper we prove that rook pivoting cannot have exponential error growth. We also introduce a combination of partial pivoting and rook pivoting which we call Gaussian elimination with partial rook pivoting and we prove the partial rook pivoting can not have exponential error growth. We include numerical experiments showing that on a serial computer the run times for rook pivoting are almost always close to those of partial pivoting and the run times for partial rook pivoting appear to be the same as those of partial pivoting.

Key words: Gaussian elimination, growth factor, rook pivoting

AMS classification: 65F05, 65G05

1 Introduction

This paper is dedicated to Bill Gragg on his sixtieth birthday.

Gaussian elimination is one of the most widely used algorithms in scientific computing. When applied to an $n \times n$ matrix A Gaussian elimination with complete pivoting, partial pivoting or no pivoting produce factorizations $PAQ = LU$ where P and Q are permutation matrices, L is unit lower triangular, and U is upper triangular. Let $\hat{\boldsymbol{x}}$ represent solution to $A\boldsymbol{x} = \boldsymbol{b}$ computed in floating point arithmetic on a computer with relative machine

precision ϵ. Then it is known [12,20] that

$$\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} \leq \gamma \, \mathrm{cond}_\infty(A) \rho \, \epsilon, \tag{1}$$

where $\boldsymbol{x}$ is the exact solution, $\mathrm{cond}_\infty(A)$ is the condition number of A in the supremum norm, γ is a low degree polynomial in n and is not large, and ρ is the growth factor. Following LAPACK [1], we define the growth factor, for Gaussian elimination with partial pivoting or complete pivoting, by

$$\rho = \frac{\max_{i,j} |u_{i,j}|}{\max_{i,j} |a_{i,j}|}. \tag{2}$$

Due to (1) Gaussian elimination will be numerically stable if ρ is not large.

The classical definition [20] of the growth factor is $\max_{i,j,k} |a_{i,j}^{(k)}| / \max_{i,j} |a_{i,j}|$ where $a_{i,j}^{(k)}$ denotes the i,jth element of the matrix after the kth step of elimination. This definition and the definition (2) are equivalent for complete pivoting and within a factor n of each other for partial pivoting. We will find the definition (2) more convenient.

For partial pivoting, with either the classical definition of the growth factor or the definition (2), it is not difficult to show [8,13,20] that

$$\rho \leq 2^{n-1} \tag{3}$$

and that this bound is attained, for example, for the matrix on page 212 of [20] or matrices in the class presented in [13]. Therefore ρ can grow exponentially in n, by which we mean that $\rho \geq ab^n$ for constants $a > 0$ and $b > 1$. Although such growth is rare in practice, [7] and [21] have presented examples where exponential growth is achieved for matrices arising from commonly used discretizations of integral and differential equations. Therefore in principle and, at times, in practice ρ can be large enough so that partial pivoting is numerically unstable.

For complete pivoting the theoretical bounds are better. In [19] it is shown for complete pivoting that

$$\rho \leq n^{1/2} \left(2 \, 3^{1/2} 4^{1/3} \cdots n^{1/(n-1)}\right)^{1/2} \leq 2\sqrt{n} \, n^{\ln(n)/4}. \tag{4}$$

These functions are relatively slowly growing when compared to the potential exponential growth of ρ for partial pivoting. Furthermore it is known [19] for complete pivoting that these bounds cannot be attained for $n \geq 3$ and no one has been able to find any examples where the growth factor for complete pivoting is bigger than, for example, $2n$ (see [4]). For these reasons complete pivoting is considered to be numerically stable.

The disadvantage with complete pivoting is that it requires approximately $n^3/3$ comparisons, beyond the work required by Gaussian elimination with no pivoting whereas partial pivoting requires approximately $n^2/2$ comparisons. We assume in these counts and throughout the paper that A is a dense matrix. Since Gaussian elimination with no pivoting requires approximately $2n^3/3$ floating point operations, for large n Gaussian elimination with complete pivoting will require substantially more time than Gaussian elimination with no pivoting whereas Gaussian elimination with partial pivoting will require approximately the same time as Gaussian elimination with no pivoting.

Gaussian elimination with rook pivoting, which we define in Section 2, appears to be intermediate between complete pivoting and partial pivoting in terms of efficiency and accuracy. For symmetric matrices, pivoting schemes related to those discussed here have been presented by Fletcher [6] and recently by Ashcraft, Grimes and Lewis [2]. Our focus is on pivoting techniques for nonsymmetric matrices. The first algorithm which we discuss was presented by Neal and Poole in [15]. Their motivation for the method was based on a geometric interpretation of Gaussian elimination. Our motivation for rook pivoting is that its growth factor cannot grow exponentially. In Section 3 we prove that for rook pivoting

$$\rho \leq 1.5\, n^{3\ln(n)/4}. \tag{5}$$

We also show that the growth factor for a combination of rook pivoting and partial pivoting, which we call partial rook pivoting, cannot grow exponentially. In Section 4 we discuss the efficiency of rook pivoting and partial rook pivoting and in Section 5 we describe the results of numerical experiments using modifications of the Fortran code in LAPACK. In Section 6 we present conclusions and, finally, in an appendix we include details of the proof of a lemma used in Section 3.

2 Rook Pivoting and Partial Rook Pivoting Algorithms

Let $A^{(k)}$ be the updated matrix A at the kth step of Gaussian elimination and let $a_{ij}^{(k)}$ be its entries. The only difference between the various Gaussian elimination algorithms is the selection of the pivot element. For Gaussian elimination with rook pivoting we choose, at step k, a pivot element that will be intermediate in size between that of partial pivoting and complete pivoting by searching more of $A^{(k)}$ than partial pivoting and less of than complete pivoting. The algorithm [15] for the selection of the pivot element at step k in rook pivoting is:

Algorithm 1 (*pivot selection for rook pivoting*)

Let $c_0 = k$ and $r_1 =$ row index of a largest magnitude entry on or below row k in column c_0. Let $c_1 =$ the column index of the largest magnitude entry at or right of column k in row r_1. Let continue $= (c_0 \neq c_1)$, $i = 1$, and $j = 0$.

Repeat while continue is true:

If $j = 0$ let r_{i+1} = the row index of a largest magnitude entry on or below row k in column c_i. Let continue = $(r_i \neq r_{i+1})$ and $j = 1$;
else let c_{i+1} = the column index of the largest magnitude entry at or right of column k in row r_{i+1}. Let continue = $(c_i \neq c_{i+1})$, $i = i + 1$, and $j = 0$.

The variable continue must eventually be false, for example after all the elements of the $(n - k + 1) \times (n - k + 1)$ lower right hand corner of $A^{(k)}$ are examined. At this point if $c_i \neq k$ interchange columns c_i and k and if $r_i \neq k$ interchange rows r_i and k.

There may be more than one largest magnitude entry in a row or column. In our implementation if $|a^{(k)}_{r_i c_i}|$ (for $j = 0$) or $|a^{(k)}_{r_{i+1} c_i}|$ (for $j = 1$) is among the largest entries we let $r_{i+1} = r_i$ (for $j = 0$) or $c_{i+1} = c_i$ (for $j = 1$). This will stop the algorithm as quickly as possible in the case of ties.

Note that in implementing the column and row searches one does not need to search elements of any row or column already searched. This can be implemented, as we did in our implementation, with space for storage of two integer vectors of length $n + 1$. The saving in search time is usually minor since, as we will see, rook pivoting typically requires searching very few rows and columns at each step of Gaussian elimination.

Finally we should mention that rook pivoting will produce a factorization $PAQ = LU$ where the largest magnitude entry in each row of U is on the diagonal. Therefore we could write $PAQ = L_1 D L_2^T$ where L_1 and L_2 are both unit lower triangular with all entries one or smaller in magnitude. In this sense the factorization produced by rook pivoting is a generalization of the LDL^T for symmetric matrices.

Gaussian elimination with partial pivoting is numerically stable unless the growth factor is large, as we have noted earlier, and we can monitor the growth factor as partial pivoting proceeds. For example let r_1 and c_1 be defined as in Algorithm 1. For some tolerance TOL, assume that $|a^{(k)}_{r_1 c_1}| \leq \text{TOL} \times \max |a_{ij}|$ at each step of partial pivoting. Then it follows from the definition (2) that for partial pivoting the growth factor $\rho \leq \text{TOL}$. With this in mind we only need to modify partial pivoting if $|a^{(k)}_{r_1 c_1}|$ is large. This is the motivation for our Gaussian elimination with partial rook pivoting algorithm.

Algorithm 2 (pivoting selection for partial rook pivoting)

In the description of Algorithm 1 replace the definition "continue = $(c_0 \neq c_1)$" with "continue = $(c_0 \neq c_1$ and $|a^{(k)}_{r_1 c_1}| > \text{TOL} \times \max |a_{ij}|)$."

A reasonable choice of TOL is n since a growth factor of n for partial pivoting will provide acceptable accuracy in practical cases of interest. If the growth for the partial pivoting portion of the algorithm becomes greater than n, the algorithm switches to steps of rook pivoting. In most cases the growth factor for partial pivoting is less than n [5,18,20] and so usually partial rook pivoting will not require any rook pivoting steps. We should mention that the ideas motivating partial rook pivoting are similar to the ideas in [3] for combining partial pivoting and complete pivoting. However, as we will see, an algorithm using rook

pivoting should be more efficient than an algorithm using complete pivoting.

3 Growth factor bounds

Theorem 1 *For $n \geq 1$ and $1 \leq k \leq n$, let s_k be the positive solution to*

$$s_k(1+s_k)^{k-1} = \frac{k^{k/2}}{(k-1)^{(k-1)/2}} \tag{6}$$

then the growth factor for Gaussian elimination with rook pivoting satisfies

$$\rho \leq s_1(1+s_2)(1+s_3)\cdots(1+s_n) \equiv t_n. \tag{7}$$

For $n \geq 3$ this inequality is a strict inequality.

PROOF. The key tool in our proof will be Hadamard's inequality [14], which is also the tool used by Wilkinson in his derivation [19] of the bound (4) for the growth factor of complete pivoting.

We may assume without loss of generality that the entries in A have already been permuted so that no further pivoting is required in rook pivoting. Then $A = LU$ where L is unit lower triangular and U upper triangular with the property that $|u_{ij}| \leq |u_{ii}|$ for $i, j = 1, 2, \cdots, n$. Assume that the elements of A have been normalized so that $\max_{i,j} |a_{ij}| = 1$. With this assumption if we let $p_i = |u_{ii}|$ for $i = 1, 2, \cdots, n$, we wish to bound the maximum of p_i, $i = 1, \cdots, n$.

Let $\boldsymbol{\ell}_i$ be the ith column of L and $\boldsymbol{u}_i$ be the ith row of U so that $A = \sum_i^n \boldsymbol{\ell}_i \boldsymbol{u}_i$. For some h, $1 \leq h \leq n$, let $I = \{i_k,\, k = 1, \cdots, h\}$ be a subset of $N = \{1, 2, \cdots, n\}$ and let J be the $n - h$ integers in N but not in I. Then $\sum_{i \in I} \boldsymbol{\ell}_i \boldsymbol{u}_i = A - \sum_{j \in J} \boldsymbol{\ell}_j \boldsymbol{u}_j \equiv B$. Let $\hat{L}, \hat{U}$ and $\hat{B}$ be the submatrix with rows and columns in I of, respectively, L, U, and B. It then follows that $\hat{B} = \hat{L}\hat{U}$ where $\hat{L}$ is unit lower triangular, $\hat{U}$ is upper triangular with, for k, $j = 1, \cdots, h$, $|\hat{u}_{kj}| \leq |\hat{u}_{kk}| = p_r$ where r is the kth smallest integer in I . In addition we know that the magnitude of each element in $\hat{B}$ is less than or equal to $1 + \sum_{j \in J} p_j$ since the absolute value of each element in $\boldsymbol{u}_j$ is less than or equal to p_j .

We now apply Hadamard's inequality [14] to $\hat{B}$ to get

$$|\det(\hat{B})| \leq \prod_{k=1}^{h} \left(\|k\text{th column of } \hat{B}\| \right) \leq \left[h^{1/2} \left(1 + \sum_{j \in J} p_j \right) \right]^h . \tag{8}$$

Since $\det(\hat{B}) = \det(\hat{U})$ we may conclude for any subset I of $\{1, 2, \cdots, n\}$ and its complement J that

$$\prod_{i \in I} p_i \leq \left[h^{1/2} \left(1 + \sum_{j \in J} p_j \right) \right]^h . \tag{9}$$

The equation (9) leads to 2^n inequalities, one for each subset of N. We will select n of these for which we can find an explicit formula for the solution, under these constraints, for the maximum of p_i, $i = 1, \cdots, n$. To do this consider the magnitudes of the diagonal entries $|u_{ii}| = p_i$ and order them so that $p_{i_1} \geq p_{i_2} \geq \cdots \geq p_{i_n}$. For $1 \leq h \leq n$ let $I_h = \{i_1, i_2, \cdots, i_h\}$. Then by (9) we have

$$\prod_{k=1}^{h} p_{i_k} \leq \left[h^{1/2} \left(1 + \sum_{k=h+1}^{n} p_{i_k} \right) \right]^h \quad \text{for } h = 1, 2, \cdots, n. \tag{10}$$

$$\text{where } p_{i_1} \geq p_{i_2} \geq \cdots \geq p_{i_n}.$$

To find the maximum of p_i, $i = 1, \cdots, n$ or the maximum of p_{i_1} under the constraints (10) we can renumber the indices, without loss of generality, so that $p_1 \geq p_2 \geq \cdots \geq p_n$ and use the following result whose proof is in the Appendix.

Lemma 2 *The value of the maximum of p_1 subject to the constraints $p_1 \geq p_2 \geq \cdots \geq p_n \geq 0$ and*

$$\prod_{i=1}^{h} p_i \leq \left[h^{1/2} \left(1 + \sum_{i=h+1}^{n} p_i \right) \right]^h \quad \textit{for } h = 1, 2, \cdots, n \tag{11}$$

is achieved only when all the inequalities in (11) are equalities. The value of the maximum is t_n defined in (7).

Since the inequalities (10) are necessary conditions on the magnitudes of the diagonal entries of U the growth factor for Gaussian elimination with rook pivoting is less than or equal to t_n. Since equality in Hadamard's inequality (8) is not achieved for $h = 3$ and other values of h [14], equality in (7) can not be achieved for $n \geq 3$. □

It is easy to solve the equation (6) numerically and so we can calculate the bounds (7). Using (3), (6), (7) and the leftmost bound in (4) we obtain Figure 1 which shows that our bound for the growth factor in rook pivoting is many orders of magnitude less than the bound for partial pivoting and is larger than the bound for complete pivoting.

According to Theorem 1 the bound in (7) cannot be achieved for any $n \geq 3$. Our numerical experiments in Section 5 suggest that the maximum growth factor for rook pivoting is

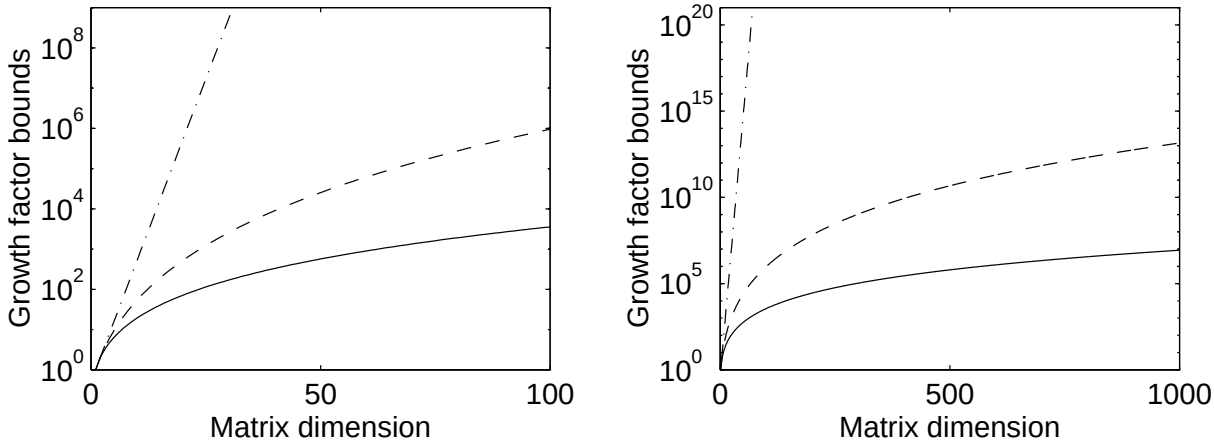

Figure 1. Bounds on the growth factor for complete pivoting (solid), rook pivoting (dashes) and partial pivoting (dash-dots).

much smaller than the bound (7). For complete pivoting the bounds (4) also appear to be much larger than the actual growth factor for any matrix. Indeed the derivation of sharp upper bounds for complete pivoting has been an open question that many mathematicians have worked on [12].

We can use Theorem 1 to obtain an explicit bound for the growth factor for rook pivoting.

Theorem 3 *For $n \geq 1$ the growth factor for Gaussian elimination with rook pivoting satisfies (5).*

PROOF. A direct computation of the solution to (6) and (7) shows that $t_n \leq 1.5n^{3\ln(n)/4}$ for $n < 18$. We may therefore assume that $n \geq 18$.

We will first show that for $n \geq 18$, $s_n < \frac{3\ln n}{2n}$. To do so we will use the result from calculus

$$(1 - nx^2/2)\,\mathrm{e}^{nx} \leq (1+x)^n \leq \mathrm{e}^{nx}\,. \tag{12}$$

Define $p_n(x) = x(1+x)^{n-1} - n^{n/2}/(n-1)^{(n-1)/2}$. It is easy to see that for $x \geq 0$, $p_n(x)$ is an increasing function of x and that $p_n(0) < 0$. Therefore there is a unique positive solution to $p_n(x) = 0$. By (12), $p_n(x) \geq x(1+x)^{-1}(1 - nx^2/2)\,\mathrm{e}^{nx} - n^{1/2}\,\mathrm{e}^{1/2}$ and therefore

$$p_n\left(\frac{3\ln n}{2n}\right) \geq \left[\frac{1.5\ln n}{1 + 1.5\ln(n)/n}\left(1 - \frac{9\ln^2 n}{8n}\right) - \mathrm{e}^{1/2}\right] n^{1/2}. \tag{13}$$

Since, for $n > 0$, $\ln n$ and $n^{1/2}$ are increasing and, for $n > \mathrm{e}^2$, $\ln(n)/n$ and $\ln(n)/n^2$ are decreasing it follows that for $n > \mathrm{e}^2$ the right hand side of (13) is an increasing function of n. A calculation shows that at $n = 18$ this right hand side is positive. Therefore, for $n \geq 18$, $p_n(\frac{3\ln n}{2n}) > 0$. Since $p_n(s_n) = 0$ we may conclude that $s_n < \frac{3\ln n}{2n}$.

From (7) $t_n = s_1 \prod_{k=1}^{n}(1+s_k) = \prod_{k=1}^{n}(1+s_k)$. Therefore

$$\begin{aligned}
\ln(t_n) &= \sum_{k=1}^{18}\ln(1+s_k) + \sum_{k=19}^{n}\ln(1+s_k) \quad \le \sum_{k=1}^{18}\ln(1+s_k) + \sum_{k=19}^{n} s_k \\
&\le \sum_{k=1}^{18}\ln(1+s_k) + \sum_{k=19}^{n}\frac{3\ln k}{2k} \quad \le \sum_{k=1}^{18}\ln(1+s_k) + \int_{k=18}^{n}\frac{3\ln k}{2k}\,\mathrm{d}k \\
&= \sum_{k=1}^{18}\ln(1+s_k) + \frac{3\ln^2(n) - 3\ln^2(18)}{4}
\end{aligned}$$

So, for $n \ge 18$, $t_n \le [\, s_1 \prod_{k=1}^{18}(1+s_k) \,/\, 18^{3\ln(18)/4} \,] \, n^{3\ln(n)/n} = .679\, n^{3\ln(n)/n}$ where the factor .679 comes from a direct calculation. □

It follows from Theorem 3 that, for large n, the growth factor for rook pivoting is more slowly growing than any function of the form ab^n for $a > 0$ and $b > 1$. The following theorem shows that this is also true of partial rook pivoting.

Theorem 4 *For TOL* ≥ 1 *the growth factor for partial rook pivoting satisfies*

$$\rho \le n\,(TOL)\, t_n \le 1.5\, n\,(TOL)\, n^{3\ln(n)/4} \tag{14}$$

PROOF. We may assume without loss of generality that the entries of $A = LU$ have already been permuted so that no further pivoting is required for partial rook pivoting. Also assume that the elements of A have been normalized so that $\max_{ij}|a_{ij}| = 1$. Let J be the set of row indices where, in partial rook pivoting, $|a^{(k)}_{r_1 c_1}| \le \mathrm{TOL} \times \max|a_{ij}|$ and let I be the set of row indices where this is not true. If the number of indices h in I is zero then clearly the theorem is true. Suppose that $h \ge 1$. Using the notation introduced in the proof of Theorem 1, it follows that $\hat{B} = \hat{L}\hat{U}$ where $\hat{L}$ is unit lower triangular, $\hat{U}$ is upper triangular with $|\hat{u}_{ij}| \le |\hat{u}_{ii}|$ for $i,\, j = 1, \cdots, h$ and $\hat{B}$ is the submatrix with rows and columns in I of $B = A - \sum_{j\in J} \boldsymbol{\ell}_j \boldsymbol{u}_j$. The magnitude of each element in $\hat{B}$ is less than or equal to $1 + \sum_{j\in J}\mathrm{TOL} \le n\mathrm{TOL}$ since $1 \le \mathrm{TOL}$ and the absolute value of each element in $\boldsymbol{u}_j$ is less than or equal to TOL. Now $\hat{L}$ and $\hat{U}$ correspond to a rook pivoting factorization of $\hat{B}$ and the result follows from Theorem 1 and Theorem 3. □

4 Efficiency

For rook pivoting to be efficient it is essential that only a few rows and columns of A be searched at each step of Gaussian elimination. The following theorem suggests why this is true. The theorem is a generalization of a result in [16] which requires the more restrictive assumption that the elements of A come from a uniform distribution.

Theorem 5 *At step k of Gaussian elimination if the elements of the $(n-k+1)\times(n-k+1)$ submatrix in the lower right hand corner of $A^{(k)}$ are independent identically distributed random variables from any continuous probability distribution then the expected number of comparisons in step k of rook pivoting is less than or equal to $(n-k)\,\mathrm{e}$.*

PROOF. Let $\gamma = 2(n-k+1)-1$ and let ℓ, $1 \le \ell \le \gamma$, represent the number of rows or columns searched in $A^{(k)}$. Suppose that we search column $c_0 = k$, row r_1, column c_1, row r_2, $\cdots$, row $r_{\ell/2}$ (if ℓ is even) or column $c_{(\ell-1)/2}$ (if ℓ is odd). If $i \le \ell$ is even let x_i equal the maximum magnitude of the elements in row $r_{i/2}$ of $A^{(k)}$ that are left of or on column k and are not in the columns $(c_0, c_2, \cdots, c_{i/2})$ previously searched. If $i \le \ell$ is odd let x_i equal the maximum magnitude of the elements in column $c_{(i-1)/2}$ that are on or below row k and are not in the rows $(r_1, r_2, \cdots, r_{(i-1)/2})$ already searched. Then for $i = 1, 2, \cdots, \ell$ the x_i's form a set of continuous, independent (since the sets of elements of $A^{(k)}$ searched for each x_i are disjoint) random variables. Let $F_i(x)$ be the cumulative probability function for x_i and let $\rho_i(x)$ be the density function of x_i. Since the number of elements searched for each x_i is nonincreasing as i increases and since the elements of $A^{(k)}$ are assumed to independent, identically distributed it follows that for any x, $F_1(x) \le F_2(x) \le \cdots \le F_\ell(x)$. Let P_ℓ be the probability that exactly ℓ rows and columns of $A^{(k)}$ are searched at step k of rook pivoting and let, for $1 \le m \le \gamma$, $P(\ell \ge m)$ be the probability that m or more rows or columns are searched. Now $P(\ell \ge 1) = P(\ell \ge 2) = 1$ and, for $m \ge 2$, the probability that m or more rows or columns are searched is equivalent to $x_1 < x_2 < \cdots < x_{m-1}$. Therefore

$$
\begin{aligned}
P(\ell \ge m) &= \int_0^\infty \int_0^{x_{m-1}} \cdots \int_0^{x_3} \int_0^{x_2} \prod_{i=1}^{m-1} \rho_i(x_i)\, \mathrm{d}x_1\, \mathrm{d}x_2 \cdots \mathrm{d}x_{m-1} \\
&= \int_0^\infty \int_0^{x_{m-1}} \cdots \int_0^{x_3} F_1(x_2) \prod_{i=2}^{m-1} \rho_i(x_i)\, \mathrm{d}x_2 \cdots \mathrm{d}x_{m-1} \\
&\le \int_0^\infty \int_0^{x_{m-1}} \cdots \int_0^{x_3} F_2(x_2) \prod_{i=3}^{m-1} \rho_i(x_i) \left[\rho_2(x_2)\, \mathrm{d}x_2\right] \mathrm{d}x_3 \cdots \mathrm{d}x_{m-1} \\
&= \qquad\qquad \cdots \\
&\le \int_0^\infty \frac{1}{(m-2)!} \left[F_{m-1}(x_{m-1})\right]^{m-2} \rho_{m-1}(x_{m-1})\, \mathrm{d}x_{m-1} \;=\; \frac{1}{(m-1)!}.
\end{aligned}
$$

We may conclude that for $1 \le m \le \gamma$, $\sum_{\ell=m}^{\gamma} P_\ell \le 1/(m-1)!$. Summing and rearranging these equations we get

$$
\sum_{m=1}^{\gamma} \sum_{\ell=m}^{\gamma} P_\ell = \sum_{\ell=1}^{\gamma} \ell P_\ell \le \sum_{m=1}^{\gamma} \frac{1}{(m-1)!} < \mathrm{e}\,.
$$

Therefore the expected value of ℓ, the number of rows and columns searched, is less than e. The theorem follows since, at step k of Gaussian elimination with rook pivoting, one can search ℓ rows and columns, with $\ell(n-k)$ or fewer comparisons. $\square$

If the assumptions of Theorem 5 are true it follows that the expected number of comparisons in a complete factorization by rook pivoting would be less than or equal to $\mathrm{e}\, n(n-1)/2$. If one wishes to compute the growth factor, an additional n^2+n-2 comparisons are needed to calculate $\max_{ij}|a_{ij}|$ and $\max_{ij}|u_{ij}|$ for an approximate total of $(1+\mathrm{e}/2)n^2 \cong 2.36n^2$ or fewer comparisons. We should note that we know of no probability distribution for the elements of the initial matrix A the will imply, after the first step of Gaussian elimination, that the assumptions of Theorem 5 are true . However the theorem suggests that the number of comparisons in factoring a matrix by rook pivoting should be a small multiple of n^2. This is supported by our numerical experiments where rook pivoting required $3.25n^2$ or fewer comparisons on each of the more than 110,000 test matrices described in the next section.

These counts can be compared with the $n(n-1)/2$ comparisons required for basic partial pivoting. If one wishes to calculate the growth factor in partial pivoting an additional $(3n^2+n-4)/2$ comparisons or a total of $2n^2-2$ comparisons are required. Partial rook pivoting, except in the case where partial pivoting has a growth factor bigger that TOL, will also require $2n^2-2$ comparisons plus an additional n comparisons to test if rook pivoting should be implemented at any step. Approximately $n^3/3$ comparisons are required by complete pivoting. All of these algorithms, except for Gaussian elimination with no pivoting, require interchanging elements of A. The number of such interchanges is $\leq 2n^2$ for complete pivoting, rook pivoting and partial rook pivoting and $\leq n^2$ for partial pivoting. Recall that the number of floating point operations required by the elimination steps in each version of Gaussian elimination is approximately $2n^3/3$.

Finally, in this section we should note that rook pivoting can require $O(n^3)$ comparisons as is illustrated by, for $c>1$, the matrix whose diagonal has entries $c, c^3, c^5, \cdots, c^{2n-1}$, whose superdiagonal has entries $c^2, c^4, \cdots, c^{2n-2}$ and otherwise the matrix is zero. Rook pivoting requires approximately $n^3/4$ comparisons to factor this matrix. Although examples where rook pivoting requires $O(n^3)$ comparisons exist, as we see in the next section they appear to be very rare.

5 Numerical experiments

In this section we compare via numerical experiments the growth factor and the efficiency of Gaussian elimination with partial pivoting, complete pivoting, rook pivoting, partial rook pivoting and no pivoting.

We modified the all Fortran implementation of LAPACK [1]. For example to implement partial pivoting we changed LAPACK's utility DGESVX by omitting the portions of the code that do condition estimation and iterative refinement. We tested two variations of the code for partial pivoting. In one we kept DGESVX's calculation of the growth factor (we label this gepp+gf in our tables) and in the other version we omitted this calculation (we label this gepp). For the remaining algorithms we modified DGESVX to

implement complete pivoting (gecp), rook pivoting (gerp), partial rook pivoting (geprp) and no pivoting (genp). We made as few changes in DGESVX as we could and still have efficient implementations of these algorithms. To make the timings in our test runs consistent, we deleted the feature of LAPACK that tests whether a multiplier is zero.

Our runs were done on a SUN 4 computer using Fortran version 2.0 with optimization level 3. This is a serial computer environment and we let LAPACK's block size be one. Choosing a larger block size did not decrease the run times in this environment.

Our runs report five quantities: the average computer time required to solve $A\boldsymbol{x} = \boldsymbol{b}$, the average size of the growth factor, the average number of comparisons required by the algorithm, the maximum growth factor for the samples in a set of matrices, and the maximum number of comparisons for any sample in the set. We did our runs when the computer was otherwise idle and for each run we carried out enough repetitions so that the reported computer times were consistent within a few percent for repeated runs. For Gaussian elimination with no pivoting the size of the elements of L should be included in the growth factor [8,12] and ρ as defined in (2) is only a lower bound on the growth factor. For simplicity we report the ρ of equation (2) for all of our test runs.

In Tables 1 through 3 we report test runs for 100,000 50 by 50 matrices with elements chosen from a uniform distribution between -1 and 1, 10,000 100 by 100 matrices from this class, and 100 500 by 500 such matrices. In Table 4 we report tests for 100 100 by 100 matrices of the form UDV^T where U and V are random orthogonal matrices from the Haar distribution [17] and D is a diagonal matrix with its diagonal entries chosen from a variety of sequences (one small entry, one large entry, uniformly decreasing entries and uniformly logarithmically decreasing entries) such that $\text{cond}(A) = 6.7 \times 10^7$. Table 5 reports test runs for 64 100 by 100 matrices from the collections of Higham [10,11] and of Hansen [9]. Most of these matrices are not random and some are very ill conditioned. We included all the examples generated by Higham's "matrix" function except for matrices that are identified as exactly singular by Gaussian elimination with partial pivoting and except for Higham's example where partial pivoting has a large growth factor. We included all of the examples in Hansen's set of examples relating to inverse problems. For Table 6 we collected together thirty examples of matrices where partial pivoting has large growth factors. These examples include the example of Wilkinson [20], matrices from the class of examples of Higham and Higham [13], the practical examples reported by Foster [7] and those reported by Wright [21]. In these examples partial pivoting had a growth factor of 6.9×10^7 or, in some cases, much larger.

These tables do not report information about the numbers of elements of A interchanged as rows or columns are switched. The run for 10,000 100 by 100 matrices on average interchanged $.96n^2$ elements for partial pivoting and partial rook pivoting, $1.59n^2$ elements for rook pivoting and $1.91n^2$ elements for complete pivoting. These numbers are typical of our other tests. Rook pivoting requires slightly more (relative to the $O(n^3)$ overall operations in Gaussian elimination) interchanges than does partial pivoting or partial rook pivoting and requires fewer interchanges than complete pivoting.

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	.0106	.0121	.0143	.0142	.0156	.0344
ave. ρ	3.8×10^{11}	7.2	7.2	7.2	4.8	3.8
ave. compares / n^2	0	.49	2.00	2.02	2.48	18.2
max. ρ	3.5×10^{16}	23.0	23.0	23.0	10.5	5.6
max. compares / n^2	0	.49	2.00	2.02	2.86	18.2

Table 1. Results for 100,000 50 by 50 matrices with elements from a uniform distribution.

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	.096	.107	.116	.115	.121	.296
ave. ρ	6574	11.7	11.7	11.7	7.3	5.5
ave. compares / n^2	0	.50	2.00	2.01	2.54	34.8
max. ρ	7.9×10^6	33.6	33.6	33.6	13.9	7.6
max. compares / n^2	0	.50	2.00	2.01	2.78	34.8

Table 2. Results for 10,000 100 by 100 matrices with elements from a uniform distribution.

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	11.4	11.9	12.1	12.2	12.4	34.8
ave. ρ	28530	32.5	32.5	32.5	19.0	14.0
ave. compares / n^2	0	.50	2.00	2.00	2.67	168.2
max. ρ	6.6×10^5	43.7	43.7	43.7	26.8	16.8
max. compares / n^2	0	.50	2.00	2.00	2.75	168.2

Table 3. Results for 100 500 by 500 matrices with elements from a uniform distribution.

These numerical results support our theoretical remarks about rook pivoting. The growth factor for rook pivoting is acceptable in all our tests including the examples where partial pivoting has very large growth factors. The largest growth factor for rook pivoting in these more than 110,000 test matrices was 50.5. This occurred for a matrix presented in [13] for which the growth factor is approximately $n/2$ for complete pivoting and rook pivoting. Our numerical results also show that rook pivoting requires somewhat more comparisons

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	.095	.107	.115	.114	.121	.295
ave. ρ	998	4.5	4.5	4.5	3.3	2.6
ave. compares / n^2	0	.50	2.00	2.01	2.59	34.8
max. ρ	25930	23.2	23.2	23.2	15.2	9.1
max. compares / n^2	0	.50	2.00	2.01	2.88	34.8

Table 4. Results for 100 100 by 100 matrices of the form UDV^T where U and V are random orthogonal matrices and the singular values of A are chosen from various distributions with cond(A) = 6.7×10^7.

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	.095	.103	.111	.111	.114	.292
ave. ρ	∞	2.85	2.85	2.85	2.36	2.32
ave. compares / n^2	0	.50	2.00	2.01	2.33	34.8
max. ρ	∞	56.1	56.1	56.1	50.5	50.7
max. compares / n^2	0	.50	2.00	2.01	3.25	34.8

Table 5. Results for 64 100 by 100 matrices in the collection of Higham [10,11] and in the collection of Hansen [9]. Most of these matrices are not random.

	genp	gepp	gepp+gf	geprp	gerp	gecp
ave. time (sec)	.095	.098	.107	.109	.112	.292
ave. ρ	∞	3.5×10^{28}	3.5×10^{28}	147.6	1.36	1.27
ave. compares / n^2	0	.50	2.00	2.03	2.26	34.8
max. ρ	∞	6.3×10^{29}	6.3×10^{29}	251.6	5.00	2.00
max. compares / n^2	0	.50	2.00	2.08	2.50	34.8

Table 6. Results for 30 100 by 100 matrices from [7,13,20,21]. For all of these examples Gaussian elimination partial pivoting has a growth factor larger than 6.9×10^7.

(at most 65 percent more in these tests) than a partial pivoting algorithm that calculates the growth factor. This leads to a much smaller increase (from 3 to 10 percent) in the overall computation time of rook pivoting. In summary, the run times for rook pivoting are close to those of basic partial pivoting and are very close to the run times of a partial pivoting algorithm that calculates the growth factor.

The results also indicate that partial rook pivoting successfully switches to rook pivoting in cases where partial pivoting has disastrous error growth. The growth factor for partial rook pivoting was always 252 or smaller for matrices where partial pivoting had a growth factor as large as 6×10^{29}. A growth factor of 252 is not significant since, for example, one step of iterative refinement will eliminate the error introduced by such a growth factor. As noted in [7], for partial pivoting the growth factor can be large enough so that iterative refinement will not improve the calculated solution. Within the uncertainty of a few percent in our timing routines, the computer times for partial rook pivoting are identical to those of a partial pivoting algorithm which calculates the growth factor. The number of comparisons required by the two algorithms are close enough so that there is no significant effect on the computer run times.

Our results also support the view that complete pivoting is substantially slower than partial pivoting and that Gaussian elimination with no pivoting often has unacceptable error growth.

We should note that we tested the example mentioned at the end of Section 4 where rook pivoting required $O(n^3)$ comparisons. For $n = 100$ the two versions of partial pivoting required .108 and .116 seconds, partial rook pivoting required .115 seconds, rook pivoting required .269 seconds and complete pivoting required .295 seconds. Although examples exist where partial rook pivoting is slow, they did not show up in test runs on more that 110,000 random matrices or matrices in the published collections that we tested.

Our tests and the tests in [15,16] support each other. Both sets of tests appear to show that for random matrices the number of comparisons in rook pivoting is not large and that rook pivoting calculates accurate answers when this is possible. The tests we have reported here include results on growth factors, results on computer run times for Fortran code, results for partial rook pivoting, tests for the nonrandom matrices in the standard collections of Higham and Hansen and tests for matrices where partial pivoting has large growth factors. Tests such as these are not included in the experiments in [15,16].

6 Conclusions and further work

Gaussian elimination with no pivoting, partial pivoting and complete pivoting can be put on continuums of increasing work and decreasing bounds on the error in the calculated solution. Rook pivoting and partial rook pivoting are close to partial pivoting in efficiency and closer to complete pivoting in terms of the size of the error bounds. Rook pivoting

almost always requires only a small amount of additional computer time beyond that of partial pivoting and, in our test runs, partial rook pivoting required the same amount of computer time as a partial pivoting algorithm that calculates the growth factor. Theorems 1, 3 and 4 prove that rook pivoting and partial rook pivoting can not have the exponential error growth that can, at times, make partial pivoting numerically unstable.

There are several interesting extensions to this work. Our implementation of rook pivoting was tested on a serial computer. It would be of interest to explore effective implementations on high performance computers. Also it appears that rook pivoting produces rank revealing factorizations and it is of interest to explore this further. Development of sharp bounds for the growth factor in rook pivoting is a third area of interest.

A Appendix. Proof of Lemma 2

It is not hard to show that if all the inequalities in (11) are equalities then $p_1 = t_n$. However Lemma 2 is not true (p_1 is unbounded) for $n \geq 3$ without the additional inequalities $p_1 \geq p_2 \geq \cdots \geq p_n \geq 0$ and we are not able to prove Lemma 2 by directly manipulating the inequalities in (11). In our proof Lemma 2 follows from Lemma A.2, below, by letting $k = n$, $p_{k+1} = 0$ and $C = n^{n/2}$. Lemmas A.1 and A.3 are used in the proof of Lemma A.2.

Lemma A.1 *For $k \geq 1$, let s_k be defined by (6). Then*

$$1 = s_1 = s_2 > s_3 > s_4 > \cdots \tag{A.1}$$

PROOF. Note that $s_1 = s_2 = 1$ is trivial. Suppose that $k \geq 2$ and let $f(s) = (k-1)^{(k-1)/2} s(1+s)^{k-1} - k^{k/2}$. Then $f(\sqrt{k}) > 0$ follows easily and since $f(0) < 0$, we may conclude that $s_k < \sqrt{k}$. For this reason and due to (6) it follows that $1 + s_k > \left(\frac{k}{k-1}\right)^{1/2}$ for $k \geq 2$. Let s_k' be the derivative of s_k with respect of k. Taking logarithms of (6) and differentiating with respect to k it follows that $s_k' \left(\frac{1}{s_k} + \frac{k-1}{1+s_k}\right) = \ln\left(\frac{k}{k-1}\right)^{1/2} - \ln(1+s_k) < 0$. Therefore s_k is decreasing for $k \geq 2$, proving the result. □

Lemma A.2 *Given $k \geq 1$, $0 < C \leq k^{k/2}$ and a fixed $p_{k+1} \geq 0$ consider the problem of maximizing p_1 subject to the constraints*

$$p_1 \geq p_2 \geq \cdots \geq p_k \geq p_{k+1}, \tag{A.2}$$

$$\prod_{i=1}^{k} p_i \leq C \text{ and} \tag{A.3}$$

$$\prod_{i=1}^{h} p_i \leq \left[h^{1/2} \left(1 + \sum_{i=h+1}^{k} p_i \right) \right]^h \text{ for } h = 1, 2, \cdots, k-1. \tag{A.4}$$

Let $q_{k+1} = 0$ and, for $r = 1, \cdots, k$, let $q_{k-r+1} > 0$ satisfy

$$(k-r)^{(k-r)/2}\,(q_{k-r+1})^r(1+rq_{k-r+1})^{k-r} = C. \tag{A.5}$$

We will show in Lemma A.3 that $q_{k+1} < q_k < \cdots < q_1$. The maximum value of p_1 depends on p_{k+1}. If for some r, $0 \le r \le k-2$, $q_{k-r+1} < p_{k+1} \le q_{k-r}$, then find $p_{k-r} \ge p_{k+1}$ such that

$$(k-r-1)^{(k-r-1)/2}\,p_{k-r}(1+rp_{k+1}+p_{k-r})^{k-r-1} = C/(p_{k+1})^r \text{ and let} \tag{A.6}$$

$$p_1 = s_1(1+s_2)\cdots(1+s_{k-r-1})(1+rp_{k+1}+p_{k-r}); \tag{A.7}$$

if $q_2 < p_{k+1} \le q_1$ then

$$p_1 = C/(p_{k+1})^{k-1}; \tag{A.8}$$

and if $q_1 < p_{k+1}$ there is no feasible solution to (A.2) - (A.4). In the case that $p_{k+1} \le q_k$ the maximum p_1 is achieved only when all the inequalities in (A.3) and (A.4) are equalities.

PROOF. For $k = 1$ the theorem is clearly true. Now assume that the lemma is true for $k = n$. We will show that it follows that it is true for $k = n+1$. In our notation below terms with hats (e.g. $\hat{p}_1$) will represent variables for $k = n$ and terms without hats (e.g. p_1) will represent the terms that correspond to $k = n+1$. As we will see, by using the induction hypothesis we can find, for each fixed $p_{n+1} \ge p_{n+2} \ge 0$, the maximum of p_1 subject to (A.2)-(A.4) with $k = n+1$. This will lead to a solution for any $p_{n+1} \ge p_{n+2}$. We consider two cases in our solution: $C \ge n^{n/2}p_{n+1}(1+p_{n+1})^n$ and $C < n^{n/2}p_{n+1}(1+p_{n+1})^n$.

In the case that $C \ge n^{n/2}p_{n+1}(1+p_{n+1})^n$ equations (A.2)-(A.4) for $k = n+1$ are equivalent to (A.2) and (A.4). Let us define $\hat{p}_i = p_i/(1+p_{n+1})$ for $i = 1, \cdots, n+1$. Then by algebra it follows that (A.2), (A.3) and (A.4) are true for $k = n$ with all the p_i's replace by $\hat{p}_i$ and with C replaced by $n^{n/2}$. Furthermore in this case $p_{n+1}(1+p_{n+1})^n \le C/n^{n/2} \le (n+1)^{(n+1)/2}/n^{n/2}$. Since $x(1+x)^n$ is an increasing function of x and due to (6) it follows that $p_{n+1} \le s_{n+1}$ and therefore by Lemma A.1 that $\hat{p}_{n+1} \le s_{n+1}/(1+s_{n+1}) \le s_{n+1} \le s_n$. However by (A.5) and (6) $\hat{q}_n$, the solution to (A.5) with $k = n$, $r = 1$ and C replaced by $n^{n/2}$, is just s_n. This implies that $0 < \hat{p}_{n+1} \le \hat{q}_n$ and we can apply the induction hypothesis to conclude, by (A.6) and (A.7) with $r = 0$, that the maximum of $\hat{p}_1$ is achieved for $\hat{p}_1 = s_1(1+s_2)\cdots(1+s_{n-1})(1+\hat{p}_n) = s_1(1+s_2)\cdots(1+s_{n-1})(1+\hat{q}_n) = s_1(1+s_2)\cdots(1+s_{n-1})(1+s_n)$. Therefore, for fixed p_{n+1} such that $n^{n/2}p_{n+1}(1+p_{n+1})^n \le C$

$$p_1 = s_1(1+s_2)\cdots(1+s_{n-1})(1+s_n)(1+p_{n+1}). \tag{A.9}$$

In the case that $n^{n/2}\,p_{n+1}(1+p_{n+1})^n > C$ equations (A.2)-(A.4) for $k = n+1$ are equivalent to (A.2), (A.3) and, for $h = 1, 2, \cdots, n-1$, the equations in (A.4). Again we

define $\hat{p}_i = p_i/(1+p_{n+1})$ for $i = 1, \cdots, n+1$. Then by algebra it follows that (A.2), (A.3) and (A.4) are true for $k = n$ with all the p_i's replace by $\hat{p}_i$ and with C replaced by $\hat{C} \equiv C \,/\, [p_{n+1}(1+p_{n+1})^n] \leq n^{n/2}$.

The solutions $q_1, q_2, \cdots, q_{n+1}$ to (A.5) with $k = n+1$ are also the solutions (replace r by $r+1$) to

$$(n-r)^{(n-r)/2}\,(q_{n-r+1})^{r+1}(1+(r+1)q_{n-r+1})^{n-r} = C \text{ for } r = 0, 1, \cdots, n. \tag{A.10}$$

Define $\hat{q}_{n+1} = 0$ and $\hat{q}_{n-r+1}$, $r = 1, 2, \cdots, n$, to be the solutions to

$$(n-r)^{(n-r)/2}\,(\hat{q}_{n-r+1})^{r}(1+r\hat{q}_{n-r+1})^{n-r} = C \,/\, [p_{n+1}(1+p_{n+1})^n] = \hat{C}. \tag{A.11}$$

To facilitate our induction argument we will show that for $r = 1, 2, \cdots, n$,

$$p_{n+1} \leq q_{n-r+1} \text{ if and only if } \hat{p}_{n+1} \leq \hat{q}_{n-r+1}. \tag{A.12}$$

To do this assume for some r, $1 \leq r \leq n$, that $0 < p_{n+1} \leq q_{n-r+1}$. Define $\tilde{q}_{n-r+1} = q_{n-r+1}/(1+q_{n-r+1})$ and recall that $\hat{p}_{n+1} = p_{n+1}/(1+p_{n+1})$. Since $x/(x+1)$ is increasing for $x \geq 0$, $p_{n+1} \leq q_{n-r+1}$ implies $\hat{p}_{n+1} \leq \tilde{q}_{n-r+1}$. By (A.10) and since $x(1+x)^n$ is increasing for $x \geq 0$ it follows that

$$(n-r)^{(n-r)/2}\,(\tilde{q}_{n-r+1})^{r}(1+r\tilde{q}_{n-r+1})^{n-r} = C \,/\, [q_{n-r+1}(1+q_{n-r+1})^n] \leq \hat{C}. \tag{A.13}$$

Since $x^r(1+rx)^{n-r}$ is increasing for $x \geq 0$, (A.11) and (A.13) imply that $\tilde{q}_{n-r+1} \leq \hat{q}_{n-r+1}$ and we may conclude that $\hat{p}_{n+1} \leq \hat{q}_{n-r+1}$. Now all the $\leq$ inequalities in the inequalities from equation (A.10) through the conclusion that $\hat{p}_{n+1} \leq \hat{q}_{n-r+1}$ can be replaced, using identical arguments, with $>$ and this proves (A.12).

Continuing the case where $n^{n/2}p_{n+1}(1+p_{n+1})^n > C$ we may use the induction hypothesis. If for some r, $0 \leq r \leq n-2$, $q_{n-r+1} < p_{n+1} \leq q_{n-r}$ then by (A.12) $\hat{q}_{n-r+1} < \hat{p}_{n+1} \leq \hat{q}_{n-r}$ and consequently equations (A.6) and (A.7) are true with $k = n$ and p_1, p_{n-r}, p_{n+1} and C replaced, respectively, by $\hat{p}_1$, $\hat{p}_{n-r}$, $\hat{p}_{n+1}$ and $\hat{C}$. Also by the induction hypothesis $\hat{p}_{n-r} \geq \hat{p}_{n+1}$ solving (A.6) exists. Therefore by algebra $p_{n-r} \geq p_{n+1}$ where

$$(n-r-1)^{(n-r-1)/2}\,p_{n-r}[1+(r+1)p_{n+1}+p_{n-r}]^{n-r-1} = C/(p_{n+1})^{r+1} \text{ and} \tag{A.14}$$

$$p_1 = s_1(1+s_2)\cdots(1+s_{n-r-1})(1+(r+1)p_{n+1}+p_{n-r}). \tag{A.15}$$

If $q_2 < p_{n+1} \leq q_1$ then $\hat{q}_2 < \hat{p}_{n+1} \leq \hat{q}_1$ and by (A.8) $\hat{p}_1 = \hat{C}/(\hat{p}_{n+1})^{n-1}$ which implies that

$$p_1 = C/(p_{n+1})^n. \tag{A.16}$$

Finally if $q_1 < p_{n+1}$ then $\hat{q}_1 < \hat{p}_{n+1}$ and by the induction hypothesis there is no feasible solution to the constraints.

By use of the induction hypothesis we have reduced the $n+1$ dimensional problem of finding the maximum of p_1 subject to (A.2)-(A.4) with $k = n+1$ to a problem involving one variable p_{n+1} where p_1 as a function of p_{n+1} is given, for $n^{n/2} p_{n+1}(1+p_{n+1})^n \leq C$, by (A.9) and, for $n^{n/2} p_{n+1}(1+p_{n+1})^n > C$, by (A.14) through (A.16).

Note that it follows easily that p_1 given by (A.9) is an increasing function of p_{n+1}. We now show that p_1 given by (A.14) through (A.16) is a non-increasing function of p_{n+1}. To do so note that p_1 given by (A.16) is clearly decreasing. For some r, $0 \leq r \leq n-2$ assume that (A.14)-(A.15) apply. Let p_1' and p_{n-r}' be the derivatives of, respectively, p_1 and p_{n-r} with respect to p_{n+1}. Implicitly differentiating (A.15) and equation (A.14) multiplied by $(p_{n+1})^{r+1}$ we obtain $p_1' = s_1(1+s_2)\cdots(1+s_{n-r-1})[(r+1)+p_{n-r}']$ and

$$\begin{aligned}
&[(r+1)+p_{n-r}'](p_{n+1})^{r+1}[1+(r+1)p_{n+1}+p_{n-r}]^{n-r-2}[(n-r)p_{n-r}+(r+1)p_{n+1}+1] \\
&\quad = \; -(r+1)(p_{n+1})^{r}[1+(r+1)p_{n+1}+p_{n-r}]^{n-r-1}[p_{n-r}-p_{n+1}]. \qquad \text{(A.17)}
\end{aligned}$$

Since $p_{n-r} \geq p_{n+1}$ we may conclude that $p_1' \leq 0$ or that p_1 is non-increasing. Note by (A.10) that q_{n+1} is the solution to $n^{n/2} q_{n+1}(1+q_{n+1})^n = C$. It follows easily from Lemma A.3 that if $p_{n+1} = q_{n+1}$ then $p_n \neq p_{n+1}$. Therefore for p_{n+1} slightly larger than q_{n+1}, $p_n > p_{n+1}$ and by (A.17) with $r = 0$, p_1 is strictly decreasing.

To finish the proof of the lemma for a given $p_{n+2} > 0$ we need to look at (A.2)-(A.4) with $k = n+1$ for any $p_{n+1} \geq p_{n+2}$. First we consider the case that $p_{n+2} \leq q_{n+1}$. By (A.9) p_1 is an increasing function of p_{n+1} for $p_{n+2} \leq p_{n+1} < q_{n+1}$, by the remarks in the last paragraph p_1 is decreasing for p_{n+1} slightly bigger than q_{n+1} and for p_{n+1} larger, p_1 is not increasing. Therefore the maximum of p_1 is achieved for $p_{n+1} = q_{n+1}$ and this value of p_{n+1} is unique. Letting $p_{n+1} = q_{n+1}$ in (A.9) it follows that for $r = 0$ and $k = n+1$ (A.6) and (A.7) are true and that $p_{k-r} \geq p_{k+1}$. In this case in order to achieve the maximum p_1 the value of p_{n+1} can only be q_{n+1}. Using this and an induction argument, one can easily show that all the inequalities in (A.3) and (A.4) are equalities.

In the case that $p_{n+2} > q_{n+1}$ we know that p_1 is a non-increasing function of $p_{n+1} \geq p_{n+2}$. Therefore, in this case we can set $p_{n+1} = p_{n+2}$. With this value of p_{n+1} (A.14)-(A.16) are just (A.6)-(A.8) with $k = n+1$ and r increased by 1. Also we know that $p_{n-r} \geq p_{n+1}$ and therefore $p_{n-r} \geq p_{n+2}$. This completes the induction proof. □

Lemma A.3 *If $k \geq 2$, $0 < C \leq k^{k/2}$ and $q_1, q_2, \cdots, q_k$ satisfy (A.5) then*

$$q_k < q_{k-1} < \cdots < q_1. \qquad \text{(A.18)}$$

PROOF. We begin by showing that $q_k < q_{k-1}$. Let $f(x) = (k-1)^{(k-1)/2} x(1+x)^{k-1} - C$ and $g(x) = (k-2)^{(k-2)/2} x^2(1+2x)^{k-2} - C$. Then q_k is the positive solution to $f(x) = 0$

and q_{k+1} is the positive solution to $g(x) = 0$. Now $f(0) = g(0) = -C < 0$, $f'(0) > 0$ and $g'(0) = 0$. Therefore for small $x > 0$, $g(x) < f(x)$. Suppose that for some $x > 0$ that $f(x) = g(x)$. Then $(k-1)^{(k-1)/2}\, x(1+x)^{k-1} = (k-2)^{(k-2)/2}\, x^2(1+2x)^{k-2}$. Dividing by $(1+x)^{k-1}$ and defining $\hat{x} = x/(1+x)$ then $(k-1)^{(k-1)/2} = (k-2)^{(k-2)/2}\, \hat{x}(1+\hat{x})^{k-2}$ and it follows from (6) that $\hat{x} = s_{k-1}$. Therefore $g(x) < f(x)$ for $0 < x < s_{k-1}/(1-s_{k-1})$. Now the condition $C \le k^{k/2}$, (6) and (A.5) imply that $q_k \le s_k$ and by Lemma A.1 $s_k < s_{k-1} < s_{k-1}/(1-s_{k-1})$. Therefore $g(x) < f(x)$ for $0 < x \le q_k$. Since $f(q_k) = 0$ it follows that the zero q_{k-1} of $g(x)$ satisfies $q_k < q_{k-1}$. This establishes the base steps for the induction arguments below.

We now show (A.18) by using induction on k. We assume that (A.18) is true for $k = n$ and show that this implies (A.18) for $k = n+1$. As in our earlier notation we will use hats over variables to refer to the case that $k = n$ and variable without hats will refer to the case that $k = n+1$. We also will use induction on r and will assume for some r, $1 \le r \le n-1$, that

$$q_{n+1} < q_n < \cdots < q_{n-r+1}. \tag{A.19}$$

We will show that $q_{n-r+1} < q_{n-r}$ where (by (A.5) with $k = n+1$ and r increased by 1)

$$(n-r)^{(n-r)/2}\, (q_{n-r+1})^{r+1}[1 + (r+1)q_{n-r+1}]^{n-r} = C \tag{A.20}$$

and

$$(n-r-1)^{(n-r-1)/2}\, (q_{n-r})^{r+2}[1 + (r+2)q_{n-r}]^{n-r-1} = C \tag{A.21}$$

with $C \le (n+1)^{(n+1)/2}$. Now let $\hat{q}_{n-r+1} = q_{n-r+1}/(1+q_{n-r+1})$. Then dividing (A.20) by $(1+q_{n-r+1})^n$ we get

$$(n-r)^{(n-r)/2}\, (\hat{q}_{n-r+1})^{r}[1 + r\hat{q}_{n-r+1}]^{n-r} = \frac{C}{q_{n-r+1}(1+q_{n-r+1})^n} \equiv \hat{C}. \tag{A.22}$$

Now define $h(x) = C/[x(1+x)^n]$ and define $\hat{q}_{n-r}$ by

$$(n-r-1)^{(n-r-1)/2}\, (\hat{q}_{n-r})^{r+1}[1 + (r+1)\hat{q}_{n-r}]^{n-r-1} = \hat{C} = h(q_{n-r+1}). \tag{A.23}$$

By the induction hypothesis (A.19) in r we know that $q_{n+1} < q_{n-r+1}$ where by (A.5) $n^{n/2} q_{n+1}(1+q_{n+1})^n = C$. Since $h(x)$ is decreasing it follows that $\hat{C} = h(q_{n-r+1}) \le h(q_{n+1}) = n^{n/2}$. We can now apply the induction hypothesis (in n) to conclude that

$$\hat{q}_{n-r+1} < \hat{q}_{n-r}. \tag{A.24}$$

To finish the proof let us define $\tilde{q}_{n-r} = q_{n-r}/(1+q_{n-r})$. By (A.21) it follows that

$$(n-r-1)^{(n-r-1)/2}\,(\tilde{q}_{n-r})^{r+1}[1+(r+1)\tilde{q}_{n-r}]^{n-r-1} = \frac{C}{q_{n-r}(1+q_{n-r})^n}. \qquad \text{(A.25)}$$

Suppose that $\tilde{q}_{n-r} < \hat{q}_{n-r}$, then by (A.23) and (A.25) it follows that $C/[q_{n-r}(1+q_{n-r})^n] = h(q_{n-r}) < \hat{C} = h(q_{n-r+1})$ and therefore $q_{n-r+1} < q_{n-r}$ which is the desired conclusion. On the other hand suppose that $\hat{q}_{n-r} \le \tilde{q}_{n-r}$. Then (A.24) implies that $\hat{q}_{n-r+1} < \tilde{q}_{n-r}$. Therefore by the definitions $\hat{q}_{n-r+1} = q_{n-r+1}/(1+q_{n-r+1})$ and $\tilde{q}_{n-r} = q_{n-r}/(1+q_{n-r})$ we may conclude that $q_{n-r+1} < q_{n-r}$. Again we get the desired conclusion which completes the induction proof. □

References

[1] E. Anderson, Z. Bai, C. H. Bischof, J. J. Dongarra J. W. Demmel, J. J. Du Croz, S. J. Hammarling A. Greenbaum, A. McKenney, S. Ostrouchov, and D. C. Sorensen. *LAPACK Users Guide, Release 2.0.* SIAM, Philadelphia, 1995.

[2] Cleve Ashcraft, Roger G. Grimes, and John G. Lewis. Accurate symmetric indefinite linear equation solvers. To appear in *SIAM J. Matrix. Anal. Appl.*, 1998.

[3] P. A. Businger. Monitoring the numerical stability of Gaussian elimination. *Numer. Math.*, 16:360–361, 1971.

[4] Alan Edelman. The complete pivoting conjecture for Gaussian elimination is false. *The Mathematica Journal*, 2:58–61, 1992.

[5] Alan Edelman and Walter Mascarenhas. On the complete pivoting conjecture for a Hadamard matrix of order 12. *Linear and Multilinear Algebra*, 38:181–185, 1995.

[6] Roger Fletcher. Factorizing symmetric indefinite matrices. *Linear Algebra and its Applications*, 14:257–272, 1976.

[7] Leslie V. Foster. Gaussian elimination with partial pivoting can fail in practice. *SIAM J. Matrix Anal. App.*, 15:1354–1362, 1994.

[8] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, 1989.

[9] Per Christian Hansen. Regularization tools. Technical Report UNIC-92-03, Danish Computing Center for Research and Education, Technical University of Denmark, 1995.

[10] Nicholas J. Higham. Algorithm 694: A collection of test matrices in Matlab. *ACM Trans. Math. Software*, 17:289–395, 1991.

[11] Nicholas J. Higham. The test matrix toolbox for Matlab, version 3.0. Technical Report for Numerical Analysis, No. 276, University of Manchester, Manchester, England, 1995.

[12] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 1996.

[13] Nicholas J. Higham and Desmond J. Higham. Large growth factors in Gaussian elimination with pivoting. *SIAM J. Matrix. Anal. Appl.*, 10:155–164, 1989.

[14] Roger A. Horn and Charles B. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 1985.

[15] Larry Neal and George Poole. A geometric analysis of Gaussian elimination, II. *Linear Algebra and its Applications*, 173:39–264, 1992.

[16] Larry Neal and George Poole, Mathematics Department, East Tennessee State University. The rook's pivoting strategy. Private communication, 1996.

[17] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17:403–409, 1980.

[18] Lloyd N. Trefethen and Robert J. Schreiber. Average case stability of Gaussian elimination. *SIAM J. Matrix. Anal. Appl.*, 11:335–360, 1990.

[19] J. H. Wilkinson. Error analysis of direct methods for matrix inversion. *J. Soc. Indust. Appl. Math.*, 10:162–195, 1962.

[20] J. H. Wilkinson. *The Algebraic EIgenvalue Problem.* Oxford University Press, London, 1965.

[21] Stephen J. Wright. A collection of problems for which Gaussian elimination with partial pivoting is unstable. *SIAM Journal on Scientific Computing*, 14:231–238, 1993.