# Alan Turing and the origins of modern Gaussian elimination*

Froilán M. Dopico

December 14, 2012

Instituto de Ciencias Matemáticas CSIC-UAM-UC3M-UCM and
Departamento de Matemáticas, Universidad Carlos III de Madrid,
Avda. Universidad 30, 28911 Leganés, Spain. (dopico@math.uc3m.es).

**Abstract**

The solution of a system of linear equations is by far the most important problem in Applied Mathematics. It is important both in itself and because it is an intermediate step in many other relevant problems. Gaussian elimination is nowadays the standard method for solving this problem numerically on a computer and it was the first numerical algorithm to be subjected to rounding error analysis. In 1948, Alan Turing published a remarkable paper on this topic: "Rounding-off errors in matrix processes" (Quart. J. Mech. Appl. Math. 1, pp. 287-308). In this paper, Turing formulated Gaussian elimination as the matrix LU factorization and introduced the "condition number of a matrix", both of them fundamental notions of modern Numerical Analysis. In addition, Turing presented an error analysis of Gaussian elimination for general matrices that deeply influenced the spirit of the definitive analysis developed by James Wilkinson in 1961. Alan Turing's work on Gaussian elimination appears in a fascinating period for modern Numerical Analysis. Other giants of Mathematics, as John Von Neumann, Herman Goldstine, and Harold Hotelling were also working in the mid-1940s on Gaussian elimination. The goal of these researchers was to find an efficient and reliable method to solve systems of linear equations in modern "automatic computers". At that time, it was not clear at all whether Gaussian elimination was a right choice or not. The purpose of this paper is to revise, at an introductory level, the contributions of Alan Turing and other authors to the error analysis of Gaussian elimination, the historical context of these contributions, and their influence on modern Numerical Analysis.

## 1 Introduction

Alan Turing made several contributions that are considered fundamental in Mathematics and Computer Science and that are widely known by all mathematicians and computer scientists. Even more, the names of some of these contributions are also very well known by many educated (but not specialists) people as, for instance, the name *Turing Machine* or the name *Enigma*. In addition, Alan Turing made other fundamental contributions that remain almost unknown for most mathematicians and computer scientists and, of course, completely unknown outside the academic world. One of these contributions is Alan Turing's work on the error analysis of the method of *Gaussian Elimination* (GE) for solving systems of linear equations, *which is one of the most important and ubiquitous numerical algorithms* and, perhaps, the most, since it is used by many other numerical algorithms. Curiously enough, basic versions of GE are explained in high school courses of Mathematics and, therefore, GE is one of the best known algorithms by common people, but most professional mathematicians and computer scientists are unaware of its relationship with Alan Turing's scientific contributions.

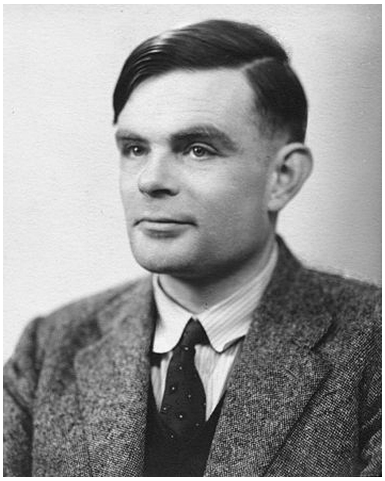
Many numerical analysts know that Alan Turing was one of the first researchers working on the error analysis of GE. This is clearly explained in some standard references on Numerical Analysis. In particular, an excellent text that gives a detailed account on Turing's contributions to the analysis of GE is Nicholas Higham's *"Accuracy and Stability of Numerical Algorithms"* [12]. Not incidentally, Nicholas Higham is "Richardson Professor" of Applied Mathematics in the School of Mathematics at Alan Turing Building in The University of Manchester, precisely the institution where Alan Turing spent the last part (1948-1954) of his short life, and [12] is dedicated

*This work was partially supported by the Ministerio de Economía y Competitividad of Spain through grant MTM-2009-09281 and was originated by a talk with the same title presented in the International Symposium *"The Alan Turing Legacy"* held in Madrid (Spain) in October 23-24, 2012. This symposium was organized and funded by the *Real Academia de Ciencias Exactas, Físicas y Naturales* of Spain and *Fundación Ramón Areces*.

to Alan Turing and James Wilkinson, who will be another important character in our story (see Figure 1). Concerning Alan Turing's work on GE, one can read the following paragraph in [12, pp. 184-185]:

> *"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper "Rounding-off errors in matrix processes" [20]. In this paper, Turing made several important contributions. He formulated the LU factorization of a matrix ... showing that Gaussian elimination computes an LU factorization. He introduced the term "condition number" and defined two matrix condition numbers ... He exploited backward error ideas ... Finally, and perhaps most importantly, he analysed Gaussian elimination with partial pivoting for general matrices and obtained a bound for (the error) ..."*

I am not mentioning above all the contributions of Turing listed by N. Higham in [12], but only those that I will consider in this manuscript because, in my opinion, they are the most interesting for a general audience.



Alan Turing (1912-1954)

James Wilkinson (1919-1986)

Figure 1: Alan Turing on the left and James Wilkinson on the right.

Probably most mathematicians and computer scientists, and for sure most common people, are unaware that Alan Turing was not the only great mathematician working on the error analysis of GE in the 1940's. However, for numerical analysts it is well known that, before him, other giants of Mathematics considered the error analysis of GE as a very important problem and worked on it in the 1940's. In fact, although Alan Turing certainly did a number of key and original contributions in [20], some of the results presented in [20] were previously known or were closely related to previous work by other authors. This has been pointed out in the complete recent survey *"John von Neumann's Analysis of Gaussian Elimination and the Origins of Modern Numerical Analysis"*[1] by Joseph Grcar [9], where one can find the following [9, p. 633]:

> *"Turing coined the name "condition number" ... for measures of sensitivity of problems to error, and he acronym "LU" for the general decomposition. Textbooks tend to intimate that Turing introduced modern concepts by introducing the modern nomenclature, but the history is more complex. Algorithms had been described with matrix decompositions before Turing's paper ... Measures of sensitivity evolved from as early as Wittmeyer in the 1930s ..."*

In this context, the main goal of this manuscript is to bring to the attention of a "widest as possible" audience the work of Alan Turing on GE and to explain why this problem was (*"is"*) so important in Numerical Analysis in particular, and in Mathematics in general. For this purpose, I aim to explain at an introductory level, accessible to readers with a basic background in Mathematics (the level of a last *high school course*), the most important ideas included in [20] and their role in modern Numerical Analysis. I also want to briefly describe the fascinating historical period in which Alan Turing's paper [20] was written and published, as well as the work made by other very relevant researchers (Hotelling, von Neumann, Goldstine, Wilkinson) on the rounding error analysis of GE *before and after* Turing's paper. I will stress *the unique spirit of Alan Turing's approach* to

[1] The title of [9] and the present paper are rather similar and this is not by chance!!

the problem and its influence on modern Numerical Analysis. In my opinion, this spirit reflects very well the genius of Turing and establishes a difference between his work and the work by others. Finally, I will discuss a couple of very recent developments on error analysis of GE and the main problem still open on this topic.

Before starting, let me say a few words about what this paper is not. *It is not a rigorous* mathematical paper, since Numerical Analysis is a branch of Mathematics full of technical details that can hide the main ideas for non-specialist readers. Therefore, I will omit to state many rigorous theorems in the exposition, although I will provide references where interested readers may find complete information. Moreover, *this paper is not a work on the History of Mathematics*. After reading with detail Turing's paper [20] and some recent works on the History of Numerical Analysis, I am convinced that [20] deserves to be analyzed in depth, both from the point of view of Turing's scientific biography and from the point of view of the History of Numerical Analysis. An extensive study in the spirit of the recent paper by Joseph Grcar on von Neumann's contribution to GE [9] is clearly necessary. However, this would lead to a very long paper or to a paper for specialists who already know the error analysis of GE and are interested in its origins and evolution. Therefore, I have chosen to write a paper on modern mathematical results, with modern mathematical notation, and where the history enters in the form of comments and remarks instead as explicit statements of original results from the 1940's.

The paper is organized as follows. In Section 2, a brief history of GE is presented and the classic and modern descriptions of GE are refreshed for those readers who have forgotten GE or who are not familiar with its modern treatment. Section 3 describes the historical context, from the point of view of Mathematics and Computer Science, in which the paper [20] was published. Since the title of [20] is *"Rounding-off errors in matrix processes"*, it is essential to describe in Section 4 in simple terms which are the errors committed by GE when it is run on a computer. This will allow us to understand why this problem is so interesting and difficult and to understand why a complete solution still remains as an open problem. The error analysis of GE currently accepted was not developed in the 1940's. It was developed by James Wilkinson in 1961 [24]. Therefore, we discuss in Section 5 some key points about what Alan Turing did and did not in [20]. It is important to note that rounding error analysis of GE is still an active area of research and some recent works in this area are briefly described in Section 6. Finally, some conclusions are presented in Section 7.

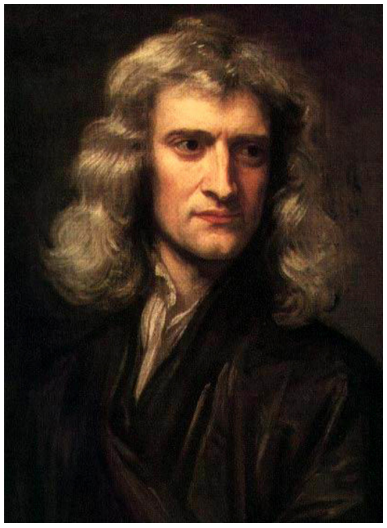
## 2 A brief history and description of Gaussian elimination

Classic books on the History of Mathematics, as well as recent studies on this subject, place the origins of GE in a variety of ancient texts from different places and times: China, Greece, Rome, India, medieval Arabic countries, and European Renaissance. However, in my opinion, it is not exact to say that these ancient/medieval/renaissence texts describe what we understand today as the *method of GE*, since these texts mainly present some specific problems that are solved in a way that fits in what today is accepted as GE, but they do not include any explicit statement of the set of rules that constitute the method of GE. In this context, I refer the reader to the excellent recent papers [8, 10] for a detailed account of the History of GE (including many interesting technical details) and of the researchers who contributed to its development. Here, for the sake of brevity and simplicity, I will only highlight the most important contributions and contributors.

The developments of GE that include explicit statements of algorithmic rules can be organized essentially in three periods [10] that are called the *schoolbook elimination* period, the *professional elimination* period, and the *modern elimination* period.

The *schoolbook elimination* period corresponds to the development of GE essentially as it is presented in current high school textbooks. This period started with Isaac Newton (see Figure 2), who lectured on Algebra as it appeared in Renaissance texts while working for his promotion to the Lucasian professorship. In 1669-1670 Newton wrote some notes where he established the systematic rules for solving systems of linear equations via the *extermination* (today *elimination*) method [8]. Taking into account the extremely powerful and systematic mind of Newton, I conjecture that he was not satisfied with the unsystematic way in which Renaissance Algebra texts described the solution of linear systems of equations and that this motivated him to write his notes. These notes remained unpublished until they were published in Latin in 1707 and in English in 1720. The clarity of these notes, as well as the immense prestige of Newton, motivated that many Algebra textbooks in the eighteenth century presented the solution of systems of linear equations by following essentially Newton's rules. We only mention here the very well-presented text *"Elemens d'algèbre"* (Paris, 5th ed., 1804) by Sylvestre Lacroix, where the modern word *elimination* was used for the first time instead of *extermination*.

The way GE is presented in high school textbooks is highly inefficient for solving moderately large systems



Isaac Newton (1643-1727)

Carl Friedrich Gauss (1777-1855)

Figure 2: Isaac Newton established first the rules of Gaussian elimination as they are still presented in current high school textbooks. Carl Friedrich Gauss developed efficient methods for solving normal equations, i.e., the special type of linear systems arising in the solution of least squares problems, via *Gaussian* elimination.

of linear equations via *hand* computations. This was not a problem for some time, because large systems of linear equations did not arise in relevant real-world applications. This abruptly changed with the invention of the *method of least squares* by Adrien-Marie Legrendre (1805) and Carl Friedrich Gauss (1809) (or by Gauss and Legendre in reverse order!) at the beginning of the nineteenth century.

The *method of least squares* answered the question of how to make accurate predictions from measurements with errors, a question that was motivated by practical measurements in astronomy and, more important in real-life applications, by geodetic research for cartography, an activity that was generously funded by governments in the nineteenth century. The least squares method finds a minimum of a certain quadratic function of many variables, but the important point for our story is that *this minimum is the solution of a linear systems of equations* that are called the *normal equations*. These systems of equations are very particular since, in modern nomenclature, their coefficient matrices are symmetric and positive definite. At the time of Gauss, normal equations might have as much as 20 equations and 20 unknowns and this was a formidable task for *professional human computers* if the elimination method was applied as described by Newton to compute the solution.

These difficulties motivated Gauss to modify the high school elimination method of Newton in a nontrivial way and this is the start of the *professional elimination* period of GE. The details are too technical to be explained here (see [8]), but the key point of Gauss's method is to avoid to write symbolic algebraic equations and unknowns. By the use of a clever notation, Gauss computations were stored in lists of numbers. In addition, he halved the number of needed arithmetic operations with respect the high school elimination method by taking into account the symmetry of normal equations. Gauss's method does not superficially resemble neither high school elimination method nor modern GE, but it was very important from the point of view of applications and it became part of the syllabus of geodesists, cartographers, and military engineers.

Gauss's method was significatively improved by Myrick Doolittle (1881), André-Louis Cholesky (1924), who adapted it for being used with mechanical multiplying calculators, and Prescott Durand Crout (1941), who developed a method valid for general systems of equations and not only for normal equations. This essentially closes the *professional elimination* period of GE, since modern computers came into scene in the next few years.

The *modern elimination* period of GE started in 1947 with the key paper [21], by John von Neumann and Herman Goldstine, and continued one year later with the paper [20], by Alan Turing, that motivates this manuscript. These authors considered implementations of GE with the aim of being used on *digital, electronic, and programmable computers*, i.e., modern computers. The motivation was not just to get an efficient implementation, but also a *guaranteed and reliable implementation from the point of view of the rounding errors* committed by modern computers. This required the development both of algorithmic improvements and of error analyses of GE. The interest on error analysis represents a fundamental difference with respect the activity in previous periods. The definitive error analysis of GE accepted today was presented by James Wilkinson in [24].

The contents of the references mentioned in this paragraph will be described in more detail in next sections.

It is important to observe that, since the 1940's, the research on different aspects of GE has remained, and still remains, very active. The interested reader is invited to consult the wide collection of references included in [12, Chapters 8-14], as well as, the recent, complete, and easy-to-read review [13]. It may be also interesting to know that during the early stages of the modern elimination period GE took the name *"Gaussian"* (before, it was known simply as the *"elimination method"*), apparently as a consequence of misattributing high school elimination to Gauss instead of Newton. More precisely, Turing writes *"...Gauss's elimination method. This is the method almost universally taught in schools..."* in the first page of [20] and it seems that George Forsythe was the fist to call it *"Gaussian elimination"* in 1953 [8, 10].

## 2.1 Refreshing Gaussian elimination from high school with Newton

In this section, I refresh the method of GE as it appears in high school textbooks via an example. Later, I will use the same example to illustrate how modern GE is presented in Numerical Analysis textbooks at the University-level. So, I propose the readers to imagine themselves again very young, living the good times of high school, and, for making this exercise even more exciting, the readers may think that Newton is their teacher!

Consider that we are asked to solve the following system of equations.

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\ -4x_1 & - & 9x_2 & + & 3x_3 & + & 2x_4 & = & -15 \\ 6x_1 & + & 21x_2 & - & 3x_3 & - & 11x_4 & = & 23 \\ 2x_1 & - & 3x_2 & - & 27x_3 & - & 3x_4 & = & -37 \end{array} . \tag{1}$$

The key point of *Gaussian elimination* is to *eliminate* unknowns from certain equations. For describing the method of GE in a precise way, we number the equations in (1) from top to bottom, i.e., the top equation is equation(1) and the bottom equation is equation(4). In a *first stage*, we eliminate $x_1$ in all equations below the first one via the following *replacement operations*: replace "equation(2)" by "equation(2) $-$ $(-2)\times$equation(1)"; replace "equation(3)" by "equation(3) $-$ $3\times$equation(1)"; and replace "equation(4)" by "equation(4) $-$ $1\times$equation(1)". So, we obtain the linear system

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\ & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\ & & 12x_2 & & & - & 14x_4 & = & -4 \\ & - & 6x_2 & - & 26x_3 & - & 4x_4 & = & -46 \end{array} .$$

Next, we perform the *second stage*, where we eliminate $x_2$ in all equations below the second one via the replacement operations: replace "equation(3)" by "equation(3) $-$ $(-4)\times$equation(2)"; and replace "equation(4)" by "equation(4) $-$ $2\times$equation(2)". So, we obtain the linear system

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\ & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\ & & & & 4x_3 & + & 2x_4 & = & 8 \\ & & & - & 28x_3 & - & 12x_4 & = & -52 \end{array} .$$

Finally, we perform the *third stage*, where we eliminate $x_3$ in all equations below the third one via the replacement operation: replace "equation(4)" by "equation(4) $-$ $(-7)\times$equation(3)". This leads to the following *upper triangular* linear system

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\ & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\ & & & & 4x_3 & + & 2x_4 & = & 8 \\ & & & & & & 2x_4 & = & 4 \end{array} . \tag{2}$$

This in the end of the GE process that has transformed the linear system (1), which we did not know how to solve it, into the linear system (2), which has the same solution and that can be solved very easily: from equation(4) we compute $x_4$; next from equation(3) we compute $x_3$; next from equation(2) we compute $x_2$; and, finally, from equation(1) we compute $x_1$. This procedure of solving (2) is known as *backward substitution*, and

it computes the following solution:

$$\begin{array}{rcrcrcrcl} 2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \longrightarrow x_1 = 1 \\ & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \longrightarrow x_2 = 2 \\ & & & & 4x_3 & + & 2x_4 & = & 8 \longrightarrow x_3 = 1 \\ & & & & & & 2x_4 & = & 4 \longrightarrow x_4 = 2 \end{array} . \tag{3}$$

The process above can be easily generalized to systems with any number of equations and unknowns.

The linear system (1) has some key features that have to be commented. First note that (1) has the same numbers of equations as unknowns and that its solution is unique. In more advanced mathematical language, this is equivalent to say that *the coefficient matrix of* (1) *is nonsingular. This is the only case in which GE is used on modern computers and is the only case that is considered in this paper*[2]. Probably, many readers recall from their time in high school that GE was also used for linear systems with any number of equations and unknowns for determining whether they have solution or not, and/or, in the case they have, to find a parametric description of the infinite number of solutions. However, in these cases, GE is not reliable from a numerical point of view and other methods are used in actual numerical computations [3, 7, 19].

Another feature of (1) is that GE has run without *interchanging equations*. Interchanges of equations are needed, for instance, if after the second stage $x_3$ does not appear in equation(3). I will discuss later how equations are interchanged when GE is currently implemented on computers. Finally, the readers might recall that in high school they used, in addition to replacement and interchange operations, *scaling* of equations, i.e., to multiply an equation by a nonzero number. Scaling operations are never used in modern GE.

I am almost sure that most readers, apart from many happy memories in high school, have also recalled that to perform GE *by hand* is a long and boring process and that it is easy to commit errors that spoil the whole solution. An important point to be noted is that, although GE, as explained by Newton, is very efficient from the point of view of the number of arithmetic operations, it requires to write several systems of equations. In our toy example (1), we have written just 4, but for solving a system of 20 equations with 20 unknowns, we should write 20 large systems! This makes Newton's high school elimination very inefficient for solving large systems and motivated Gauss to developed his nontrivial *professional elimination method*. We skip the description of this procedure and move directly to the description of modern GE.

## 2.2 From high school to modern GE: the LU Matrix Factorization

The most important mathematical concept of modern GE is the LU matrix factorization. It allows us to state in a compact and elegant matrix language the GE method described above, it plays an important role in the implementation of GE in modern computers, and, finally, it is essential to facilitate the rounding error analysis of the algorithm. To explain the LU factorization, we use again the linear system (1). To begin with, let us write (1) in *matrix* notation as

$$Ax = b, \qquad \text{where} \quad A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}, \; x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \; b = \begin{bmatrix} 9 \\ -15 \\ 23 \\ -37 \end{bmatrix}. \tag{4}$$

The matrix $A$ is called the coefficient matrix of system (1), $x$ the unknown vector, and $b$ the vector of independent terms (since it does not depend on the unknowns). The replacement operations for equations that were performed for transforming (1) into the upper triangular system (2) can be translated into *replacement operations for rows* of the matrix $A$ and by applying them

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \qquad \text{is transformed into} \qquad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \tag{5}$$

where $U$ is the coefficient matrix of the system (2). Note that, at the moment, we are not paying attention to the vector $b$ in (4). Let us collect the information in the paragraphs after (1) and list the row replacement

[2]Readers should note that for systems having the same number of equations as unknowns this is, by far, the most frequent case in practice, since the probability that a square matrix is singular is zero.

operations applied in (5)

$$\begin{array}{lcll} \text{row}(2) & \to & \text{row}(2) & - & (-2) \times \text{row}(1) \\ \text{row}(3) & \to & \text{row}(3) & - & 3 \times \text{row}(1) \\ \text{row}(4) & \to & \text{row}(4) & - & 1 \times \text{row}(1) \\ \text{row}(3) & \to & \text{row}(3) & - & (-4) \times \text{row}(2) \\ \text{row}(4) & \to & \text{row}(4) & - & 2 \times \text{row}(2) \\ \text{row}(4) & \to & \text{row}(4) & - & (-7) \times \text{row}(3) \end{array} . \tag{6}$$

The numbers $-2, 3, 1, -4, 2$, and $-7$ that multiply rows in each row replacement operation in (6) are called the *multipliers of GE* and our next step is to store them in a $4 \times 4$ matrix $L$. The entries where they are stored are easily determined by the two rows involved in each replacement operation in (6): $-2$ is stored in the entry $(2,1)$, 3 is stored in the entry $(3,1)$, 1 is stored in the entry $(4,1)$, and so on. Clearly, the multipliers only fill the entries below the diagonal of $L$. The remaining entries are defined as follows: all diagonal entries are set equal to one and all entries above the diagonal are set equal to zero. In this way we get

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix} . \tag{7}$$

Up to now, the matrix $L$ is nothing else that a table where the multipliers of GE are stored, but from (5) and (7), the reader may check that the following *miracle* happens!!

$$A = LU , \tag{8}$$

which is the very famous *LU factorization* of the matrix $A$. This factorization expresses $A$ as a product of a lower triangular matrix $L$ with 1's on the diagonal times an upper triangular matrix $U$. The LU factorization exists for almost all matrices and was introduced by von Neumann and Goldstine in 1947 in their celebrated paper [21]. It was also considered later by Turing in [20], where its current name LU was introduced. Here L stands for "lower" (triangular) and U for "upper" (triangular). Turing also stated the condition for the existence and uniqueness of the LU factorization in terms of the nonsingularity of the leading principal minors of $A$ [20, p. 289], as it is still stated today in standard texts on matrix computations [3, 7, 12].

I want to remark that we have constructed (8) via the multipliers of GE and the final matrix $U$ obtained by the GE method. Conversely, if a matrix $A$ is constructed as a product of an arbitrary lower triangular matrix $L$ with 1's on the diagonal times an arbitrary upper triangular matrix $U$, then the multipliers of GE applied on $A$ are the lower triangular entries of $L$ and $U$ is the final matrix obtained by GE.

The LU factorization is not the only factorization of a matrix involving triangular factors that is important in Numerical Analysis. In fact, accurate and efficient algorithms for computing different triangular factorizations of matrices were considered among the *top ten* algorithms of the twentieth century [5], since they are widely used in the numerical solution of many applied problems.

## 2.3 Modern GE: Solving linear systems via the LU factorization

Nowadays, the solution of a linear system $Ax = b$, where $A$ is an $n \times n$ matrix, via the LU factorization is performed in three steps:

1. Compute the LU factorization of $A$: $A = LU$.
2. Solve for $y$ the lower triangular linear system $Ly = b$ by *forward substitution*, i.e., start by computing the first unknown $y_1 = b_1$ from the first equation, then use $y_1$ to compute the second unknown $y_2$ from the second equation, then use $y_1, y_2$ to compute the third unknown $y_3$ from the third equation, and so on.
3. Solve for $x$ the upper triangular linear system $Ux = y$ by *backward substitution* as in (3).

It is easy to see that these steps compute the solution, because if we substitute $y$ from the third step into the equation in the second step, then we get $L(Ux) = b$, which is $Ax = b$. This three-step approach was suggested first by Turing in [20, p. 291], together with several other approaches for solving $Ax = b$. However, it is interesting to mention that Turing did not identify the three-step approach as preferred over other options. Today, it is widely recognized that the three-step approach has several important advantages as, for instance,

that it allows us to solve very easily, and almost without extra computational cost, other linear systems $Ax = b'$ with the same coefficient matrix but different right-hand sides (a situation arising often in applications), and that it simplifies considerably the rounding error analysis of solving $Ax = b$ on a computer.

Explicit algorithms for solving the triangular systems $Ly = b$ and $Ux = y$ appearing in the three-step approach can be written very easily and are not discussed here (see the standard references [3, 7, 12]). However, the computation of the LU factorization of $A$ deserves some comments. The first one is that it is only needed to store the strictly lower triangular part of $L$, i.e., the entries below the diagonal, since the remaining entries are known to be zeros below or ones on the diagonal. Analogously, it is only needed to store the upper triangular part of $U$, i.e., the entries above and on the diagonal. Therefore, the nontrivial parts of $L$ and $U$ fit into the original matrix $A$ and this saves storage requirements in computers and allows us to write the elegant and simple Algorithm 1 for computing the LU factorization of a matrix. I do not pretend at this level that average readers understand Algorithm 1. It is not difficult, but it requires some work and familiarity with programming matrix algorithms (see [3, 7, 12]). However, please, trust me! Yes, this simple algorithm computes really the LU factorization! Also, please look carefully at Algorithm 1 before reading my comments below.

**Algorithm 1** (LU factorization of a matrix)
`Input:` *A matrix of size* $n \times n$
`Output:` *L stored in strictly lower triangular part of A*
*U stored in upper triangular part of A*

*for* $k = 1 : n-1$
  *for* $i = k+1 : n$
    $a_{ik} = a_{ik}/a_{kk}$
    *for* $j = k+1 : n$
      $a_{ij} = a_{ij} - a_{ik}a_{kj}$
    *end*
  *end*
*end*

Observe that Algorithm 1 consists only of *two lines* of arithmetic operations and *three for-loops*. *Its simplicity is fascinating*, specially when it is compared with the long explanation process that is required to present GE and the construction of the LU factorization in most textbooks. Although it may not be obvious, note that the outer *for-loop* of Algorithm 1 corresponds to the *"stages"* of GE, i.e., the $k$-th step in the loop corresponds to the operations needed to eliminate (to set to zero) all entries below the diagonal in the $k$th column.

The computational cost of Algorithm 1 is $2n^3/3 + O(n^2)$ arithmetic operations and this is also the cost of the three-step approach for solving $Ax = b$, since the solution of the triangular systems $Ly = b$ and $Ux = y$ costs $2n^2 - n$ arithmetic operations.

## 2.4 Modern GE: Partial pivoting

Algorithm 1 may produce huge errors when it is implemented on a computer if a very small *pivot* $a_{kk}$ appears in some $k$th stage[3], $k = 1, 2, \ldots, n-1$. In actual computational practice, it is necessary to permute the rows of the matrix $A$ (equivalently, the equations of the system $Ax = b$) for obtaining a reliable algorithm. The permutations are performed "on line" as GE proceeds and several permutation (or pivoting) strategies are described in textbooks on *Numerical Linear Algebra* [3, 7, 12]. However, only one of these strategies is universally adopted in professional software for solving linear systems. This is the *partial pivoting* strategy.

For describing *partial pivoting*, it is convenient to introduce some additional notation. Let us define the matrices $A^{(1)} := A$, $A^{(k)}$ as the matrix produced by GE at the start of the $k$th stage for $k = 2, \ldots, n-1$, and $A^{(n)} := U$ as the upper triangular $U$ factor obtained at the end of GE. The entries of $A^{(k)}$ are denoted by $a_{ij}^{(k)}$, as usual. Recall that the $k$th stage of GE sets to zero the entries below the diagonal in the $k$th column. *Partial pivoting* interchanges at the start of the $k$th stage the $k$th and $r$th rows, where

$$|a_{rk}^{(k)}| := \max_{k \le i \le n} |a_{ik}^{(k)}|,$$

[3] Note that $a_{kk}$ at $k$th stage is not the $(k,k)$ entry of the original matrix since the entries of $A$ are updated by Algorithm 1. Although it is very rare, $a_{kk} = 0$ may happen and, in this case, Algorithm 1 fails.

and after that an standard $k$th stage of GE is performed. To understand better how partial pivoting proceeds, let us apply it to the matrix $A = A^{(1)}$ in (4). Note that the entry with largest absolute value in the first column of $A^{(1)}$ is 6 in position $(3,1)$. Then *partial pivoting exchanges* rows 1 and 3 and after that the replacement operations "row(2) $\to$ row(2) $-(-2/3)\times$row(1)", "row(3) $\to$ row(3) $-(1/3)\times$row(1)", and "row(4) $\to$ row(4) $-(1/3)\times$row(1)" are performed to obtain $A^{(2)}$. This is summarized in the following equation:

$$A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \to \begin{bmatrix} 6 & 21 & -3 & -11 \\ -4 & -9 & 3 & 2 \\ 2 & 3 & -1 & 1 \\ 2 & -3 & -27 & -3 \end{bmatrix} \to A^{(2)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & 5 & 1 & -16/3 \\ 0 & -4 & 0 & 14/3 \\ 0 & -10 & -26 & 2/3 \end{bmatrix}.$$

Next, observe that the entry with largest absolute value in the second column of $A^{(2)}$ *on and below the diagonal* is $-10$ in position $(4,2)$. Then *partial pivoting exchanges* rows 2 and 4 and after that the replacement operations "row(3) $\to$ row(3) $-(2/5)\times$row(2)" and "row(4) $\to$ row(4) $-(-1/2)\times$row(2)" are performed to obtain $A^{(3)}$. This is summarized in the following equation:

$$A^{(2)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & 5 & 1 & -16/3 \\ 0 & -4 & 0 & 14/3 \\ 0 & -10 & -26 & 2/3 \end{bmatrix} \to \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & -4 & 0 & 14/3 \\ 0 & 5 & 1 & -16/3 \end{bmatrix} \to A^{(3)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & 52/5 & 22/5 \\ 0 & 0 & -12 & -5 \end{bmatrix}.$$

Next observe that the entry with largest absolute value in the third column of $A^{(3)}$ *on and below the diagonal* is $-12$ in position $(4,3)$. Then *partial pivoting exchanges* rows 3 and 4 and after that the replacement operation "row(4) $\to$ row(4) $-(-13/15)\times$row(3)" is performed to obtain the upper triangular matrix $A^{(4)} =: U_P$, and the process of GE with *partial pivoting* finishes. This is summarized in the following equation:

$$A^{(3)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & 52/5 & 22/5 \\ 0 & 0 & -12 & -5 \end{bmatrix} \to \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & -12 & -5 \\ 0 & 0 & 52/5 & 22/5 \end{bmatrix} \to U_P = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & -12 & -5 \\ 0 & 0 & 0 & 1/15 \end{bmatrix}.$$

Once the row exchanges that have been done by partial pivoting are known, it is clear that the process is mathematically equivalent to permute $A$ in advance accordingly and then to perform GE without any pivoting. Therefore GE with partial pivoting computes an LU factorization of a matrix $PA = L_P U_P$ that is obtained by exchanging rows 1 and 3 of $A$, after that rows 2 and 4, and, finally rows 3 and 4. We already know the matrix $U_P$ and I propose the reader to deduce from the replacement operations performed above *and taking into account the row interchanges* the lower triangular factor $L_P$. The final factorization is

$$\underbrace{\begin{bmatrix} 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \\ -4 & -9 & 3 & 2 \\ 2 & 3 & -1 & 1 \end{bmatrix}}_{PA} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/3 & 1 & 0 & 0 \\ -2/3 & -1/2 & 1 & 0 \\ 1/3 & 2/5 & -13/15 & 1 \end{bmatrix}}_{L_P} \underbrace{\begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & -12 & -5 \\ 0 & 0 & 0 & 1/15 \end{bmatrix}}_{U_P}. \tag{9}$$

The comparison of the LU factorization of $PA$ in (9) with the one of the original matrix $A$ in (8)-(7)-(5) reveals that row permutations in $A$ induce drastic changes in the LU factors: $L$ and $L_P$ are very different, as well as $U$ and $U_P$. This is an indicator of why rounding errors in GE depend deeply on the pivoting strategy and why the error analysis of GE is extremely difficult. A key observation is that all entries of $L_P$ coming from partial pivoting have absolute values less than or equal to 1, while this does not happen for $L$. This property is of fundamental importance and the reader may find more information on it in [3, 7, 12].

Partial pivoting can be easily and elegantly incorporated in Algorithm 1. The details are omitted, but can be found in [3, 7, 12]. Partial pivoting allows us to write the definitive algorithm of modern Gaussian elimination.

**Algorithm 2** (Modern Gaussian Elimination)
`Input:` *$A$ matrix of size $n \times n$ and $b$ vector of size $n \times 1$*
`Output:` *Solution of linear system $Ax = b$ given as vector $x$ of size $n \times 1$*

1. *Compute the LU factorization of $A$ with partial pivoting: $PA = LU$.*

2. *Solve for $y$ the lower triangular linear system $Ly = Pb$ by* forward substitution.

3. *Solve for $x$ the upper triangular linear system $Ux = y$ by* backward substitution.

The term "partial pivoting" was introduced by Wilkinson in [24], but pivoting techniques were in use in the 1940s and it is not clear who can be said to have invented them.

# 3 Historical context of Alan Turing's paper on rounding errors

In the 1940s there were three very famous papers giving error analyses of GE. The first one was written by Harold Hotelling in 1943 [15]; the second one was written by John von Neumann and Herman Goldstine in 1947 [21]; and the third one is the paper published by Alan Turing in 1948 [20]. There were, of course, other papers published in the 1940's on the same subject, but they have had much less influence than the three papers mentioned above and, therefore, are not considered in this work. Among the three papers [15, 21, 20], the one by Von Neumann and Goldstine in the best known and, without any doubt, the most influential. It is a key paper that has been considered by several top numerical analysts as *the first paper of modern Numerical Analysis*, where *"modern"* has here the sense, already used before, of *"analyzing methods to be used on digital, electronic, programmable computers"*. More information on [21] can be found in [9, 27].

The three papers were written before modern computers existed, but they were motivated by the existence of several projects for constructing the first *"modern computers"* in United Kingdom and USA. In this work, as usual, the term *"modern computer"* should be understood as a *"digital, electronic, and programmable computer"*. To fully realize the context in the 1940s with respect numerical computations, we can imagine ourselves as researchers in the 1940s. Then, it would be clear for us that modern computers would come very soon and that they would offer a huge power of computation compared with that of available desk electro-mechanical calculators. For taking advantage of this "computational giant step", a key question would be to determine whether the numerical methods used in the 1940s and before would be accurate and efficient on modern computers or not. For those problems where a negative answer was obtained, new methods had to be developed.

In the 1940s, as well as today, one of the most important numerical problems was the solution of "large" (the precise meaning of "large" changes continuously with time) systems of linear equations, since they appear in many applications. In addition, in Turing's own words [20, p. 287], *"The best known method for the solution of linear equations is Gauss's elimination method. This is the method almost universally taught in schools."*. Therefore, we can see as very natural for a researcher in the 1940s to study GE from a new perspective: its practical use on modern computers.

The paper by Hotelling was mainly motivated by applications in Statistics. GE was considered in pages 6-7 in [15], where Hotelling presented a very simple error analysis that produces an *error bound that increases exponentially with the number of equations $n$*, more precisely, it increases as $4^{n-1}$. This error bound led Hotelling to state [15, pp. 7-8]: *"The rapidity with which this increases with $n$ is a caution against relying on the results of ... elimination methods ... when the number of equations and unknowns is at all large."* and *"There is here a distinct need of using an iterative process ..."*

Hotelling's results led to general pessimism in mid 1940s about the practical use of GE for solving large systems of equations and motivated the papers by von Neumann and Goldstine [21] and by Turing [20]. In particular we can read in the first page of Turing's paper: *"(GE) has, unfortunately, recently come into disrepute on the ground that rounding off will give rise to very large errors. It has, for instance, been argued by Hotelling (ref. 5) that in solving a set of $n$ equations we should keep $n \log_{10} 4$ extra or "guarding" figures."* A key point of this discussion is to realize that during a period of five years GE was almost discarded as a reliable method for solving linear systems of equations in modern computers and, as a consequence, that several other methods were actively investigated. In addition, note that the cause of this situation was the absence of an *adequate rounding error analysis of GE* guaranteeing good error bounds for the computed solution, but that GE was accepted in the 1940s to be very efficient with respect the number of needed arithmetic operations. *The technical discipline*, often considered boring and too specialized in the 21st century, *of rounding error analysis came to scene as a first actor, and it was essentially created in the 1940s for solving the GE problem.*

The error analyses developed by von Neumann and Goldstine [21] and by Turing [20] are much more sophisticated than the one by Hotelling and they restored the confidence on GE. However, it should be remarked that none of these papers solved satisfactorily the problem of the error analysis of GE: this problem was too formidable even for genius as von Neumann and Turing, two of the greatest mathematicians in history, who are famous for solving some of the hardest problems in the History of Mathematics. This difficulty in the analysis

is in stark contrast with the fact that GE is a very simple algorithm taught at high school level. The error analysis of GE accepted nowadays came much later, in 1961, with the pioneer work by James Wilkinson in [24] and it will be discussed with detail in Subsection 4.3. However, *a complete rigorous solution of the problem of the rounding error analysis of GE remains as one of the major unsolved problems in Numerical Analysis*, and its precise formulation will be discussed in Subsection 4.4.

The results in [21] deserve a few words. The long, difficult, and rigorous error analysis by von Neumann and Goldstine is not general because is only valid for systems $Ax = b$ where the matrix $A$ is positive definite. This covers the important case of the normal equations arising in least squares problems[4], but not many other linear systems that are important in applications. After developing a theory of GE and LU factorization for general matrices $A$, von Neumann and Goldstine honestly recognize at the beginning of Section 5.1 in [21] that they are unable of performing a general rounding error analysis and they limit their analysis to positive definite matrices. We quote from [21, p. 1056]: *"We have not so far been able to obtain satisfactory error estimates for the pseudo-operational equivalent of the elimination method in its general form, ... We did, however, succeed in securing everything that is needed in the special case of a definite A."*.

In contrast, Turing considered in [20] the error analysis of GE in the general case. Facing a problem that von Neumann could not solve is a very strong indicator of Turing's great courage, self-confidence, and unbounded ambition as researcher. However, the analysis done by Turing has some important drawbacks, although his conclusions on the errors committed by GE with partial pivoting and its practical use on modern computers still remain valid today. These questions will be further discussed in Section 5.

Next, two additional points on the three papers are discussed. The three papers [15, 20, 21] were written by top researchers, who considered the problem very important from an applied point of view, but also, *from a fundamental point of view*, since GE was the first algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established for the first time. This is especially evident in the paper by von Neumann and Goldstine that spends 19 pages establishing the sources of errors in computations and the rules to perform rounding error analyses of algorithms running on modern computers.

## 3.1 A few words on the authors of the three papers

I will not explain in detail the mathematical contributions of John von Neumann (see Figure 3) and Alan Turing (see Figure 1), since both are very well-known and are considered as two of the most important mathematicians in history. Jean Dieudonné wrote that John von Neumann has been *"the last of great mathematicians"* [4], as a consequence of the large number of different fields where von Neumann did major contributions. These fields include, among others, set theory, functional analysis, numerical analysis, quantum mechanics, game theory, and computer science. In fact, von Neumann was a founder of some of these fields as, for instance, game theory and computer science. Alan Turing made also fundamental contributions in several areas. He solved the famous "decision problem" posed by David Hilbert in 1928 via the invention of Turing's machines. In addition, Turing was one of the founders of modern cryptanalysis, of computers science, of artificial intelligence, of modern numerical analysis, and of mathematical biology. The definitive source of information about Turing's life and contributions is the monumental biography [14] by Andrew Hodges.

Harold Hotelling and Herman Goldstine (see Figure 3), the other authors of the *three papers*, were also top researchers in their times, although not of the same level as von Neumann and Turing. Hotelling was born in Minnesota in 1895. He was a mathematical statistician and an influential economic theorist. He held positions in prestigious institutions as Stanford University (1927-31), Columbia University (1931-46), and finally he became Professor of Mathematical Statistics at the University of North Carolina at Chapel-Hill (1946-1973). He received the North Carolina Award for contributions to science in 1972 and a street in Chapel Hill bears his name. He is widely known to statisticians because he introduced Hotelling T-square distribution and, more importantly, the canonical correlation or principal component analysis, which is a fundamental technique in statistics.

Herman Goldstine was born in Chicago in 1913. He was awarded bachelor (1933), master (1934) and PhD (1936) degrees in mathematics from the University of Chicago. In 1941 he wrote the technical description for ENIAC (Electronic Numerical Integrator And Computer), which was the first electronic computer starting to work in 1946 (up to 1955). He joined the Army in 1942, when the United States entered into World War II and

[4]It should be noted that today, it is widely known that the use of normal equations for solving least squares problems may be unstable and they are never used in professional software. The standard algorithm for least squares problems is based on another famous matrix factorization: the QR factorization [3, 7, 12]. However, this was unknown when the paper by von Neumann and Goldstine was published.

John von Neumann (1903-1957)

Herman Goldstine (1913-2004)

Harold Hotelling (1895-1973)

Figure 3: John von Neumann and Herman Goldstine were the authors of *"Numerical inverting of matrices of high order"* in 1947 [21], often considered as the *first paper of modern Numerical Analysis*. Harold Hotelling was an influential statistician who introduced principal component analysis, among other contributions. In 1943, he did an error analysis of GE which led to general pessimism on its practical use in modern computers.

he persuaded the Army to fund the construction of ENIAC in 1943. He became program manager of ENIAC. Although ENIAC was thousands of times faster than previously available electro-mechanical machines and was programmable, there was no way to issue orders at electronic speed (modern programs)[5] and ENIAC had to be configured with patch cords and rotary switches for each task. Therefore, the need of a ENIAC's successor was evident even before ENIAC was completed and this motivated an indirect contribution, but extremely important for the history of computer science and GE, by Goldstine: In 1944 Goldstine involved von Neumann in planning ENIAC's successor and this resulted in the famous von Neumann's 1945 report *"First draft of a report on the EDVAC"* on how to build a modern computer (available in [1]), and in a long and fruitful collaboration between Goldstine and von Neumann. Goldstine was awarded the USA National Medal of Science in 1985.

An additional information may be of interest for readers on the fascinating 1940's: It is often said that von Neumann's famous report *"First draft of a report on the EDVAC"*, together with Turing's also famous 1946 report *"Proposed Electronic Calculator"* (available in [2]) are the foundational documents of computer architecture and that most of the ideas stated there still remain valid today.

### 3.2 Turing's and von Neumann's projects for building modern computers

Many projects for constructing modern computers got underway in the 1940s. A brief account of them may be found in [9] and a complete history in [17]. Here, I say just a few words on the projects in which Alan Turing and John von Neumann were involved, because then, they simultaneously became interested in the error analysis of GE. This stresses further the fact that the research on rounding error analysis of GE is motivated by applications and runs parallel with the development of modern computers.

Turing was involved in the NPL Pilot ACE (National Physical Laboratory Pilot Automatic Computing Engine) project developed in Teddington, England. Turing worked at NPL from 1945 to 1948 and during this period he also worked in rounding error analysis of GE. Basically, Turing did the first design of Pilot ACE in 1946 which was, probably, very ambitious for the resources of NPL and was never constructed. The Pilot ACE started to work in May 1950, without Turing, based mainly on ideas of Harry Huskey and James Wilkinson (see more comments in [28]).

Turing moved to The University of Manchester in September 1948. There, he collaborated in the Baby/Mark 1 project. The Small-Scale Experimental Machine, known as the "Baby", made its first successful run of a program on June 21st 1948. It was the first machine that had all the components now regarded as characteristic of a modern computer. Most importantly it was the first computer that could store not only data but any user program in electronic memory and process it at electronic speed. From the "Baby" a full-sized machine

[5] Therefore ENIAC is not considered a "modern computer".

was designed and built, the Manchester Mark 1, which by April 1949 was generally available for computation in scientific research in the University of Manchester. Turing worked in the project and helped to design the program language of the computer.

Von Neumann started to lead the computer project at the Institute of Advanced Studies at Princeton (USA) (the "IAS computer project") in 1946 and Goldstine joined him from the very beginning. In this period they became interested in rounding errors in GE. The first IAS computer started to work in 1951, i.e., later than its UK competitors. However, its influence on modern computers is, probably, more important, since several clones of the IAS computer were built from 1952 to 1957, including the first IBM mainframe.

# 4 Rounding error bounds for Gaussian elimination

After explaining the history of GE and its particular historical context in the 1940's, now the key properties of the rounding errors committed by GE are considered. This is the most technical section of the paper and "for encouraging" the reader, I will start with a quotation from the Preface of one of the most popular textbooks on Numerical Linear Algebra, written by Lloyd N. Trefethen and David Bau [19]:

> *"...We have departed from the customary by not starting with* Gaussian elimination. *That algorithm is atypical of Numerical Linear Algebra,* exceptionally difficult to analyze*, yet at the same time tediously familiar to every student..."*

In plain words, this means that GE is recognized by professional numerical analysts as very easy to explain, but *very difficult to analyze*. Therefore, I am asking the reader an extra effort for understanding its analysis!

## 4.1 The axioms of rounding error analysis

Rounding errors in computers come from two facts. First, computers can only represent a finite subset of the real numbers, which is called the set of *floating point numbers*, and is usually denoted by $\mathbb{F}$. This fact alone, obviously, produces errors when storing the data of any problem on the computer. Second, $\mathbb{F}$ is not closed under the basic arithmetic operations $(+, -, \times, /)$, however when these operations are performed on a computer, they must give another number of $\mathbb{F}$, and this fact produces further errors. These two facts are encapsulated into the *axioms of rounding error analysis*, that can be found in many textbooks [3, 7, 12, 19]. These axioms are:

**Axiom 1** (Rounding) *If $x \in \mathbb{R}$ lies in the range of $\mathbb{F}$, then $x$ is approximated by a number $fl(x) \in \mathbb{F}$ such that*

$$fl(x) = x\,(1+\delta), \qquad |\delta| \leq \mathbf{u},$$

*where $\mathbf{u}$ is the* unit roundoff of the computer.

In current computers $\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$ in double precision and $\mathbf{u} = 2^{-24} \approx 5.96 \times 10^{-8}$ in single precision. In Axiom 1, the exact meaning of the sentence "$x \in \mathbb{R}$ lies in the range of $\mathbb{F}$" is that the absolute value of $x$ is smaller than or equal to the largest absolute value of the numbers in $\mathbb{F}$ and larger than or equal to the smallest absolute value of the nonzero numbers in $\mathbb{F}$.

**Axiom 2** (Floating Point Arithmetic) *If $x, y \in \mathbb{F}$ and $\mathbf{op} \in \{+, -, \times, /\}$, then*

$$\text{computed}(x\,\mathbf{op}\,y) = (x\,\mathbf{op}\,y)\,(1+\alpha), \qquad |\alpha| \leq \mathbf{u},$$

*where $(x\,\mathbf{op}\,y)$ is the exact result of the operation, that may not be in $\mathbb{F}$, and* computed$(x\,\mathbf{op}\,y)$ *is the result produced by the computer.*

Axiom 2 is sometimes called the *"exact-round principle"* and, in plain words, it can be stated as *"computers should be thought of as performing each arithmetic operation exactly and then rounding to a floating point number"*. The reader should note the key role that the unit roundoff $\mathbf{u}$ plays in Axioms 1 and 2. All algorithms are combinations of (many) basic $\{+, -, \times, /\}$ operations and the idea of rounding error analysis is to combine via the axioms above the errors in all these operations to produce a final relative error in the computed magnitude. In most cases, only the first order term in $\mathbf{u}$ of the relative error is necessary and this makes the analyses much simpler. So, *in this paper, we will restrict ourselves to state first order rounding error bounds.*

From a historical point of view, it should be noted that Axioms 1 and 2 of floating point arithmetic were introduced by Wilkinson in 1960 [23], but that the original idea of establishing simple axioms for rounding error analysis goes back to von Neumann and Goldstine in their 1947 paper [21], where they introduced corresponding axioms for the fixed point arithmetic used in the 1940s. The error analysis in Turing's 1948-paper [20] does not include axioms for rounding errors and, in this sense, is very far from current error analyses.

## 4.2 A simple explanation of Hotelling's exponentially-increasing error bound

The main reason why Hotelling obtained a rounding error bound for GE that increases exponentially with the size of the matrix is easy to understand by combining Axioms 1 and 2 (that Hotelling did not know!) with Algorithm 1, and *performing a naive direct rounding error analysis*. If Algorithm 1 is written in formal mathematical language, i.e., avoiding equalities like $a_{ij} = a_{ij} - a_{ik}a_{kj}$ that in formal Mathematics have the only meaning of $a_{ik}a_{kj} = 0$, then the following updating is obtained

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}, \qquad k+1 \leq i,j \leq n\,. \tag{10}$$

Here $a_{ij}^{(k)}$ are entries of the matrix $A^{(k)}$, $k = 1, 2, \ldots, n$, defined in Subsection 2.4 and note that only the entries $k+1 \leq i,j \leq n$ are updated at the $k$th stage, so we define $a_{ij}^{(k+1)} = a_{ij}^{(k)}$ for the remaining entries. In addition, due to the fact that Algorithm 1 stores the nontrivial entries of the $L$ factor in the strictly lower triangular part of $A$, the matrix $A^{(k)}$ has the following structure: it has zeros below the diagonal in the first $k-1$ columns and the rest of the entries are the $a_{ij}^{(k)}$ entries defined above.

Now, let us proceed with a simplified analysis and denote by $\widehat{A}^{(k)}$ the computed matrix in floating point arithmetic by Algorithm 1 corresponding to the exact matrix $A^{(k)}$. $\widehat{A}^{(1)}$ does not involve any arithmetic operation, since it comes from storing $A^{(1)} := A$ in the computer and, therefore, Axiom 1 implies that $\widehat{a}_{ij}^{(1)} = a_{ij}^{(1)}(1+\delta_{ij})$ with $|\delta_{ij}| \leq \mathbf{u}$. This is equivalent to the following bound for the relative error[6] in each entry: $\left|\widehat{a}_{ij}^{(1)} - a_{ij}^{(1)}\right| / \left|a_{ij}^{(1)}\right| = |\delta_{ij}| \leq \mathbf{u}$. Assume now that the entries of $\widehat{A}^{(k)}$ satisfy the relative error bound

$$\left|\frac{\widehat{a}_{ij}^{(k)} - a_{ij}^{(k)}}{a_{ij}^{(k)}}\right| \leq \mathbf{e_k}, \quad \text{for all } k \leq i,j \leq n, \tag{11}$$

i.e., $\mathbf{e_k}$ is an upper bound on *the maximum relative error at the kth stage of GE*. This is what we want to determine by induction and by taking into account that we know $\mathbf{e_1} = \mathbf{u}$. Next, let us pay attention to the last term in (10), i.e., $a_{ik}^{(k)} a_{kj}^{(k)} / a_{kk}^{(k)}$, which is in fact the responsible of the exponential growth of the error bound. As most readers learnt when they were very young (probably, even before they learnt GE or, for sure, no later than the first year in the University), the relative error of *an exact* series of products and quotients of numbers affected by relative errors is the sum of the relative errors of each individual number. So, from (11),

$$\left|\frac{\frac{\widehat{a}_{ik}^{(k)}\widehat{a}_{kj}^{(k)}}{\widehat{a}_{kk}^{(k)}} - \frac{a_{ik}^{(k)}a_{kj}^{(k)}}{a_{kk}^{(k)}}}{\frac{a_{ik}^{(k)}a_{kj}^{(k)}}{a_{kk}^{(k)}}}\right| \leq 3\,\mathbf{e_k}, \tag{12}$$

where 2nd-order terms in the errors have been discarded. Of course, there are still more errors, coming from Axiom 2, when computing (10) in floating point arithmetic, but their effect is *to increase the error bound in* (12), they are not essential in our simplified analysis, and they are omitted. Therefore, a bound on the *maximum relative error at* $(k+1)$*th stage of GE*, i.e., $\mathbf{e_{k+1}}$, satisfies $\mathbf{e_{k+1}} \gtrsim 3\mathbf{e_k}$, and, since $\mathbf{e_1} = \mathbf{u} \approx 10^{-16}$ and GE performs $(n-1)$ stage transitions for an $n \times n$ matrix, the following error bound

$$\mathbf{e_n} \gtrsim 3^{n-1}\mathbf{e_1} \approx 3^{n-1} \cdot 10^{-16} \tag{13}$$

is finally obtained. The error bound in (13) is really huge even for small sized matrices: for $n = 30$, $\mathbf{e_n} \gtrsim 6.9 \times 10^{-3}$; for $n = 40$, $\mathbf{e_n} \gtrsim 4.1 \times 10^{2}$; and, for $n = 50$, $\mathbf{e_n} \gtrsim 2.4 \times 10^{7}$. Therefore, for $n \geq 40$, the error bound

[6] In this informal analysis, it is assumed that all entries $a_{ij}^{(k)}$, for $1 \leq k \leq n$ and $k \leq i,j \leq n$, are different from zero. This assumption is generic and allows us to avoid technicalities that would obscure the main ideas.

(13) does not guarantee a single digit of accuracy in the results of GE! As it was explained in Section 3, this led to general pessimism in mid 1940s about the practical use of GE and motivated the papers by von Neumann and Goldstine [21] and by Turing [20]. However, the error analysis that appears in modern textbooks is the one presented by James Wilkinson in his fundamental paper [24]. This is discussed in next subsection.

### 4.3 James Wilkinson's backward error analysis of GE

*Backward error analysis represents a drastic change of mind.* The natural approach to rounding error analysis seems to be to bound the difference between the exact solution $x$ of the linear system $Ax = b$ and the approximate solution $\widehat{x}$ computed in floating point arithmetic by Algorithm 2, i.e., by modern GE. In contrast, *backward error analysis* bounds the difference between the matrix $A$ and a certain matrix $A + \Delta A$ such that $(A + \Delta A)\widehat{x} = b$. If this difference is small, then *backward error analysis establishes that the computed solution is the exact solution of a nearby linear system.* Although this might seem odd at a first glance, note that the exact matrix $A$ is never available for the computer, because errors are committed just by storing $A$ in the computer (see Axiom 1) and, in addition, very often in practice the entries of $A$ are affected by experimental or modelling errors. Therefore, even in the ideal case that GE does not make errors after storing $A$ and $b$ in the computer, the computed solution would be just the solution of a nearby linear system and *backward error analysis aims to describe the best possible situation* that one can imagine in practice.

Before the famous result by Wilkinson is stated, it is necessary to establish effective ways to measure differences between matrices ($A$ and $A + \Delta A$) and vectors ($x$ and $\widehat{x}$). This is done via matrix and vector norms [12, Chapter 6]. In this paper only the infinite-norm is used. If $x$ is an $n \times 1$ vector and $A$ is an $n \times n$ matrix, then their vector and matrix infinite-norms are defined as

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i| \quad \text{and} \quad \|A\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| \,.$$

Nowadays, every numerical analyst is familiar with matrix norms, but this was not the case in the 1940's. In fact, the paper by Von Neumann and Goldstine [21] is the first one to bring matrix norms to the attention of numerical analysts in a systematic way. Now we can state (to first order in $\mathbf{u}$) the result by Wilkinson.

**Theorem 1 (Wilkinson, 1961 [24]. Backward errors in GE.)** *Let $A$ be a real $n \times n$ nonsingular matrix, let $b$ be a real $n \times 1$ vector, and let $\widehat{x}$ be the approximate solution of the linear system $Ax = b$ computed by GE with partial pivoting in a computer with unit roundoff $\mathbf{u}$. Then $\widehat{x}$ satisfies*

$$(A + \Delta A)\widehat{x} = b, \qquad \textit{with} \quad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \le 3\, n^3\, \mathbf{u}\, \boldsymbol{\rho}_n \,, \tag{14}$$

*where*

$$\boldsymbol{\rho}_n = \frac{\max_{ijk} |a_{ij}^{(k)}|}{\max_{ij} |a_{ij}|} \tag{15}$$

*is the growth factor of GE with partial pivoting. Here $A^{(1)} = A, A^{(2)}, \ldots, A^{(n)} = U$ are the matrices appearing in the GE process as they were defined in Section 2.4.*

The proof of Theorem 1 is not difficult with the tools currently available, but it requires some technical work and is omitted. Interested readers can found two different modern proofs in [12] (shorter and sharper) and [7] (following step by step Algorithm 2). Observe that equation (14) indeed states that the computed solution $\widehat{x}$ is the exact solution of a linear system that is very close to the original one, i.e., to $Ax = b$, as long as the growth factor $\boldsymbol{\rho}_n$ is not large. This is in fact the case as we will discuss in Subsection 4.4 and, so, it is said that GE with partial pivoting is a *backward stable algorithm.* Theorem 1 is an instance of the "mantra" that every numerical analyst working on matrix computations should repeat again and again: *"The ideal objective of an algorithm is to compute outputs that are exact for nearby inputs, because this means that the algorithm achieves as much accuracy as the data warrants".* The most reputed algorithms of Numerical Linear Algebra are backward stable, but not all algorithms used in practice are.

Some modern texts [12, p. 185] and papers [9] indicate that Von Neumann & Goldstine and Turing introduced "backward error analysis" in [21, 20] . In my opinion, *this is not completely true.* Von Neumann & Goldstine and Turing indeed mentioned backward errors (without the name) in these papers, but in a rather marginal way, and

did not realize the importance of this concept. For instance, Turing mentions backward errors in the last page of his 22 pages long paper and Von Neumann & Goldstine in page 71 of their 79 pages long paper. Wilkinson attributed in [27] the credit for the first backward error analysis to Wallace Givens in 1954 for an analysis of an algorithm for computing the eigenvalues of symmetric tridiagonal matrices by using the Sturm sequence property of their leading principal minors. *Nowadays, backward error analysis is one of the most fundamental and powerful ideas in Numerical Analysis and this is mostly a consequence of the monumental research work done by James Wilkinson on rounding errors* (see Figure 1).

James Wilkinson was a Cambridge-trained English mathematician who worked as Turing's assistant at NPL (1946-48). He is considered as the founder of modern rounding error analysis by using systematically backward errors for analyzing many numerical algorithms for matrix computations. He wrote two influential books on Numerical Analysis: *Rounding Errors in Algebraic Processes* in 1963 [25] and *The Algebraic Eigenvalue Problem* in 1965 [26]. James Wilkinson said in [28, pp. 143-144] that his first contact with backward errors happened while he was serving in the United Kingdom *Armament Research Department* during World War II. At that time, he had to solve a system $Ax = b$ of twelve linear equations and decided to use GE with partial pivoting. Wilkinson was sure that the computed solution $\widehat{x}$ had errors several orders of magnitude larger than the unit roundoff (of those times!). However, when he substituted $\widehat{x}$ in the equations to his *"astonishment the left-hand side agreed with the given right-hand side to"* full accuracy. In modern language, this means that *the residual* $b - A\widehat{x}$ satisfied $\|b - A\widehat{x}\|_\infty \approx \mathbf{u}\,\|b\|_\infty$, a fact that is deeply connected to backward errors, as we will discuss in Section 4.6. Wilkinson claimed at that time *"I have the exact solution corresponding to a right-hand side which differs only in the tenth figure from the given one"*. Unfortunately, Wilkinson did not pursue then this line of research since it was not appreciated by his taskmaster at the Armament Research Department.

## 4.4 One of the major unsolved problems in Numerical Analysis

The backward error bound (14) for GE with partial pivoting (GEPP) includes the growth factor $\boldsymbol{\rho}_n$ of the matrix $A$. This factor is the ratio of the maximum absolute value of the entries of the matrices arising in GEPP and the maximum absolute value of the entries of the original matrix $A$. Example 1 illustrates the growth factor in a matrix that has been arranged, for simplicity, in such a way that GEPP does not require any permutation.

**Example 1**

$$A = \begin{bmatrix} -4 & 2 & 1 & -1 \\ 1 & 6 & 2 & -2 \\ 1 & -2 & 5 & 1 \\ 3 & -4 & 2 & -10 \end{bmatrix} \longrightarrow A^{(2)} = \begin{bmatrix} -4 & 2 & 1 & -1 \\ 0 & 6.5 & 2.25 & -2.25 \\ 0 & -1.5 & 5.25 & 0.75 \\ 0 & -2.5 & 2.75 & -10.75 \end{bmatrix} \longrightarrow$$

$$A^{(3)} = \begin{bmatrix} -4 & 2 & 1 & -1 \\ 0 & 6.5 & 2.25 & -2.25 \\ 0 & 0 & 5.77 & 0.23 \\ 0 & 0 & 3.62 & -11.62 \end{bmatrix} \longrightarrow A^{(4)} = \begin{bmatrix} -4 & 2 & 1 & -1 \\ 0 & 6.5 & 2.25 & -2.25 \\ 0 & 0 & 5.77 & 0.23 \\ 0 & 0 & 0 & -11.76 \end{bmatrix}.$$

The maximum absolute value of the entries of $A$ is 10 and the maximum absolute value of the entries of $A^{(2)}$, $A^{(3)}$, and $A^{(4)}$ is 11.76. Therefore

$$\boldsymbol{\rho} = \frac{11.76}{10} = 1.1760\,.$$

Note that the growth factor is larger than or equal to one for any matrix $A$ by definition.

The key question in this context is to determine whether there exist matrices with very large growth factors or not. The answer is given in Theorem 2 and is *yes*. Wilkinson knew this fact as early as in 1954 [22], long before developing his backward error analysis.

**Theorem 2 (Wilkinson, 1954)** *Let $A$ be an $n \times n$ nonsingular matrix. Then the growth factor of $A$ for GE with partial pivoting satisfies*

$$\rho_n(A) \leq 2^{(n-1)},$$

*and this bound is attained for the $n \times n$ matrix*

$$B_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 1 & 0 & \cdots & 0 & 1 \\ -1 & -1 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & -1 & \cdots & 1 & 1 \\ -1 & -1 & -1 & \cdots & -1 & 1 \end{bmatrix}.$$

Combining Theorem 2 with the bound in (14) for the unit roundoff $\mathbf{u} = 2^{-53}$ of double precision, one obtains

$$\frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3\, n^3\, \mathbf{u}\, \boldsymbol{\rho}_n \leq 3\, n^3\, 2^{(n-54)}\,, \tag{16}$$

which is a huge bound for matrices with $n \geq 54$, i.e., for very small matrices, and would make GEPP useless in practice. In some sense, (16) says us that Hotelling was right: there is no way of avoiding in GEPP errors that increase exponentially with the size of the matrix. However, note that Wilkinson's analysis gives us much more information than the one by Hotelling, since (14) implies that the backward errors are tiny whenever the growth factor $\boldsymbol{\rho}_n$ of $A$ is moderate. Therefore, *GEPP would be a reliable algorithm in practice if matrices with large growth factors are very rare*, otherwise it may produce frequently large errors. This is indeed the case: large growth factors are extremely rare, as it is stated in the next paragraph by Nick Higham [12, p. 168]:

> *"To summarize, although there are practically occurring matrices for which partial pivoting yields a moderately large, or even exponentially large, growth factor, the growth factor is almost invariably found to be small. Explaining this fact remains one of the major unsolved problems in Numerical Analysis."*

How should we pose precisely this unsolved problem? One option is to consider random matrices whose entries are independent random variables and to prove that the probability of encountering a growth factor $\boldsymbol{\rho}_n > \alpha$ decreases extremely fast as $\alpha$ increases (perhaps, exponentially fast!). More information on the solution of this problem, including a money prize, can be found in [18]. Here, I want to stress some important facts on this open problem. First, its solution would not change at all the algorithm of modern GE. Second, *GEPP has not waited to the solution of the open problem for being widely used, since GEPP is nowadays the standard method for solving linear systems of equations on computers, despite the fact that its stability is not rigorously proved.* This is based on years of practical experience with GEPP that have shown that large growth factors never occur in real computing. Third, the idea that practical numerical methods do not need to be fully supported by proofs for being useful goes back to Turing's paper [20], as it will be discussed in Section 5. Finally, there are methods that are perfectly backward stable for solving linear systems (like the one based on the QR factorization [19]), but they are not used in practice since they are computationally more expensive than GEPP.

## 4.5 From backward to forward errors: The condition number of a matrix

The fact that a numerical algorithm is *"backward stable"* is very satisfactory for numerical analysts, since it is equivalent to say that the errors are the best that can be expected from the input data. However, for users of software it may be a somewhat obscure concept and many times a bound on the *forward errors* is preferred. In the case of GEPP, the forward error is $\|\widehat{x} - x\|_\infty / \|x\|_\infty$, where $x$ and $\widehat{x}$ are, respectively, the exact and the computed solution of $Ax = b$. From Theorem 1, the problem of bounding the forward error in the solution can be posed as a pure mathematical problem of *perturbation theory*, i.e., if the input matrix $A$ is perturbed, how much does the solution change? We use the notation of Theorem 1 and present the solution of this problem as it was stated by Wilkinson in [25, p. 93]

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\dfrac{\|\Delta A\|_\infty}{\|A\|_\infty}}{1 - \|A^{-1}\|_\infty \, \|\Delta A\|_\infty}\,, \tag{17}$$

where it is assumed that $\|A^{-1}\|_\infty \, \|\Delta A\|_\infty < 1$. By discarding second order terms in the perturbation $\|\Delta A\|_\infty$ and by using (14), equation (17) becomes

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, (3\, n^3\, \mathbf{u}\, \boldsymbol{\rho}_n)\,. \tag{18}$$

The inequalities in (18) tell us that *tiny relative perturbations of the matrix $A$ may produce large relative variations in the solution if the number $\|A\|_\infty \|A^{-1}\|_\infty$ is huge and that, even in the case that the growth factor of $A$ is moderate, the "forward errors" committed by GEPP may be large if $\|A\|_\infty \|A^{-1}\|_\infty$ is huge.* We see that the number $\|A\|_\infty \|A^{-1}\|_\infty$ plays a fundamental role in the perturbation theory of the solution of linear systems and in the "forward errors" committed by GEPP. It is the very famous *condition number of a matrix*:

$$\kappa_\infty(A) := \|A\|_\infty \|A^{-1}\|_\infty . \tag{19}$$

I want to insist more on a fact that is very familiar to numerical analysts, but that it is still surprising for many users of numerical software. *There are not numerical algorithms that solve linear systems of equations with guaranteed tiny forward errors*, i.e., with forward errors that are always of order unit roundoff. Bounds $O(\mathbf{u})\,\kappa_\infty(A)$ as the one in (18) are the best that hold for linear solvers valid for general matrices. There is no way to avoid in general the presence of the condition number.

The condition number $\kappa_\infty(A)$ (or in other norms) arises in many other problems in matrix computations and from the point of view of perturbation theory and numerical applications is the most important single number attached to a matrix [3, 7, 12]. It is not easy to determine who discovered the *"condition number"*. No question that the name was introduced by Turing in [20] and, in my opinion, Turing also deserves the credit for the concept. The essentials are in [20], although it is true that Turing gives an "unusual" definition of "condition number" and also that shows in an unusual way its relationship with the variation of the solution of a linear system $Ax = b$ under perturbations of $A$ and $b$. Before Turing's paper, von Neumann & Goldstine used the condition number (in the 2-norm and with the name "figure of merit") in their error bounds, but they do not show any clear perturbation inequality involving the condition number. The first fully rigorous perturbation results on condition numbers were proved by Bauer in 1959 for matrix inverses and by Wilkinson in 1963 for linear systems (see [9] for more details).

## 4.6 Backward errors and residuals

Theorem 1 presents backward errors of GEPP, but it does not show how Wilkinson reached this "at a first-glance unnatural" way of presenting/analyzing rounding errors. I have already commented in the last paragraph of Section 4.3 that Wilkinson was motivated by a few numerical tests that always produced tiny residuals. In fact, we will see in Section 5 that this was also Turing's motivation for undertaking his error analysis of GE. Therefore, I discuss in this section the deep connection existing between backward errors and residuals and how residuals can be used to give sharp optimal estimates of backward errors. The results can be applied to any algorithm for solving linear systems and not only to GEPP.

First, observe that if the approximated solution, $\widehat{x}$, of $Ax = b$ computed by a certain algorithm satisfies $(A + \Delta A)\widehat{x} = b$, with $\|\Delta A\|_\infty = O(\mathbf{u})\,\|A\|_\infty$, then $b - A\widehat{x} = \Delta A\,\widehat{x}$. So, $\|b - A\widehat{x}\|_\infty \leq \|\Delta A\|_\infty \|\widehat{x}\|_\infty = O(\mathbf{u})\,\|A\|_\infty \|\widehat{x}\|_\infty$. In plain words, this means that a tiny relative backward error of order $\mathbf{u}$ implies a tiny *relative residual* $\|b - A\widehat{x}\|_\infty / (\|A\|_\infty \|\widehat{x}\|_\infty)$ also of order $\mathbf{u}$. Much more surprising is that the implication in the opposite direction is also true, i.e., that a tiny relative residual implies a tiny relative backward error. In fact, for any matrix $A$ and for any vectors $\widehat{x}$ and $b$, it can be proved that

$$\frac{\|b - A\widehat{x}\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = \min \left\{ \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \ : \ (A + \Delta A)\widehat{x} = b \right\} . \tag{20}$$

According to the discussion in [12, pages 12 and 29], the result in (20) was proved by Wilkinson for the 2-norm in some moment in the 1950s and, after discovering it, he began to develop backward error analysis systematically. Rigal and Gaches proved in 1967 [16] a result much more general than (20), where they allow perturbations in $A$ and $b$ and the use of any vector norm and the corresponding subordinate matrix norm. An excellent modern reference on different relationships between residuals and backward errors for linear systems is [12, Chapter 7].

Now, the reader can fully appreciate why the fact that GEPP computed solutions $\widehat{x}$ with relative residuals $\|b - A\widehat{x}\|_\infty / (\|A\|_\infty \|\widehat{x}\|_\infty) = O(\mathbf{u})$ in all the numerical tests that Wilkinson performed was a strong motivation for trying to prove a result in the spirit of Theorem 1, but the proof had to wait for some years and came from the hand of the nontrivial growth factor. Also note that the left-hand side of (20) provides a simple practical way for computing the "best possible backward error" of the approximate solution $\widehat{x}$ with respect the linear system $Ax = b$. We finish this section with an example that illustrates the presented concepts.

**Example 2** Consider the matrix

$$A = \begin{bmatrix} 8.679678 \cdot 10^{13} & -1.992403 \cdot 10^{-1} & 2.542877 \cdot 10^{-1} \\ 2.274573 \cdot 10^{14} & -5.221239 \cdot 10^{-1} & 6.663797 \cdot 10^{-1} \\ -1.134348 \cdot 10^{14} & 2.603875 \cdot 10^{-1} & -3.32329 \cdot 10^{-1} \end{bmatrix}$$

and store it in the program MATLAB [12, p. 575], which uses double precision floating point arithmetic. MATLAB gives the following value for the condition number of $A$: $\kappa_\infty(A) \approx 1.06 \times 10^{22}$. Define the $3 \times 1$ vector $x = [1, 1, 1]^T$ and compute in MATLAB $b = Ax$. So, we have constructed a linear system whose exact solution is known. Compute the solution $\widehat{x}$ by using the *backslash* command (\) of MATLAB, which uses GEPP. Then we get, also in MATLAB,

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 1 \quad \text{and} \quad \frac{\|b - A\widehat{x}\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} \approx 1.4 \cdot 10^{-16}\,.$$

The growth factor (15) of $A$ for GEPP is 1. Observe that the relative error in the solution is huge, which is explained by (18) with $\mathbf{u} \approx 10^{-16}$. However, the relative residual is of order $\mathbf{u}$, according to (14) and (20). As Wilkinson used to say, huge errors in the solution must be "diabolically correlated" to give tiny residuals.

# 5 Remarks on Alan Turing's paper on rounding errors

Alan Turing wrote his famous "rounding-off error" paper [20] when he and James Wilkinson were in the National Physical Laboratory. The story of the genesis of the paper is told by Wilkinson in [28, pp. 144-45], where one can read the following

> *"... it happened that some time after my arrival, a system of 18 equations arrived in Mathematics Divison and ... we finally decided to abandon theorizing and to solve it ... The operation was manned by Fox, Goodwin, Turing, and me, and we decided on Gaussian elimination with complete pivoting.* **Turing was not particularly enthusiastic** *... partly* **because he was convinced that it would be a failure**. *History repeated ... and the residuals were again of order* $10^{-10}$*, that is of the size corresponding to the exact solution rounded to ten decimals. ... I suppose this must be regarded as a defeat for* **Turing since he**, *at that time,* **was a keener adherent than any of the rest of us to the pessimistic school**. *However, I'm sure that this experience made quite an impression on him and set him thinking afresh on the problem of rounding errors in elimination processes. About a year later he produced his famous paper "Rounding-off errors in matrix processes" ..."*

In the modern language introduced in Section 4.6 what Turing, Wilkinson and coworkers observed was that the relative residual was of order unit roundoff. Turing recognized in [20, pp. 287] that he was prompted to carry out his *"research largely by the practical work of L. Fox in applying the elimination method"*. Curiously, Turing did not mention here to Wilkinson, although he cited among the references in [20] a paper by Fox, Huskey, and Wilkinson on this subject published in the same journal and volume as [20] but 140 pages before.

Turing believed, *based on a few numerical tests available in the 1940s*, that GE with pivoting was an stable method for solving *general* systems of linear equations on a computer, and he undertook for first time the task of developing the corresponding rounding error analysis. Recall, in this context, that Von Neumann and Goldstine [21] only analyzed the stability of positive definite linear systems. However, *Turing knew that GE could fail, although only in exceptional cases.* This is made explicit in the first page of [20], where one finds

> *"Actually, although examples can be constructed where as many as* $n \log_{10} 2$ *extra figures would be required, these are exceptional. In the present paper the magnitude of the error is described in terms of quantities not considered in Hotelling's analysis; from the inequalities proved here it can immediately be seen that in all normal cases the Hotelling estimate is far too pessimistic."*

Therefore, *Turing essentially reached in the 1940s the same conclusion that remains valid today* and that has been discussed in Section 4.4: although the error bounds of GEPP may increase exponentially with the size for some matrices, these matrices are very rare and GEPP can be used with confidence in practice. *This Turing's pioneer insight has influenced in depth Numerical Analysis in general, and Matrix Computations in particular.* Observe also in this point, the differences between Turing's way of thinking and those of Hotelling and Von Neumann & Goldstine. Hotelling discovered that GE may produce errors that increase exponentially with the size of the matrix, something that is fully true, and this led him to pessimism on the use of GE. He was not able

of thinking if these exponentially increasing error bounds happen very rarely or not. On the other hand Von Neumann & Goldstine did not consider even the possibility of performing an error analysis of GE for general matrices, since they were not able of avoiding the exponential error bound.

However, it should be also remarked that Turing's analysis is non-standard from a modern point of view. In particular, it is based on the next key assumption [20, pp. 302 and 306], that I quote literally

> *"We assume that in the calculation of each quantity*
>
> $$\boldsymbol{A}_{ij}^{(r-1)} - \frac{\boldsymbol{A}_{rj}^{(r-1)} \boldsymbol{A}_{ir}^{(r-1)}}{\boldsymbol{A}_{rr}^{(r-1)}},$$
>
> *an error of at most $\epsilon$ is made. How this is to be secured need not be specified, but it is clear that the number of figures to be retained in $\boldsymbol{A}_{ir}^{(r-1)}/\boldsymbol{A}_{rr}^{(r-1)}$ will have to depend on the values of the $\boldsymbol{A}_{rj}^{(r-1)}$."*

The difficulty with this assumption is that no computer, neither present or past, can guarantee a rounding error bound like this in finite precision arithmetic. In fact, *"the error at most $\epsilon$"* eliminates from Turing's analysis any possibility of discovering the *growth factor*, which does not appear at all in [20]. Turing's rounding error bound for the solution of $Ax = b$ are expressed in terms of the unknown quantity $\epsilon$. Even with the unrealistic and ideal assumption $\epsilon = \mathbf{u}\,\|A\|_\infty$, Turing's error bound for the approximate solution $\widehat{x}$ computed by GEPP becomes a non-optimal bound of the type $\|x-\widehat{x}\|_\infty/\|x\|_\infty \lesssim \left(\|A\|_\infty \|A^{-1}\|_\infty\right)^2 p(n)\,\mathbf{u}$, with $p(n)$ a low degree polynomial in $n$ that does not depend on the growth factor. A trivial change in the last steps of Turing's analysis would produce $\|x-\widehat{x}\|_\infty/\|x\|_\infty \lesssim \left(\|A\|_\infty \|A^{-1}\|_\infty\right)\,(p(n)\,\mathbf{u})$, which has the standard form (18) but does not include the growth factor.

# 6 Research on error analysis of Gaussian elimination is still active

Since Wilkinson's pioneer paper [24] was published in 1961 many papers have been written on rounding error analysis of GE. This is a consequence of the fact that Wilkinson's Theorem 1 is essentially the best that can be proved for general nonsingular matrices $A$ via a normwise analysis, i.e., bounding just the norm of $\Delta A$. However, if the matrix $A$ belongs to some particular classes, then the properties of those classes can be exploited to obtain better bounds. It is also possible to perform a *componentwise* backward error analysis that often produces sharper results. The discussion of these topics is beyond the scope of this introductory paper, and I refer the reader to [12] and the references therein for a complete information on these topics.

As examples of very recent activities on the error analysis of GE, I discuss here very briefly the researches presented in the papers [11] and [6] published in the last two years. Reference [11] considers the LU factorization in the context of one of the hottest topics of numerical computations of the last years: "communication avoiding algorithms". In current and future computers the cost of communication (moving data between different levels of memory or between different processors) greatly exceeds the cost of performing arithmetic operations, therefore there is a strong motivation for developing new algorithms that communicate as little as possible, even if they do more arithmetic. In GE, this prevents the use of partial pivoting and a new strategy known as "tournament pivoting" has been proposed, which has required a new error analysis to prove its backward stability. Reference [6] develops and analyzes a framework that uses special implementations of GE with complete pivoting that allow us to compute solutions of linear systems with relative errors $O(\mathbf{u})$, i.e., removing the condition number in the bound (18), for the largest class of structured matrices known so far.

# 7 Conclusions

I have revised at an introductory level the first research works on the rounding error analysis of one of the most important numerical algorithms in Mathematics: Gaussian elimination for solving systems of linear equations. The pioneer work on this topic published by Alan Turing in 1948 has received particular attention, as well as the key results proved by James Wilkinson in 1961. In addition, other works published in the 1940's on the error analysis of GE have been discussed and the historical context of all these works has been considered in connection with the construction of modern computers in the 1940s. It has been pointed out that a complete and rigorous solution for the stability problem of GE still remains as an open problem.

**Acknowledgements**. The author thanks Profs. Manuel de León, David Ríos Insua, and Jesús María Sanz Serna, from the *Real Academia de Ciencias Exactas, Físicas y Naturales* of Spain, for the invitation to present a talk in the International Symposium *"The Alan Turing Legacy"*. That talk was the seed of the present paper. The author also thanks Profs. Fernando De Terán and Juan Manuel Molera for reading preliminary versions of this manuscript and for providing valuable suggestions, and Prof. Cristina Brändle for sharing with him her LaTeX and Beamer expertise. Finally, the author thanks his son Froilán for "running by hand" with him several examples of GE, for providing a fresh and young point of view about GE, and for reading a preliminary draft of this manuscript and correcting several errata.

# References

[1] W. Aspray and A. Burks, eds. *Papers of John von Neumann on Computing and Computer Theory.* MIT Press, Cambridge, MA, 1987.

[2] B. E. Carpenter and R. W. Doran, eds. *A. M. Turing's ACE report of 1946 and other papers.* MIT Press, Cambridge, MA, 1986.

[3] J. W. Demmel. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[4] J. Dieudonné. Von Neumann, Johan (or John). *Dictionary of Scientific Biographies,* Vol. 14, *C. C. Gillispie, ed., Charles Scribner's Sons,* New York, 1981, pp. 89-92.

[5] J. Dongarra and F. Sullivan. The top 10 algorithms of the century. *Comput. Sci. Eng.*, 2:22–23, 2000.

[6] F. M. Dopico and J. M. Molera. Accurate solution of structured linear systems via rank-revealing decompositions. *IMA J. Numer. Anal.*, 32(3):1096–1116, 2012.

[7] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[8] J. F. Grcar. How ordinary elimination became Gaussian elimination. *Historia Math.*, 38(2):163–218, 2011.

[9] J. F. Grcar. John von Neumann's analysis of Gaussian elimination and the origins of modern Numerical Analysis. *SIAM Rev.*, 53(4):607–682, 2011.

[10] J. F. Grcar. Mathematicians of Gaussian elimination. *Notices Amer. Math. Soc.*, 58(6):782–792, 2011.

[11] L. Grigori, J. W. Demmel, and H. Xiang. CALU: a communication optimal LU factorization algorithm. *SIAM J. Matrix Anal. Appl.*, 32(4):1317–1350, 2011.

[12] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2002.

[13] N. J. Higham. Gaussian elimination. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3:230–238, 2011.

[14] A. Hodges. *Alan Turing: the Enigma.* Princeton University Press, Princeton, NJ, centenary edition, 2012.

[15] H. Hotelling. Some new methods in matrix calculation. *Ann. Math. Statistics*, 14:1–34, 1943.

[16] J.-L. Rigal and J. Gaches. On the compatibility of a given solution with the data of a linear system. *J. Assoc. Comput. Mach.*, 14:543–548, 1967.

[17] R. Rojas and U. Hashagen, eds. *The First Computers. History and Architectures.* MIT Press, Cambridge, MA, 2000.

[18] L. N. Trefethen. The Smart Money's on Numerical Analysts. *SIAM News*, 45(9):1–5, 2012.

[19] L. N. Trefethen and David Bau, III. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[20] A. M. Turing. Rounding-off errors in matrix processes. *Quart. J. Mech. Appl. Math.*, 1:287–308, 1948.

[21] J. von Neumann and H. H. Goldstine. Numerical inverting of matrices of high order. *Bull. Amer. Math. Soc.*, 53:1021–1099, 1947.

[22] J. H. Wilkinson. Linear Algebra on the Pilot ACE. In *Automatic Digital Computation*, Her Majesty's Stationery Office, London, 1954.

[23] J. H. Wilkinson. Error analysis of floating-point computation. *Numer. Math.*, 2:319–340, 1960.

[24] J. H. Wilkinson. Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, 8:281–330, 1961.

[25] J. H. Wilkinson. *Rounding Errors in Algebraic Processes.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1963.

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

[27] J. H. Wilkinson. Modern error analysis. *SIAM Rev.*, 13:548–568, 1971.

[28] J. H. Wilkinson. Some comments from a numerical analyst. *J. Assoc. Comput. Mach.*, 18:137–147, 1971.