A NEW UPPER BOUND FOR THE GROWTH FACTOR IN GAUSSIAN ELIMINATION WITH COMPLETE PIVOTING

ANKIT BISAIN, ALAN EDELMAN, AND JOHN URSCHEL

Abstract. The growth factor in Gaussian elimination measures how large the entries of an LU factorization can be relative to the entries of the original matrix. It is a key parameter in error estimates, and one of the most fundamental topics in numerical analysis. We produce an upper bound of $n^{0.2079 \ln n + 0.91}$ for the growth factor in Gaussian elimination with complete pivoting – the first improvement upon Wilkinson's original 1961 bound of $2\, n^{0.25 \ln n + 0.5}$.

1. Introduction

The solution of a linear system $Ax = b$ is one of the oldest problems in mathematics. One of the most fundamental and important techniques for solving a linear system is Gaussian elimination, in which a matrix is factored into the product of a lower and upper triangular matrix. Given an $n \times n$ matrix A, Gaussian elimination performs a sequence of rank-one transformations, resulting in the sequence of matrices $A^{(k)} \in \mathbb{C}^{k \times k}$ for k equals n to 1, satisfying

$$A^{(k)} = M^{(2,2)} - M^{(2,1)}[M^{(1,1)}]^{-1}M^{(1,2)}, \qquad \text{where } A = \begin{array}{c} \begin{array}{cc} \scriptstyle n-k & \scriptstyle k \end{array} \\ \begin{bmatrix} M^{(1,1)} & M^{(1,2)} \\ M^{(2,1)} & M^{(2,2)} \end{bmatrix} \end{array} \begin{array}{c} \scriptstyle n-k \\ \scriptstyle k \end{array} . \tag{1.1}$$

The resulting LU factorization of A is encoded by the first row and column of each of the iterates $A^{(k)}$, $k = 1, ..., n$. Not all matrices have an LU factorization, and a permutation of the rows (or columns) of the matrix may be required. In addition, performing computations in finite precision can elicit issues due to round-off error. The error due to rounding in Gaussian elimination for a matrix A in some fixed precision is controlled by the growth factor of the Gaussian elimination algorithm, defined by

$$g(A) := \frac{\max_k \|A^{(k)}\|_{\max}}{\|A\|_{\max}},$$

where $\|\cdot\|_{\max}$ is the entry-wise matrix infinity norm (see [11, Theorem 3.3.1] for details[1]). For this reason, understanding the growth factor is of both theoretical and practical importance. Complete pivoting, famously referred to as "customary" by von Neumann [25], is a strategy for permuting the rows and columns of a matrix so that, at each step, the pivot (the top-left entry of $A^{(k)}$) is the largest magnitude entry of $A^{(k)}$. Complete pivoting remains the premier theoretical permutation strategy for performing Gaussian elimination. Despite its popularity, the worst-case behavior of the growth factor under complete pivoting is poorly understood. This is in stark contrast to partial pivoting, an alternative strategy which is incredibly popular in practice but known to be horribly unstable in the worst case (see Wilkinson's 1965 *The Algebraic Eigenvalue Problem* [27, pg. 212]). Here, we focus exclusively on the pure mathematical problem of the growth factor under complete pivoting. For the

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA.
E-mail addresses: ankitb12@mit.edu, edelman@mit.edu, urschel@mit.edu.
2020 *Mathematics Subject Classification.* Primary 65F05, 15A23.

[1] Here, we study the growth factor in exact arithmetic, while it is growth factor in floating point arithmetic that occurs in error estimates. The worst-case behavior of these two quantities is very similar (see [8, Theorem 1.5]).

reader interested in the engineering aspects of solving a linear system, a detailed discussion of the role of the growth factor and complete vs partial pivoting in modern practice is provided in Appendix A.

1.1. **Historical Overview and Relevant Results.** In their seminal 1947 paper *Numerical Inverting of Matrices of High Order*, von Neumann and Goldstine studied the stability of Gaussian elimination with complete pivoting [25]. This work was motivated by their development of the first stored-program digital computer and desire to understand the effect of rounding in computations on it [17]. Goldstine later wrote:

> Indeed, von Neumann and I chose this topic for the first modern paper on numerical analysis ever written precisely because we viewed the topic as being absolutely basic to numerical mathematics [10].

However, it was not until Wilkinson's 1961 paper *Error Analysis of Direct Methods of Matrix Inversion* that a more rigorous analysis of the backward error in Gaussian elimination due to rounding errors occurred. Indeed, Wilkinson was the first to fully recognize the dependence of this error on the growth factor. Let $g_n(\mathbb{R})$ and $g_n(\mathbb{C})$ denote the maximum growth factor under complete pivoting over all non-singular $n \times n$ real and complex matrices, respectively. Wilkinson produced a bound for the growth factor under complete pivoting using only Hadamard's inequality [26, Equation 4.15]:

$$g_n(\mathbb{C}) \le \sqrt{n}\big(2\, 3^{1/2} \ldots n^{1/(n-1)}\big)^{1/2} \le 2\sqrt{n}\, n^{\ln(n)/4}, \tag{1.2}$$

where the second inequality is asymptotically tight. This estimate was considered extremely pessimistic, with Wilkinson himself noting that "no matrix has been encountered for which [the growth factor for complete pivoting] was as large as 8 [26]." A conjecture that the growth factor for complete pivoting of a real $n \times n$ matrix was at most n was eventually formed (see [8, Section 1.1] for a detailed discussion of the conjecture and its possible mis-attribution to both Cryer and Wilkinson). According to Higham, in his now-classic text *Accuracy and Stability of Numerical Algorithms*, this conjecture "became one of the most famous open problems in numerical analysis, and has been investigated by many mathematicians [14, pg. 181]." Many researchers attempted to upper bound the growth factor, with $g_n(\mathbb{R})$ computed exactly for $n = 1, 2, 3, 4$ and shown to be strictly less than five for $n = 5$ (see the works of Tornheim [21, 22, 23, 24], Cryer [4], and Cohen [3] for details). However, no progress was made on improving the bound for arbitrary n. Many years later, in 1991, Gould found a 13×13 matrix with growth factor larger than 13 in finite precision [12] (extended to exact arithmetic by Edelman [7]), providing a counterexample to the conjecture for $n = 13$. Recently, Edelman and Urschel improved the best-known lower bounds for all $n > 8$ and showed that

$$g_n(\mathbb{R}) \ge 1.0045\, n \;\text{ for all } n \ge 11, \quad \text{and} \quad \limsup_n \big(g_n(\mathbb{R})/n\big) \ge 3.317,$$

thus disproving the aforementioned conjecture for all $n \ge 11$ by a multiplicative factor [8]. However, for the upper bound, to date no improvement has been made to Wilkinson's bound.

1.2. **Our Contributions.** In this work, we improve Wilkinson's upper bound by an exponential constant, the first improvement in over sixty years. In particular, we prove the following theorem, obtaining a leading exponential constant of $\frac{1}{2[2+(2-\sqrt{2})\ln 2]} \approx 0.20781$.

Theorem 1.1. $g_n(\mathbb{C}) \le n^{\frac{\ln n}{2[2+(2-\sqrt{2})\ln 2]}+0.91}$.

Our proof consists of four parts:

(1) A Generalized Hadamard's inequality: We prove a tighter version of Hadamard's famous inequality for matrices with a large low-rank component. This generalization allows for a more sophisticated analysis of the iterates of Gaussian elimination, providing additional constraints on the pivots of a matrix. (Subsection 3.1)

(2) An Improved Optimization Problem: Applying the improved Hadamard inequality produces an optimization problem that can be considered a refinement of the optimization problem associated with Wilkinson's proof. Unfortunately, this refinement is no longer linear upon a logarithmic transformation. (Subsection 3.2)
(3) From Non-Linear to Linear: We relax the logarithmic transformation of our optimization problem to a linear program, and prove that the optimal value of our relaxation has the same asymptotic behavior. (Subsection 3.3)
(4) An Asymptotic Analysis: Finally, we analyze the asymptotic behavior of our linear program by converting it into a continuous program and applying a duality argument, thus producing the improved bound in Theorem 1.1. (Section 4)

Our proof considers the same information regarding the underlying matrix as Wilkinson's original bound, using only the pivots at each step of elimination, and reveals further structure regarding the relationships between them. Our technique increases our understanding of the mathematical forces that constrain entries from increasing in size during Gaussian elimination, by illustrating the trade-off between having entries that grow quickly in size and having a matrix of large numerical rank (e.g., many large singular values). Improved estimates on the explicit constants in Theorem 1.1 can be obtained through a refinement of the techniques presented herein. However, tight estimates on the maximum growth factor will likely require further information regarding matrix entries.

The techniques employed here can likely be used to improve upper bounds for the growth factor problem under other pivoting strategies (e.g., rook pivoting, threshold pivoting, etc.). We leave this natural extension to the interested reader.

1.3. **Notation and Basic Observations.** Recall that $A^{(k)}$, defined in Equation 1.1, denotes the $k \times k$ matrix resulting from the $(n-k)^{th}$ step of Gaussian elimination, and let p_k denote the pivot of $A^{(k)}$ for $k = 1, \dots, n$. Let $\langle \cdot, \cdot \rangle_F$ and $\| \cdot \|_F$ denote the Frobenius inner product and norm. Gaussian elimination under complete pivoting permutes the rows and columns of a matrix A so that $p_k = \|A^{(k)}\|_{\max}$ for all k. Without loss of generality, we may assume A is already completely pivoted, removing the need for pivoting in analysis. For complete pivoting, the growth factor is given by $\max_k p_k/p_n$, as $p_k = \|A^{(k)}\|_{\max}$, $k = 1, \dots, n$, and $p_n = \|A\|_{\max}$. Because we are interested in the maximum growth factor over all $n \times n$ matrices, it suffices to consider the maximum value of p_1/p_n [5, Proposition 2.9].

2. Wilkinson's Bound Viewed as a Linear Program

The proof of Wilkinson's 1961 bound is incredibly short, requiring one page of mathematics and using only Hadamard's inequality applied to the matrix iterates $A^{(k)} \in \mathbb{C}^{k \times k}$ of Gaussian elimination and the well-known fact that the product of pivots for a matrix equals its determinant. Hadamard's inequality, that the modulus of the determinant of a matrix is at most the product of the two-norm of its columns, implies that

$$\prod_{i=1}^{k} p_i = \det(A^{(k)}) \le k^{k/2} |A^{(k)}|_\infty^k = k^{k/2} p_k^k. \tag{2.1}$$

The maximum k^{th} pivot, viewed as a function of k, is non-decreasing, and so the maximum value of p_1/p_n under these constraints provides an upper bound for the maximum growth factor:

Wilkinson's Optimization Problem

$$\begin{array}{ll} \max & p_1/p_n \\ \text{s.t.} & \prod_{i=1}^{k} p_i \le k^{k/2} p_k^k \quad \text{for } k = 1, ..., n. \end{array} \tag{2.2}$$

Performing the transformation $q_k = \ln(p_k)$ for $k = 1, ..., n$ produces the linear program (LP):

Wilkinson's Linear Program

$$\begin{array}{ll} \max & q_1 - q_n \\ \text{s.t.} & \sum_{i=1}^k q_i \le \frac{k}{2}\ln k + k\, q_k \quad \text{for } k = 1, ..., n. \end{array} \tag{2.3}$$

Wilkinson's proof, though never stated in the context of linear programming, can be viewed as a simple LP duality argument for the linear program maximize $c^T x$ subject to $Ax \le b$, where $x = (q_1, \dots, q_n)^T$, $c = (1, 0, \dots, 0, -1)^T$, and $Ax \le b$ encodes the inequalities of Linear Program 2.3:

$$Ax = \begin{pmatrix} -1 & & & & \\ 1 & -1 & & & \\ 1 & 1 & -2 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 1 & 1 & \cdots & 1 & -(n-1) \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{pmatrix} \le \begin{pmatrix} 0 \\ \frac{2}{2}\ln 2 \\ \frac{3}{2}\ln 3 \\ \vdots \\ \frac{n}{2}\ln n \end{pmatrix} = b.$$

We note that the additional constraint $q_1 \ge 0$ plays no role, as the feasible region of Program 2.3 is shift-independent. The matrix A has an easily computable inverse with $A^{-1}_{i1} = -1$ for $i = 1, ..., n$, $A^{-1}_{ii} = -\frac{1}{i-1}$ for $i = 2, ..., n$, and $A^{-1}_{ij} = -\frac{1}{j(j-1)}$ for $i > j$. The quantity

$$[A^{-1}]^T c = \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 \\ & -1 & -\frac{1}{2} & \cdots & -\frac{1}{2} \\ & & -\frac{1}{2} & & \vdots \\ & & & \ddots & -\frac{1}{(n-2)(n-1)} \\ & & & & -\frac{1}{n-1} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{(n-2)(n-1)} \\ \frac{1}{n-1} \end{pmatrix}$$

is entry-wise non-negative, implying Wilkinson's bound

$$q_1 - q_n = \left([A^{-1}]^T c\right)^T Ax \le \left([A^{-1}]^T c\right)^T b = \frac{1}{2}\left[\ln n + \sum_{k=2}^{n} \frac{\ln k}{k-1}\right].$$

This bound is the exact solution to Program 2.3, evidenced by the matching feasible point $x = A^{-1}b$. The ease with which the optimal point of the dual program can be obtained is due to the simple structure of the constraints. Our improved linear program, described in Subsection 3.3, has a more complicated set of constraints, requiring a more complex duality argument (given in Section 4).

This same argument also immediately produces bounds for the geometric mean growth factor of the iterates $A^{(k)}$, a key quantity in our proof of Theorem 1.1 that may be of independent interest. Indeed, the quantity $\frac{1}{n}\sum_{k=1}^n (q_1 - q_k)$ can be upper bounded by analyzing the linear program:

Geometric Mean Growth LP

$$\begin{array}{ll} \max & \frac{1}{n}\sum_{k=1}^n (q_1 - q_k) \\ \text{s.t.} & \sum_{i=1}^k q_i \le \frac{k}{2}\ln k + k\, q_k \quad \text{for } k = 1, ..., n. \end{array} \tag{2.4}$$

The constraints of this linear program are identical to those of Program 2.3. The only difference is in the objective; here we have $c = \left(\frac{n-1}{n}, -\frac{1}{n}, ..., -\frac{1}{n}\right)^T$. Nevertheless, the quantity

$$[A^{-1}]^T c = \begin{pmatrix} -1 & -1 & -1 & \cdots & -1 \\ & -1 & -\frac{1}{2} & \cdots & -\frac{1}{2} \\ & & -\frac{1}{2} & & \vdots \\ & & & \ddots & -\frac{1}{(n-2)(n-1)} \\ & & & & -\frac{1}{n-1} \end{pmatrix} \begin{pmatrix} \frac{n-1}{n} \\ -\frac{1}{n} \\ \vdots \\ -\frac{1}{n} \\ -\frac{1}{n} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{(n-2)(n-1)} \\ \frac{1}{(n-1)n} \end{pmatrix}$$

is entry-wise non-negative, implying the bound

$$\frac{1}{n}\sum_{k=1}^{n}(q_1 - q_k) = \left([A^{-1}]^T c\right)^T Ax \leq \left([A^{-1}]^T c\right)^T b = \frac{1}{2}\sum_{k=2}^{n}\frac{\ln k}{k-1} \leq \frac{\ln^2 n}{4} + \ln 2, \tag{2.5}$$

or, in terms of the original pivots,

$$\left[\prod_{k=1}^{n}\frac{p_1}{p_k}\right]^{\frac{1}{n}} = \frac{p_1}{\left(\prod_{k=1}^{n} p_k\right)^{1/n}} \leq 2n^{\frac{1}{4}\ln n}.$$

This can be easily generalized further to any weighted average $\sum_{k=1}^{n} w_k(q_1 - q_k)$ of the logarithmic growth factors.

3. An Improved Linear Program

In this section, we produce additional constraints that the pivots must satisfy by generalizing Hadamard's inequality for matrices with a large low-rank component. These constraints, applied to the matrix $A^{(k)}$ (viewed as a sub-matrix of $A^{(k+\ell)}$ plus a rank ℓ matrix), lead to a new linear program with optimal value at most $0.2079\ln^2 n + O(\ln n)$, the first improvement to the exponential constant of 0.25 in Wilkinson's bound (Inequality 1.2).

3.1. **Improved Determinant Bounds.** First, we recall the following basic proposition, itself a corollary of [15, Theorem 1]. [2]

Proposition 3.1. *$|\det(A+B)| \leq \prod_{i=1}^{n}\left(\sigma_i(A) + \sigma_{n-i+1}(B)\right)$ for all $A, B \in \mathbb{C}^{n\times n}$, where $\sigma_1(A) \geq ... \geq \sigma_n(A)$ and $\sigma_1(B) \geq ... \geq \sigma_n(B)$ are the singular values of A and B.*

Next, we produce a generalized version of Hadamard's inequality for matrices with a large low-rank component. Here and in what follows, we use the convention that $0^0 = 1$.

Lemma 3.2. *Let $A, B \in \mathbb{C}^{n\times n}$ with $\|A\|_F \leq n$ and $\|B\|_F \leq Cn$, and $\operatorname{rank}(B) \leq \ell$. Then*

$$|\det(A+B)| \leq \frac{n^n}{(n-\ell)^{\frac{n-\ell}{2}}\ell^{\frac{\ell}{2}}}\left(1+C\right)^{\ell}.$$

[2]Proposition 3.1 also follows from applying standard determinant bounds for Hermitian matrices [2, Theorem VI.7.1] to $\left(\begin{smallmatrix} 0 & A \\ A^* & 0 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 0 & B \\ B^* & 0 \end{smallmatrix}\right)$, and using the following well-known rearrangement inequality: for any $a_1 \geq ... \geq a_n \geq 0$, $b_1 \geq ... \geq b_n \geq 0$, and $\pi \in \mathrm{S}_n$, $\prod_{i=1}^{n}(a_i + b_{\pi(i)}) \leq \prod_{i=1}^{n}(a_i + b_{n-i+1})$.

Proof. Let $0 < \ell < n$, and $\sigma_1(A) \geq ... \geq \sigma_n(A)$ and $\sigma_1(B) \geq ... \geq \sigma_n(B)$ denote the singular values of A and B. By Proposition 3.1,

$$\begin{aligned} |\det(A+B)| &\leq \left(\prod_{i=1}^{n-\ell} \sigma_i(A)\right) \prod_{j=1}^{\ell} \big(\sigma_j(B) + \sigma_{n-j+1}(A)\big) \\ &\leq \left(\frac{1}{n-\ell}\sum_{i=1}^{n-\ell} \sigma_i^2(A)\right)^{\frac{n-\ell}{2}} \left(\frac{1}{\ell}\sum_{j=1}^{\ell}\sigma_j(B) + \frac{1}{\ell}\sum_{j=1}^{\ell}\sigma_{n-j+1}(A)\right)^{\ell} \\ &\leq \left(\frac{1}{n-\ell}\sum_{i=1}^{n-\ell} \sigma_i^2(A)\right)^{\frac{n-\ell}{2}} \left(\frac{1}{\ell^{\frac{1}{2}}}\left[\sum_{j=1}^{\ell}\sigma_j^2(B)\right]^{\frac{1}{2}} + \frac{1}{\ell^{\frac{1}{2}}}\left[\sum_{j=1}^{\ell}\sigma_{n-j+1}^2(A)\right]^{\frac{1}{2}}\right)^{\ell} \\ &\leq \left(\frac{n^2}{n-\ell}\right)^{\frac{n-\ell}{2}} \left(\frac{Cn}{\ell^{\frac{1}{2}}} + \frac{n}{\ell^{\frac{1}{2}}}\right)^{\ell} \\ &= \frac{n^n}{(n-\ell)^{\frac{n-\ell}{2}}\ell^{\frac{\ell}{2}}}(1+C)^{\ell}, \end{aligned}$$

where we have used the AM-GM inequality in the second inequality and Cauchy-Schwarz in the third. The result for the cases $\ell = 0$ and $\ell = n$ follows from gently modified versions of the same analysis. □

We note that, when $\ell = 0$, Lemma 3.2 implies the well-known corollary $|\det(A)| \leq n^{n/2}|A|_\infty$ of Hadamard's inequality. A tighter version of Lemma 3.2 can be obtained at the cost of brevity, by explicitly maximizing with respect to the parameter $x := \sum_{j=1}^{\ell}\sigma_{n-j+1}^2(A)$ rather than upper bounding both $\sum_{i=1}^{n-\ell}\sigma_i^2(A)$ and $\sum_{j=1}^{\ell}\sigma_{n-j+1}^2(A)$ with n^2. However, this optimization does not lead to any improvement in the exponential constant of Theorem 1.1, and so its derivation is left to the interested reader.

3.2. **An Improved Optimization Problem.** Lemma 3.2 applied to the matrix iterates $A^{(k)} \in \mathbb{C}^{k\times k}$ of Gaussian elimination under complete pivoting leads to further constraints on the pivots $p_k = |A^{(k)}|_\infty$. Consider some $0 < \ell < k$ with $k + \ell \leq n$. Using block notation, let $N^{(1,1)}$, $N^{(1,2)}$, $N^{(2,1)}$, and $N^{(2,2)}$ denote the upper-left $\ell\times\ell$, upper-right $\ell\times k$, lower-left $k\times\ell$, and lower-right $k\times k$ sub-matrices of $A^{(k+\ell)}$. After ℓ further steps of Gaussian elimination applied to $A^{(k+\ell)}$, we obtain

$$A^{(k+\ell)} = \begin{bmatrix} N^{(1,1)} & N^{(1,2)} \\ N^{(2,1)} & N^{(2,2)} \end{bmatrix} = \begin{bmatrix} \widetilde{L} & 0 \\ N^{(2,1)}\widetilde{U}^{-1} & I \end{bmatrix} \begin{bmatrix} \widetilde{U} & \widetilde{L}^{-1}N^{(1,2)} \\ 0 & N^{(2,2)} - N^{(2,1)}[N^{(1,1)}]^{-1}N^{(1,2)} \end{bmatrix},$$

where $\widetilde{L}\widetilde{U}$ is the LU factorization of $N^{(1,1)}$, implying that

$$A^{(k)} = N^{(2,2)} - N^{(2,1)}[N^{(1,1)}]^{-1}N^{(1,2)}.$$

For the sake of space, let $X := N^{(2,2)}$ and $Y := N^{(2,1)}[N^{(1,1)}]^{-1}N^{(1,2)}$, and note that Y has rank at most ℓ. We may rewrite $A^{(k)}$ as

$$A^{(k)} = \left(X - \frac{\mathrm{Re}\langle X, Y\rangle_F}{\|Y\|_F^2}Y\right) - \left(1 - \frac{\mathrm{Re}\langle X, Y\rangle_F}{\|Y\|_F^2}\right)Y. \tag{3.1}$$

We note that

$$\left\| X - \frac{\mathrm{Re}\langle X, Y\rangle_F}{\|Y\|_F^2}Y\right\|_F^2 = \|X\|_F^2 - \frac{\big(\mathrm{Re}\langle X, Y\rangle_F\big)^2}{\|Y\|_F^2} \leq \|X\|_F^2 \leq p_{k+\ell}^2 n^2$$

and

$$\left\| \left(1 - \frac{\mathrm{Re}\langle X, Y\rangle_F}{\|Y\|_F^2}\right) Y \right\|_F^2 = \|Y\|_F^2 - 2\,\mathrm{Re}\langle X, Y\rangle_F + \frac{\big(\mathrm{Re}\langle X, Y\rangle_F\big)^2}{\|Y\|_F^2}$$
$$\leq \|Y\|_F^2 - 2\,\mathrm{Re}\langle X, Y\rangle_F + \|X\|_F^2$$
$$= \|X - Y\|_F^2 \leq p_k^2 n^2,$$

as the entries of $A^{(k)}$ and $N^{(2,2)}$ have modulus at most p_k and $p_{k+\ell}$, respectively. Applying Lemma 3.2 to $A^{(k)}$ using the splitting in Equation 3.1, we obtain the bound

$$\frac{\prod_{i=1}^k p_i}{p_{k+\ell}^k} = \frac{\det(A^{(k)})}{p_{k+\ell}^k} \leq \frac{k^k}{(k-\ell)^{\frac{k-\ell}{2}} \ell^{\frac{\ell}{2}}} \left(1 + \frac{p_k}{p_{k+\ell}}\right)^{\ell}. \tag{3.2}$$

Making use of these additional constraints gives the following refinement of Optimization Problem 2.2:

Improved Optimization Problem

$$\begin{array}{lll} \max & p_1/p_n & \\ \text{s.t.} & \prod_{i=1}^k p_i \leq k^{k/2} p_k^k & \text{for } k = 1, ..., n \\ & \prod_{i=1}^k p_i \leq \dfrac{k^k p_{k+\ell}^{k-\ell} (p_k + p_{k+\ell})^{\ell}}{(k-\ell)^{\frac{k-\ell}{2}} \ell^{\frac{\ell}{2}}} & \text{for } \ell = 1, ..., \min\{k-1, n-k\} \\ & & \quad k = 2, ..., n-1. \end{array} \tag{3.3}$$

3.3. **From a Non-Linear to Linear Program.** The additional constraints given by Inequality 3.2 for $k = 2, ..., n-1$ and $\ell = 1, ..., \min\{k-1, n-k\}$ produce an optimization problem (Optimization Problem 3.3) that is no longer linear upon the transformation $q_k = \ln(p_k)$, $k = 1, ..., n$. For this reason, we relax Optimization Problem 3.3 in order to maintain linearity. For simplicity, we do so while giving only minor attention to lower-order terms (i.e., terms that do not affect the leading exponential constant). More complicated linear programs with improved behavior for finite n can be obtained by a more involved analysis.

Consider an arbitrary feasible point $(p_1, ..., p_n)$ of Optimization Problem 3.3. We claim that $(p_1, ..., p_n)$ also satisfies

$$\prod_{i=1}^k p_i \leq (\tfrac{11}{4} k)^{k/2} p_{k+\ell}^{k-\ell}\, p_k^{\ell} \qquad \text{for} \quad \begin{array}{l} \ell = 1, ..., \min\{k-1, n-k\} \\ k = 2, ..., n-1. \end{array} \tag{3.4}$$

We break our analysis into two cases. If $p_k \leq (\sqrt{11}/2)^{k/(k-\ell)} p_{k+\ell}$, then

$$\prod_{i=1}^k p_i \leq k^{k/2} p_k^k \leq (\tfrac{11}{4} k)^{k/2} p_{k+\ell}^{k-\ell}\, p_k^{\ell}.$$

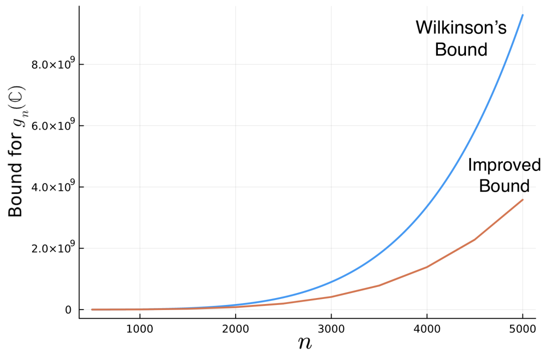
Conversely, if $p_k \geq (\sqrt{11}/2)^{k/(k-\ell)} p_{k+\ell}$, then

$$\prod_{i=1}^k p_i \leq \frac{k^k p_{k+\ell}^{k-\ell} (p_k + p_{k+\ell})^{\ell}}{(k-\ell)^{\frac{k-\ell}{2}} \ell^{\frac{\ell}{2}}} \leq k^{k/2} p_{k+\ell}^{k-\ell}\, p_k^{\ell} \left(\frac{k^{k/2} \big(1 + \big(\frac{2}{\sqrt{11}}\big)^{k/(k-\ell)}\big)^{\ell}}{(k-\ell)^{\frac{k-\ell}{2}} \ell^{\frac{\ell}{2}}} \right) \leq (\tfrac{11}{4} k)^{k/2} p_{k+\ell}^{k-\ell} p_k^{\ell},$$

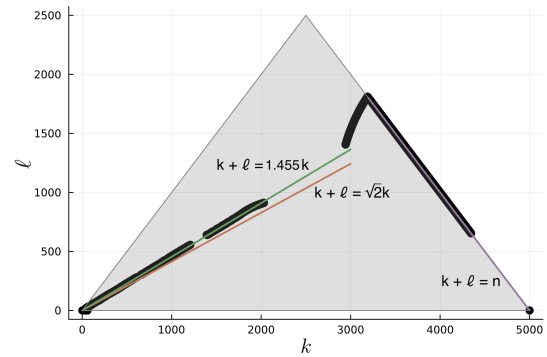
where we have used the fact that

$$\max_{t \in (0,1)} \left(\frac{1}{t}\right)^{\frac{t}{2}} \left(\frac{1}{1-t}\right)^{\frac{1-t}{2}} = \sqrt{2} \quad \text{and} \quad \max_{t \in (0,1)} \left(1 + \left(\frac{2}{\sqrt{11}}\right)^{1/(1-t)}\right)^{t} \approx 1.168 < \sqrt{\frac{11}{8}}.$$

Applying the transformation $q_k = \ln(p_k)$, $k = 1, ..., n$, to Inequality 3.4, we obtain the linear program:



(a) Comparing Wilkinson's Bound to our Improved Bound for $g_n(\mathbb{C})$ for $n \leq 5000$



(b) Active constraints parameterized by (k, ℓ) for the optimal solution to Program 3.5 at $n = 5000$

FIGURE 1. Comparing our Improved Linear Program to Wilkinson's LP: Figure (a) illustrates the difference between Wilkinson's bound for $g_n(\mathbb{C})$ (Inequality 1.2) and the upper bound produced by the optimal value of Program 3.5 for $n \leq 5000$. Figure (b) is a scatter plot of the pairs (k, ℓ) for which the corresponding inequality in Program 3.5 is tight for a numerically computed optimal solution at $n = 5000$. The grey shaded triangle shows the set of (k, ℓ) corresponding to constraints of Program 3.5, with Wilkinson's constraints parameterized by $(k, 0)$, and the black dots represent the subset of those constraints that are active for the numerically computed optimal solution. For $n = 5000$, almost none of Wilkinson's constraints are active. The red line $k + \ell = \sqrt{2}k$ is the set of constraints used to prove Theorem 1.1, and the green line denotes the asymptotically tight constraints for the feasible point produced in Subsection 4.1. While the points on the purple line $k + \ell = n$ improves the objective value, these constraints do not play a role in the asymptotic leading term of the solution to the linear program.

Improved Linear Program

$$\begin{aligned}
\max \quad & q_1 - q_n \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{k} q_i \leq \frac{k}{2}\ln(k) + kq_k && \text{for } k = 1, ..., n \\
& \textstyle\sum_{i=1}^{k} q_i \leq \frac{k}{2}\ln(\frac{11}{4}k) + (k-\ell)q_{k+\ell} + \ell q_k && \text{for } \ell = 1, ..., \min\{k-1, n-k\} \\
& && \quad k = 2, ..., n-1.
\end{aligned} \tag{3.5}$$

and note that the maximum growth factor $g_n(\mathbb{C})$ is upper bounded by e^{OPT}, where OPT is the optimal value of this linear program. Program 3.5 is an improved version of Wilkinson's linear program (Program 2.3), containing all of Wilkinson's constraints as well as additional bounds representing long-range interactions (i.e., bounds relating $A^{(k)}$ and $A^{(k+\ell)}$). In addition, we note that the optimal value of Program 3.5 and the logarithm of the optimal value of Program 3.3 are asymptotically equal up to lower order terms:

Proposition 3.3. *If OPT is the optimal value of Linear Program 3.5 for n, then the optimal value of Optimization Problem 3.3 for n lies in the interval* $[n^{-3/2}e^{OPT}, e^{OPT}]$.

Proof. Let $(q_1, ..., q_n)$ be a feasible point of Linear Program 3.5. It suffices to show that $p_k = k^{3/2}e^{q_k}$, $k = 1, ..., n$, is a feasible point of Optimization Problem 3.3. Considering an arbitrary constraint parameterized by $k > 1$ and ℓ, we have

$$\prod_{i=1}^{k} p_i = (k!)^{\frac{3}{2}} \exp\left\{\sum_{i=1}^{k} q_i\right\} \le (k!)^{\frac{3}{2}} \exp\left\{\frac{k}{2}\ln(\tfrac{11}{4}k) + (k-\ell)q_{k+\ell} + \ell q_k\right\}.$$

Rewriting the right-hand side in terms of p_k gives

$$\prod_{i=1}^{k} p_i \le \frac{(k!)^{\frac{3}{2}}(\frac{11}{4}k)^{\frac{k}{2}} p_{k+\ell}^{k-\ell} p_k^{\ell}}{(k+\ell)^{\frac{3}{2}(k-\ell)} k^{\frac{3}{2}\ell}} \le k^{\frac{k}{2}} p_{k+\ell}^{k-\ell} p_k^{\ell} \frac{(k!)^{\frac{3}{2}}(\frac{11}{4})^{\frac{k}{2}}}{k^{\frac{3}{2}k}} \le k^{\frac{k}{2}} p_{k+\ell}^{k-\ell} p_k^{\ell} \le \frac{k^k p_{k+\ell}^{k-\ell}(p_k + p_{k+\ell})^{\ell}}{(k-\ell)^{\frac{k-\ell}{2}} \ell^{\frac{\ell}{2}}},$$

completing the proof. □

In the following section, we provide nearly matching upper and lower bounds on the optimal value of Program 3.5 for sufficiently large n, thereby proving Theorem 1.1.

3.4. **Bounding the Growth Factor in Practice.** While the proof of Theorem 1.1 focuses on the behavior for large n, we note that an improvement in exponential constant exists in practice for reasonably sized matrices as well. We provide a comparison of the optimal value of Program 3.5 to the optimal value of Wilkinson's LP in Figure 1 for $n \le 5000$. The numerically computed solutions to Program 3.5 were obtained using the Gurobi Optimizer [13] called through the JuMP package for mathematical optimization [16] in the Julia programming language [1]. We stress that numerically computed solutions to a linear program can be converted into mathematical bounds via a dual feasible point verified in exact arithmetic. In addition, Program 3.5 can be adapted in a number of ways for computational efficiency. For instance, the linear transformation $Q(k) = \sum_{i=1}^{k} q_i$ produces a linear program with a simple objective and sparse constraints (at most four variables in each). Furthermore, as the analysis in Section 4 suggests, only a linear number of constraints are required to produce a reasonable upper bound for the optimal value. One natural choice consists of Wilkinson's original constraints and additional constraints of the form $k + \ell = n$ and $k + \ell \in [\sqrt{2}k - 1, \sqrt{2}k + C]$ for some constant C (Theorem 1.1 is proved using only constraints of the form $k + \ell = \lceil\sqrt{2}k\rceil$). Finally, we stress that the techniques used to produce improved estimates can be further optimized to obtain even better bounds in both theory and practice. We hope that the interested reader will do so.

4. Bounding the Optimal Value of our Linear Program

Finally, we prove that the objective of Program 3.5 satisfies the bound

$$\max\; q_1 - q_n \;\le\; \alpha \ln^2 n + (\beta + 1/2)\ln n, \qquad \text{where} \quad \alpha = \frac{1}{2(2 + (2-\sqrt{2})\ln 2)}$$

and $\beta = 0.41$, thus completing the proof of Theorem 1.1 ($\beta = 0.41$ corresponds to the constant $\beta + 1/2 = 0.91$ in Theorem 1.1). We do so via a duality argument, making use of the constraints for k and ℓ satisfying $k + \ell \approx \sqrt{2}k$. Before proving the above bound, we first illustrate why $[2(2+(2-\sqrt{2})\ln 2)]^{-1}$ is the correct choice of α for constraints of the form $k + \ell \approx \sqrt{2}k$, and show that this choice is within 0.00024 of the exact asymptotic constant of Program 3.5.

4.1. **On the Choice and Optimality of the Constant $\alpha = [2(2 + (2-\sqrt{2})\ln 2)]^{-1}$.** Suppose that $q_x - q_1 = -\gamma \ln^2 x + O(1)$. Then, for the constraint

$$\sum_{i=1}^{k}(q_i - q_1) \le \frac{k}{2}\ln(\tfrac{11}{4}k) + (k-\ell)(q_{k+\ell} - q_1) + \ell(q_k - q_1),$$

the left-hand side equals

$$\int_1^k -\gamma \ln^2 x \,\mathrm{d}x + O(k) = -\gamma k \ln^2 k + 2\gamma k \ln k + O(k)$$

and the right-hand side equals

$$-\gamma k \ln^2 k + \big[k/2 - 2\gamma(k-\ell)\ln(1+\ell/k)\big]\ln k + O(k).$$

Letting $t = \ell/k$, the right-hand side is asymptotically larger than the left-hand side if

$$\gamma \le \frac{1}{4(1+(1-t)\ln(1+t))}.$$

The values $t = 0$ and $t = 1$ (i.e., when $\ell = 0$ or $\ell = k$) correspond to the constraints of Wilkinson's linear program, and for $t = 0$ and $t = 1$, we obtain $\gamma \le 1/4$ (e.g., Wilkinson's bound). The value $t = \sqrt{2} - 1$ produces the upper bound $1/[2(2+(2-\sqrt{2})\ln 2)] \approx 0.20781$ of Theorem 1.1. The quantity $[4(1+(1-t)\log(1+t))]^{-1}$ on the interval $[0,1]$ is minimized by $t = \exp\{W(2e)-1\} - 1 \approx 0.4547$, where $W(x)$ is the Lambert W function, with a minimum value of

$$\frac{1}{4\big(1+(2-e^{W(2e)-1})(W(2e)-1)\big)} \approx 0.207576.$$

This implies the existence of a solution to Program 3.5 with $q_1 - q_n = 0.207575\ln^2 n - O(\ln n)$, thus illustrating that our upper bound of $\alpha = [2(2+(2-\sqrt{2})\ln 2)]^{-1} \approx 0.207811$ is within 0.00024 of the optimal value of the linear program. We do not pursue further improvement on this constant.

4.2. **Reducing Theorem 1.1 to Geometric Mean Growth.** For ease of analysis, we consider a continuous version of our variables $q = (q_1, ..., q_n)$. Let

$$f(x) = q_{\lceil x\rceil} - q_1 \qquad \text{and} \qquad F(x) = \frac{1}{x}\int_0^x f(t)\,\mathrm{d}t \qquad \text{for } x > 0,$$

where $\{q_k\}_{k=1}^{\infty}$ is any sequence such that $q_1 \ge q_k$ for all $k \in \mathbb{N}$ and $(q_1, ..., q_n)$ is a feasible point of Program 3.5 for all $n \in \mathbb{N}$. $F(x)$ can be thought of a continuous version of geometric mean growth (described in Section 2). Any optimal solution $(q_1, ..., q_n)$ for the n-dimensional linear program can be converted into such a sequence by simply setting $q_k = q_n$ for all $k > n$. The constraint of Program 3.5 with $k = \lceil x\rceil$ and $\ell = \lceil\sqrt{2}x\rceil - \lceil x\rceil$ implies that for all $x > 0$,

$$\begin{aligned} F(\lceil x\rceil) &\le \frac{\ln(\frac{11}{4}\lceil x\rceil)}{2} + \left(\frac{2\lceil x\rceil - \lceil\sqrt{2}x\rceil}{\lceil x\rceil}\right) f(\sqrt{2}x) + \left(\frac{\lceil\sqrt{2}x\rceil - \lceil x\rceil}{\lceil x\rceil}\right) f(x) \\ &\le \frac{\ln(\frac{11}{4}x)}{2} + \frac{1}{2x} + \left(\sqrt{2} - 1 - \frac{\sqrt{2}}{x}\right)\left(\sqrt{2}f(\sqrt{2}x) + f(x)\right). \end{aligned} \tag{4.1}$$

We make the following claim regarding $F(x)$ (recall, $\alpha = [2(2+(2-\sqrt{2})\ln 2)]^{-1}$ and $\beta = 0.41$).

Lemma 4.1. *$F(x) > -\alpha\ln^2 x - \beta\ln x$ for all $x > 100$.*

Lemma 4.1 implies our desired result, as

$$F(n) = \frac{1}{n}\sum_{i=1}^{n}(q_i - q_1) \le \frac{1}{n}\left(\frac{n}{2}\ln n + nq_n - nq_1\right),$$

and $\alpha\ln^2 n + (\beta + 1/2)\ln n$ is larger than Wilkinson's bound for $x \le 100$. A tighter bound may be obtained by adding together constraints of the form $k + \ell = n$ for $k \ge n/(8\alpha)$ (e.g., the constraints appearing in Figure 1(b)). However, the analysis is involved and the improvement on the $1/2\ \ln n$ term produced by the argument above is minor (≈ 0.046 improvement, at the cost of lower-order terms).

4.3. **Proof of Lemma 4.1: Base Case.** The proof of Lemma 4.1 is, in spirit, by "induction on x" via a duality argument. Clearly the assertion holds for $x \in (100, 1700]$ for β sufficiently large. However, verifying the base case of $x \in (100, 1700]$ for $\beta = 0.41$ requires some analysis, as the quantity $\alpha \ln^2 n + \beta \ln n$ is strictly less than Wilkinson's bound. We have

$$F(x) = \frac{1}{x}\int_0^x q_{\lceil t \rceil} - q_1 \, \mathrm{d}t = \frac{x - \lfloor x \rfloor}{x}(q_{\lceil x \rceil} - q_1) + \frac{1}{x}\sum_{k=1}^{\lfloor x \rfloor}(q_k - q_1).$$

By Inequalities 1.2 and 2.5,

$$q_1 - q_{\lceil x \rceil} \leq \frac{\ln^2 \lceil x \rceil}{4} + \frac{\ln \lceil x \rceil}{2} + \ln 2 \qquad \text{and} \qquad \frac{1}{\lfloor x \rfloor}\sum_{k=1}^{\lfloor x \rfloor}(q_1 - q_k) \leq \frac{\ln^2 \lfloor x \rfloor}{4} + \ln 2.$$

Altogether, we obtain the lower bound

$$\begin{aligned} F(x) &\geq -\frac{1}{x}\left(\frac{\ln^2 \lceil x \rceil}{4} + \frac{\ln \lceil x \rceil}{2} + \ln 2\right) - \left(\frac{\ln^2 \lfloor x \rfloor}{4} + \ln 2\right) \\ &\geq -\frac{1}{x}\left(\frac{(\ln x + \frac{1}{x})^2}{4} + \frac{\ln x + \frac{1}{x}}{2} + \ln 2\right) - \left(\frac{\ln^2 x}{4} + \ln 2\right). \end{aligned}$$

By inspection, the right-hand side of the above inequality is strictly greater than $-(\alpha \ln^2 x + \beta \ln x)$ for our interval of interest $x \in [100, 1700]$.

4.4. **Proof of Lemma 4.1: Inductive Step.** In order to verify the claim for some $y > 1700$, we integrate over $x \in \left[\frac{y}{2}, \frac{y}{\sqrt{2}}\right]$ to obtain a lower bound for $F(y)$ in terms of $F(x)$ for $x < y$. In particular, by integrating Inequality 4.1 over $x \in \left[\frac{y}{2}, \frac{y}{\sqrt{2}}\right]$ we have

$$\begin{aligned} \frac{1}{\frac{y}{\sqrt{2}} - \frac{y}{2}}\int_{\frac{y}{2}}^{\frac{y}{\sqrt{2}}} F(\lceil x \rceil)\,\mathrm{d}x &\leq \frac{1}{\frac{y}{\sqrt{2}} - \frac{y}{2}}\left[\left(\sqrt{2} - 1 - \frac{2\sqrt{2}}{y}\right)\int_{\frac{y}{2}}^{y} f(x)\,\mathrm{d}x + \int_{\frac{y}{2}}^{\frac{y}{\sqrt{2}}} \frac{\ln(\frac{11}{4}x)}{2} + \frac{1}{2x}\,\mathrm{d}x\right] \\ &= \left(1 - \frac{4 + 2\sqrt{2}}{y}\right)\left(2F(y) - F(\tfrac{y}{2})\right) + \frac{\ln y}{2} + \frac{\ln 2 + \sqrt{2}\ln\frac{11}{4} - \sqrt{2}}{2\sqrt{2}} + \frac{(\sqrt{2}+1)\ln 2}{2y}. \end{aligned}$$

Rearranging the above inequality allows us to lower bound $F(y)$ by a positive linear combination of $F(x)$ for $x \in \left[\frac{y}{2}, \frac{y}{\sqrt{2}}\right]$. We note that this is the reason for the choice of $k + \ell \approx \sqrt{2}k$, as this approach does not give us such a bound if $\sqrt{2}$ is replaced by a larger constant. Now, suppose our claim is false, and let $y > 1700$ be the smallest value such that $F(y) \leq -\alpha \ln^2 y - \beta \ln y$. We aim to show that this contradicts the above lower bound for $F(y)$. By assumption,

$$\begin{aligned} F(\lceil x \rceil) &> -\alpha \ln^2(x+1) - \beta \ln(x+1) \\ &> -\alpha \ln^2 x - \beta \ln x - \frac{2\alpha \ln x}{x} - \frac{\beta}{x} - \frac{\alpha}{x^2} \qquad \text{for } x \in \left[\tfrac{y}{2}, \tfrac{y}{\sqrt{2}}\right], \end{aligned}$$

implying that

$$\begin{aligned} \frac{1}{\frac{y}{\sqrt{2}} - \frac{y}{2}}\int_{\frac{y}{2}}^{\frac{y}{\sqrt{2}}} F(\lceil x \rceil)\,\mathrm{d}x > -\alpha \ln^2 y - \left((\sqrt{2}\ln 2 - 2)\alpha + \beta\right)\ln y - \left(2 - \frac{(3+\sqrt{2})\ln^2 2}{2\sqrt{2}} - \sqrt{2}\ln 2\right)\alpha \\ - \left(\frac{\ln 2}{\sqrt{2}} - 1\right)\beta - \frac{2(\sqrt{2}+1)\alpha \ln 2 \ln y}{y} - \frac{(\sqrt{2}+1)(\beta \ln 2 - \frac{3}{2}\alpha \ln^2 2)}{y} - \frac{2\sqrt{2}\alpha}{y^2}. \end{aligned}$$

In addition,

$$2F(y) - F(\tfrac{y}{2}) < -\alpha \ln^2 y - (2\alpha \ln 2 + \beta)\ln y + \alpha \ln^2 2 - \beta \ln 2.$$

Combining our upper and lower bounds, we observe that the terms containing $\ln^2 y$ are equal, and the terms containing $\ln y$ are equal

$$-\big((\sqrt{2}\ln 2 - 2)\alpha + \beta\big) = \frac{1}{2} - (2\alpha \ln 2 + \beta)$$

due to the value of α. We are left with the inequality

$$\frac{(\sqrt{2}-1)\ln 2 + \sqrt{2}}{\sqrt{2}}\beta + \frac{(2-\sqrt{2})\ln^2 2 - 4(2-\sqrt{2})(\ln\frac{11}{4} - 1)\ln 2 - 8\ln\frac{11}{4}}{8(2+(2-\sqrt{2})\ln 2)} + g(\beta, y) < 0,$$

where $g(\beta, y)$ is a linear function of β of order $O(\ln^2(y)/y)$. The left-hand side is strictly greater than zero for a sufficiently large choice of β. However, verifying that our choice of $\beta = 0.41$ is sufficient requires an explicit analysis of $g(\beta, y)$ for $\beta = 0.41$ and $y > 1700$. The function $g(\beta, y)$ is given by

$$\begin{aligned} g(\beta, y) = &-\frac{2+\sqrt{2}}{2+(2-\sqrt{2})\ln 2}\,\frac{\ln^2 y}{y} - \left((4+2\sqrt{2})\beta + \frac{(5+3\sqrt{2})\ln 2}{2+(2-\sqrt{2})\ln 2}\right)\frac{\ln y}{y} \\ &+ \left(\frac{(11+7\sqrt{2})\ln^2 2}{4(2+(2-\sqrt{2})\ln 2)} - (5+3\sqrt{2})\beta\ln 2 - \frac{(\sqrt{2}+1)\ln 2}{2}\right)\frac{1}{y} - \frac{\sqrt{2}}{2+(2-\sqrt{2})\ln 2}\,\frac{1}{y^2}. \end{aligned}$$

When $\beta = 0.41$ and $y > 1700$,

$$\frac{(\sqrt{2}-1)\ln 2 + \sqrt{2}}{\sqrt{2}}\beta + \frac{(2-\sqrt{2})\ln^2 2 - 4(2-\sqrt{2})(\ln\frac{11}{4} - 1)\ln 2 - 8\ln\frac{11}{4}}{8(2+(2-\sqrt{2})\ln 2)} > 0.086$$

and

$$g(0.41, y) > -\frac{\frac{3}{2}\ln^2 y}{y} - \frac{6\ln y}{y} - \frac{3}{y} - \frac{1}{y^2} > -\frac{\frac{3}{2}\ln^2 1700}{1700} - \frac{6\ln 1700}{1700} - \frac{3}{1700} - \frac{1}{1700^2} > -0.08,$$

thus obtaining our desired contradiction. This completes the proof of Theorem 1.1.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant no. OAC-1835443, grant no. SII-2029670, grant no. ECCS-2029670, grant no. OAC-2103804, and grant no. PHY-2021825. We also gratefully acknowledge the U.S. Agency for International Development through Penn State for grant no. S002283-USAID. The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001211 and DE-AR0001222. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. This material was supported by The Research Council of Norway and Equinor ASA through Research Council project "308817 - Digital wells for optimal production and drainage". Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The third author thanks Mehtaab Sawhney for interesting conversions regarding linear programming. The authors thank Louisa Thomas for improving the style of presentation.

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
[2] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.
[3] AM Cohen. A note on pivot size in Gaussian elimination. *Linear Algebra and its Applications*, 8(4):361–368, 1974.
[4] Colin W Cryer. Pivot size in Gaussian elimination. *Numerische Mathematik*, 12(4):335–345, 1968.
[5] Jane Day and Brian Peterson. Growth in Gaussian elimination. *The American mathematical monthly*, 95(6):489–513, 1988.
[6] James Demmel. Accurate svds of structured matrices. Technical report, Citeseer, 1997.
[7] Alan Edelman. The complete pivoting conjecture for Gaussian elimination is false. *The Mathematica Journal*, 2(2):58–61, 1992.
[8] Alan Edelman and John Urschel. Some new results on the maximum growth factor in gaussian elimination. *SIAM Journal on Matrix Analysis and Applications*, 45(2):967–991, 2024.
[9] Leslie V. Foster. Gaussian elimination with partial pivoting can fail in practice. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1354–1362, 1994.
[10] Herman H Goldstine. *The computer from Pascal to von Neumann.* Princeton University Press, 1993.
[11] Gene H Golub and Charles F Van Loan. *Matrix computations.* JHU press, 2013.
[12] Nick Gould. On growth in Gaussian elimination with complete pivoting. *SIAM Journal on Matrix Analysis and Applications*, 12(2):354–361, 1991.
[13] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.
[14] Nicholas J Higham. *Accuracy and stability of numerical algorithms.* SIAM, 2002.

[15] Chi-Kwong Li and Roy Mathias. The determinant of the sum of two matrices. *Bulletin of the Australian Mathematical Society*, 52(3):425–429, 1995.

[16] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023.

[17] Carl Meyer. History of Gaussian elimination. `http://carlmeyer.com/pdfFiles/GaussianEliminationHistory.pdf`.

[18] John Peca-Medlin and Thomas Trogdon. Growth factors of random butterfly matrices and the stability of avoiding pivoting. *SIAM Journal on Matrix Analysis and Applications*, 44(3):945–970, 2023.

[19] Arvind Sankar. *Smoothed analysis of Gaussian elimination*. PhD thesis, Massachusetts Institute of Technology, 2004.

[20] Arvind Sankar, Daniel A Spielman, and Shang-Hua Teng. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM Journal on Matrix Analysis and Applications*, 28(2):446–476, 2006.

[21] Leonard Tornheim. Pivot size in Gauss reduction. *Tech Report, Chevron Research Co., Richmond CA*, 1964.

[22] Leonard Tornheim. Maximum third pivot for Gaussian reduction. In *Tech. Report*. Calif. Res. Corp Richmond, Calif, 1965.

[23] Leonard Tornheim. A bound for the fifth pivot in Gaussian elimination. *Tech Report, Chevron Research Co., Richmond CA*, 1969.

[24] Leonard Tornheim. Maximum pivot size in Gaussian elimination with complete pivoting. *Tech Report, Chevron Research Co., Richmond CA*, 10, 1970.

[25] John von Neumann and Herman H Goldstine. Numerical inverting of matrices of high order. 1947.

[26] James Hardy Wilkinson. Error analysis of direct methods of matrix inversion. *Journal of the ACM (JACM)*, 8(3):281–330, 1961.

[27] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.

[28] Stephen J Wright. A collection of problems for which Gaussian elimination with partial pivoting is unstable. *SIAM Journal on Scientific Computing*, 14(1):231–238, 1993.

Appendix A. The modern role of partial and complete pivoting in computation

This appendix reviews the role of the growth factor in applied computation and demonstrates the modern practical importance of both partial and complete pivoting.

A.1. **The growth factor and error estimates for solving linear systems.** Gaussian elimination can be used to solve a linear system $Ax = b$ by factoring $A = LU$ into the product of a lower triangular and upper triangular matrix L and U. Given the factorization $A = LU$, the linear system $Ax = b$ is mathematically equivalent to $LUx = b$, which can be efficiently and accurately solved using forward and backward substitution. This procedure, when performed in floating point arithmetic with either partial or complete pivoting, produces an approximate solution $\hat{x}$ satisfying

$$(A + \Delta A)\hat{x} = b, \qquad \|\Delta A\|_{\max} \leq \tfrac{2}{1-nu} n^3 \mathrm{u}\, \rho(A),$$

where n is the dimension of the matrix, u is the unit roundoff of the floating point arithmetic, and $\rho(A)$ is the growth factor of A in floating point arithmetic under pivoting (the same bound holds for both partial and complete pivoting); see [14, pgs. 175-177] and other related formulas for details. For this reason, understanding the growth factor is of both great practical and theoretical importance. Further details regarding the long history of research in this area can be found in [8], though we also draw special attention to the modern interest in smoothed analysis (the study of algorithms under small random perturbation to input) for Gaussian elimination [19, 20].

A.2. **Is large growth for partial pivoting as rare now as it seemed in years past?** In the classic text *The Algebraic Eigenvalue Problem* [27, pg. 212], Wilkinson showed that Gaussian elimination with partial pivoting was unstable in the worst case. He proved that, for partial pivoting, the growth factor is bounded above by 2^{n-1}, and that this exponential upper bound can be achieved by the matrix

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & 1 & 0 & 1 \\ -1 & \cdots & -1 & 1 & 1 \\ -1 & \cdots & -1 & -1 & 1 \end{pmatrix}. \tag{A.1}$$

This exponentially large growth factor can lead to catastrophic errors, even for well-conditioned matrices. Despite this, Gaussian elimination with partial pivoting remains the premier technique used to solve a general linear system $Ax = b$ computationally. The backslash "\" command in MATLAB and Julia, and the linalg.solve() command in Python, all employ Gaussian elimination with partial pivoting by calling the same LAPACK routines when faced with an arbitrary square matrix (different algorithms may be used when the input matrix has special structure). However, the fundamental issues originally noted by Wilkinson in 1965 still persist. In Figure 2, we attempt to solve a 100×100 linear system involving the matrix A of Equation A.1 using the built-in backslash "\" command in the Julia programming language. Shockingly, the algorithm produces a solution with almost no correct significant digits, and no error message is output (see [18, Subsection 2.2] for further experiments). This is unrelated to condition number, as the condition number of A when $n = 100$ is only 45. Many researchers suspect that the situations where Gaussian elimination with partial pivoting fails (such as Figure 2) are rare. For instance, in the widely used textbook *Matrix Computations* by Gene Golub and Charles Van Loan, the authors explicitly discuss the worst-case stability of Gaussian elimination with partial pivoting:

> Although there is still more to understand about ρ [the growth factor], the consensus is that serious element growth in Gaussian elimination with partial pivoting is *extremely* rare. *The method can be used with confidence* [11, pg. 131].

This sentiment may have been partially true in 1983, when the first edition of *Matrix Computations* was released, but it is simply not true today. This could even be considered an example of normalcy bias, the refusal to plan for a disaster which has never happened before. We contend that there has been an exponential increase in both the quantity and types of linear systems solved today as compared to when that word "rare" was typed (likely on an IBM Selectric Typewriter) in the years before 1983. Furthermore, the current software environments and hardware platforms were likely unimaginable then, especially when considering the 1977 (mis)quote that nobody would ever need a computer in their home.

In the early days of numerical linear algebra, the source of most numerical analysis problems were discretizations of integral and differential equations. The matrix A above and matrices with similarly large growth may seem unlikely to occur in practice when considering typical problems of classical numerical analysis. Wright [28] and Foster [9] found examples of two-point boundary value problems

```
# Wilkinson's famous matrix (in Julia)
# Matrices like wilk(n) should no longer be considered rare
# (Subsection A.2)
wilk(n) = [ i>j ? -1 : (i==j || j==n) ? 1 : 0 for i=1:n, j=1:n]

# Demonstrating the inaccuracy of GE with partial pivoting
n = 100;
A = wilk(n);
x = randn(n); b=A*x;
[A\b x][45:55,:] # interesting middle elements

#   A\b    vs     x
  0.369141   0.370758
 -0.789062  -0.787242
 -0.835938  -0.82985
  0.78125    0.790538
 -0.875     -0.86182
  0.5625     0.595671
 -3.0625    -2.99157
  1.25       1.33367
  0.0        0.302893
  1.0        1.52312
 -1.0        0.21591
```

FIGURE 2. The potential inaccuracies of partial pivoting. Surprisingly, the computed solution barely has any correct significant digits! Observing the middle elements of the exact and computed solution, one can almost feel the bits being chopped off at the end. This is not caused by the condition number, as cond(A) is only 45. No warning or error message is given.

and Volterra integral equations with large growth that at least had the appearance of a classical numerical analysis problem. Still, it was easy to argue that even these examples were somewhat contrived. Nonetheless, given the aforementioned increase in the quantity of linear systems solved, we would not rule out large growth even amongst classical-looking numerical analysis problems.

The strongest concern, however, arises from the fact that the types of problems solved today are much more varied. The matrix A above (and similar-looking matrices with exponential growth) have a high degree of symmetry and a simple combinatorial structure. The field of discrete mathematics has grown dramatically in recent decades, and with it the need to solve linear systems arising from network structures. As a result, such matrices no longer seem so unlikely to occur in practice[3].

A.3. **Consequences of modern trends in computing.** Not only is the solution of $Ax = b$ more frequent and more varied, but the problem sizes are larger and the types of hardware being used are more varied as well. It is not unusual to take advantage of hardware accelerators such as graphical processing units (GPUs), which run at lightning speed in half precision. These two factors put further pressure on the accuracy of Gaussian elimination with partial pivoting.

A.4. **Has complete pivoting software really been out there?** While LAPACK's partial pivoting routine `getrf` and variants are the workhorse under the hood of Julia, MATLAB, and Python, LAPACK[4] has a complete pivoting routine available `getc2`, and complete pivoting is also available, though arguably less prominently, in other languages as native code (e.g., on MATLAB Central or in

[3]The third author thanks Avi Widgerson for emphasizing this point during a tutorial on Gaussian elimination presented at the Institute for Advanced Study.

[4]Jim Demmel, one of the lead authors of the LAPACK library, opted for complete pivoting over partial pivoting in his analysis of high accuracy singular value decompositions [6].

the Julia package DLA.jl). Soon, complete pivoting will become even more accessible in Julia. The implication that a lack of accessibility of complete pivoting in software is to be equated with a lack of user interest is an example of confirmation bias: a software writer that has solvers embedded in a popular package used by many people may choose the extra safety of complete pivoting if it were more accessible.

Overall, our conclusion is that stable alternatives to naive partial pivoting are needed and, therefore, a deeper mathematical understanding of both complete and partial pivoting is as important as ever.