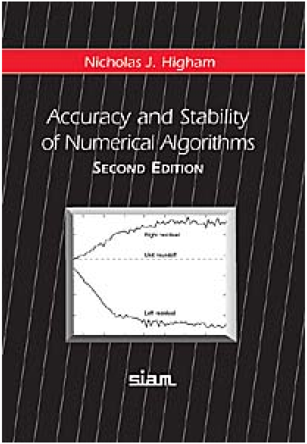
Alan Turing and the origins of modern Gaussian elimination

Froilán M. Dopico

Departamento de Matemáticas and
Instituto de Ciencias Matemáticas CSIC-UAM-UC3M-UCM,
Universidad Carlos III de Madrid, Spain

International Symposium: The Alan Turing Legacy.
Madrid, Spain, October 23-24, 2012

My scientific connection with Alan Turing (I)



I often teach a graduate course on "Numerical Linear Algebra", that is my research area, and N. Higham, "Accuracy and Stability of Numerical Algorithms", (SIAM, 2002) is one of my favorite references for this course.

My scientific connection with Alan Turing (II)

Nicholas Higham (1961-) is a prominent numerical analyst, who is Richardson Professor of Applied Mathematics in the School of Mathematics at Alan Turing Building in The University of Manchester.

Turing spent the last part of his life (1948-1954) at The University of Manchester.

My scientific connection with Alan Turing (III)

Nick Higham's book is dedicated to Alan Turing and James Wilkinson

Alan Turing (1912-1954)

James Wilkinson (1919-1986)

My scientific connection with Alan Turing (IV)

- The standard method for solving systems of linear equations in a computer is **Gaussian elimination**,
- **which is one of the most important numerical algorithms!!!**
- It is studied in depth in Chapter 9 of Nick Higham's, "Accuracy and Stability of Numerical Algorithms", where we found (pages 184-185)

"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper **"Rounding off errors in matrix processes"** *(Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308)..."*

My scientific connection with Alan Turing (IV)

- The standard method for solving systems of linear equations in a computer is **Gaussian elimination**,
- **which is one of the most important numerical algorithms!!!**
- It is studied in depth in Chapter 9 of Nick Higham's, "Accuracy and Stability of Numerical Algorithms", where we found (pages 184-185)

"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper **"Rounding off errors in matrix processes"** *(Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308)..."*

My scientific connection with Alan Turing (IV)

- The standard method for solving systems of linear equations in a computer is **Gaussian elimination**,
- **which is one of the most important numerical algorithms!!!**
- It is studied in depth in Chapter 9 of Nick Higham's, "Accuracy and Stability of Numerical Algorithms", where we found (pages 184-185)

"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper **"Rounding off errors in matrix processes"** *(Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308)..."*

My scientific connection with Alan Turing (IV)

- The standard method for solving systems of linear equations in a computer is **Gaussian elimination**,
- **which is one of the most important numerical algorithms!!!**
- It is studied in depth in Chapter 9 of Nick Higham's, "Accuracy and Stability of Numerical Algorithms", where we found (pages 184-185)

"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper **"Rounding off errors in matrix processes"** *(Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308)..."*

My scientific connection with Alan Turing (IV)

- The standard method for solving systems of linear equations in a computer is **Gaussian elimination**,
- **which is one of the most important numerical algorithms!!!**
- It is studied in depth in Chapter 9 of Nick Higham's, "Accuracy and Stability of Numerical Algorithms", where we found (pages 184-185)

"The experiences of Fox, Huskey, and Wilkinson prompted Turing to write a remarkable paper **"Rounding off errors in matrix processes"** *(Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308)..."*

ROUNDING-OFF ERRORS IN MATRIX PROCESSES

By A. M. TURING

(*National Physical Laboratory, Teddington, Middlesex*)

[Received 4 November 1947]

SUMMARY

A number of methods of solving sets of linear equations and inverting matrices are discussed. The theory of the rounding-off errors involved is investigated for some of the methods. In all cases examined, including the well-known 'Gauss elimination process', it is found that the errors are normally quite moderate: no exponential build-up need occur.

Included amongst the methods considered is a generalization of Choleski's method which appears to have advantages over other known methods both as regards accuracy and convenience. This method may also be regarded as a rearrangement of the elimination process.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

My scientific connection with Alan Turing (V)

"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 all contributions listed by N. Higham.

However, the history is not so clear...

"Turing coined the name "condition number" ... for measures of sensitivity of problems to error, and the acronym "LDU" 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 ..."

from J. F. Grcar, *John von Neumann's Analysis of Gaussian Elimination and the Origins of Modern Numerical Analysis*, SIAM Review, 53 (2011), pp. 607-682.

The goals of the talk

I pretend to explain

- these concepts at an introductory level and their role in modern Numerical Analysis;
- the fascinating historical context in which Alan Turing's paper was published;
- the work made by other authors (Hotelling, von Neumann, Goldstine, Wilkinson) on the rounding error analysis of Gaussian Elimination before and after Turing's paper;
- the unique spirit of Alan Turing's approach to the problem and its influence on modern Numerical Analysis.

The goals of the talk

I pretend to explain

- these concepts at an introductory level and their role in modern Numerical Analysis;
- the fascinating historical context in which Alan Turing's paper was published;
- the work made by other authors (Hotelling, von Neumann, Goldstine, Wilkinson) on the rounding error analysis of Gaussian Elimination before and after Turing's paper;
- the unique spirit of Alan Turing's approach to the problem and its influence on modern Numerical Analysis.

The goals of the talk

I pretend to explain

- these concepts at an introductory level and their role in modern Numerical Analysis;
- the fascinating historical context in which Alan Turing's paper was published;
- the work made by other authors (Hotelling, von Neumann, Goldstine, Wilkinson) on the rounding error analysis of Gaussian Elimination before and after Turing's paper;
- the unique spirit of Alan Turing's approach to the problem and its influence on modern Numerical Analysis.

The goals of the talk

I pretend to explain

- these concepts at an introductory level and their role in modern Numerical Analysis;
- the fascinating historical context in which Alan Turing's paper was published;
- the work made by other authors (Hotelling, von Neumann, Goldstine, Wilkinson) on the rounding error analysis of Gaussian Elimination before and after Turing's paper;
- the unique spirit of Alan Turing's approach to the problem and its influence on modern Numerical Analysis.

Outline

1. A very brief and simplified history of Gaussian elimination
2. Historical context of the paper by Alan Turing
3. Error bounds for Gaussian elimination
4. Remarks on Turing's 1948-paper
5. Conclusions

Outline

The origins of GE are found in form of solution of problems (*without including explicit statements of algorithmic rules*) in Jiuzhang Suanshu (Nine Chapters of Mathematical Art) in China (2000 years ago), Diophantus (3rd century), Aryabhata (Hindu, 5th century), Arabic texts, Renaissance European texts.

Treatments with explicit description of algorithmic rules can be organized in three periods: (from J. F. Grcar, *Mathematicians of Gaussian Elimination*, 58 (2011), pp. 782–792.)

- Period 1. **Schoolbook Elimination**: essentially the GE method currently presented in high-school textbooks. Introduced by Newton.

The three main periods of Gaussian elimination (GE) (I)

The origins of GE are found in form of solution of problems (*without including explicit statements of algorithmic rules*) in Jiuzhang Suanshu (Nine Chapters of Mathematical Art) in China (2000 years ago), Diophantus (3rd century), Aryabhata (Hindu, 5th century), Arabic texts, Renaissance European texts.

Treatments with explicit description of algorithmic rules can be organized in three periods: (from J. F. Grcar, *Mathematicians of Gaussian Elimination*, 58 (2011), pp. 782–792.)

- Period 1. **Schoolbook Elimination**: essentially the GE method currently presented in high-school textbooks. Introduced by Newton.

The three main periods of Gaussian elimination (GE) (I)

The origins of GE are found in form of solution of problems (*without including explicit statements of algorithmic rules*) in Jiuzhang Suanshu (Nine Chapters of Mathematical Art) in China (2000 years ago), Diophantus (3rd century), Aryabhata (Hindu, 5th century), Arabic texts, Renaissance European texts.

Treatments with explicit description of algorithmic rules can be organized in three periods: (from J. F. Grcar, *Mathematicians of Gaussian Elimination*, 58 (2011), pp. 782–792.)

- Period 1. **Schoolbook Elimination**: essentially the GE method currently presented in high-school textbooks. Introduced by Newton.

The three main periods of Gaussian elimination (GE) (II)

- Period 2. **Professional Elimination**: started by Legendre (1805) and Gauss (1809) for solving least-squares problems in an efficient way via the use of "human computers". They consider only positive definite linear systems in the form of normal equations

$$A^T A x = A^T b, \quad A \in \mathbb{R}^{m \times n} \;\; b \in \mathbb{R}^m .$$

These methods were improved by Doolittle (1881) (graphs and tables) and Cholesky (1924) adapting the method to "multiplying mechanical calculators".

Crout (1941) extended this type of procedures to general linear systems.

Most of these procedures are no longer in use and will not be considered in this talk.

The three main periods of Gaussian elimination (GE) (II)

- Period 2. **Professional Elimination**: started by Legendre (1805) and Gauss (1809) for solving least-squares problems in an efficient way via the use of "human computers". They consider only positive definite linear systems in the form of normal equations

$$A^T A x = A^T b, \quad A \in \mathbb{R}^{m \times n} \;\; b \in \mathbb{R}^m .$$

These methods were improved by Doolittle (1881) (graphs and tables) and Cholesky (1924) adapting the method to "multiplying mechanical calculators".

Crout (1941) extended this type of procedures to general linear systems.

Most of these procedures are no longer in use and will not be considered in this talk.

The three main periods of Gaussian elimination (GE) (II)

- Period 2. **Professional Elimination**: started by Legendre (1805) and Gauss (1809) for solving least-squares problems in an efficient way via the use of "human computers". They consider only positive definite linear systems in the form of normal equations

$$A^T Ax = A^T b, \quad A \in \mathbb{R}^{m\times n} \;\; b \in \mathbb{R}^m .$$

  These methods were improved by Doolittle (1881) (graphs and tables) and Cholesky (1924) adapting the method to "multiplying mechanical calculators".

  Crout (1941) extended this type of procedures to general linear systems.

  Most of these procedures are no longer in use and will not be considered in this talk.

- Period 3. **Modern Gaussian Elimination designed to be used in digital, electronic, programmable computers**: started by Von Neumann & Goldstine (1947) and **Turing** (1948), but established and analyzed in a definitive and solid way by Wilkinson (1961).

  Since then, research in GE algorithm and its analysis remains in continuous development: Reid (1971), Skeel (1979), Duff (1986), Higham (1980's, 90's), Demmel et al (1999), Grigori & Demmel & Xiang (2011), D & Molera (2012)...many many others

Next, we describe the essentials of GE in Periods 1 and 3.

The three main periods of Gaussian elimination (GE) (III)

- Period 3. **Modern Gaussian Elimination designed to be used in digital, electronic, programmable computers**: started by Von Neumann & Goldstine (1947) and **Turing** (1948), but established and analyzed in a definitive and solid way by Wilkinson (1961).

  Since then, research in GE algorithm and its analysis remains in continuous development: Reid (1971), Skeel (1979), Duff (1986), Higham (1980's, 90's), Demmel et al (1999), Grigori & Demmel & Xiang (2011), D & Molera (2012)...many many others

Next, we describe the essentials of GE in Periods 1 and 3.

The three main periods of Gaussian elimination (GE) (III)

- Period 3. **Modern Gaussian Elimination designed to be used in digital, electronic, programmable computers**: started by Von Neumann & Goldstine (1947) and **Turing** (1948), but established and analyzed in a definitive and solid way by Wilkinson (1961).

  Since then, research in GE algorithm and its analysis remains in continuous development: Reid (1971), Skeel (1979), Duff (1986), Higham (1980's, 90's), Demmel et al (1999), Grigori & Demmel & Xiang (2011), D & Molera (2012)...many many others

Next, we describe the essentials of GE in Periods 1 and 3.

Refreshing Gaussian Elimination from High-School with Newton

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

Refreshing Gaussian Elimination from High-School with Newton

Replace "equation(2)" by "equation(2) $-\ (-2)\times$equation(1)"

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

Refreshing Gaussian Elimination from High-School with Newton

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

Refreshing Gaussian Elimination from High-School with Newton

Replace "equation(3)" by "equation(3) $-\,3\times$equation(1)"

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

Refreshing Gaussian Elimination from High-School with Newton

$$
\begin{array}{rcrcrcrcr}
2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\
 & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\
 & & 12x_2 & & & - & 14x_4 & = & -4 \\
2x_1 & - & 3x_2 & - & 27x_3 & - & 3x_4 & = & -37
\end{array}
$$

Replace "equation(4)" by "equation(4) $- 1\times$equation(1)"

$$\begin{array}{rcrcrcrcr}
2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\
 & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\
 & & 12x_2 & & & - & 14x_4 & = & -4 \\
2x_1 & - & 3x_2 & - & 27x_3 & - & 3x_4 & = & -37
\end{array}$$

Refreshing Gaussian Elimination from High-School with Newton

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

Refreshing Gaussian Elimination from High-School with Newton

Replace "equation(3)" by "equation(3) $-$ $(-4)\times$equation(2)"

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

Refreshing Gaussian Elimination from High-School with Newton

$$\begin{array}{rcrcrcrcr}
2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\
 & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\
 & & & & 4x_3 & + & 2x_4 & = & 8 \\
 & - & 6x_2 & - & 26x_3 & - & 4x_4 & = & -46
\end{array}$$

Replace “equation(4)” by “equation(4) $-\ 2\times$equation(2)”

$$\begin{array}{rcrcrcrcr}
2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 \\
 & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 \\
 & & & & 4x_3 & + & 2x_4 & = & 8 \\
 & - & 6x_2 & - & 26x_3 & - & 4x_4 & = & -46
\end{array}$$

Refreshing Gaussian Elimination from High-School with Newton

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

Refreshing Gaussian Elimination from High-School with Newton

Replace "equation(4)" by "equation(4) $-\ (-7)\times$equation(3)"

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

Refreshing Gaussian Elimination from High-School with Newton

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

Refreshing Backward Substitution from High-School with Newton

$$
\begin{array}{ccccccccccc}
2x_1 & + & 3x_2 & - & x_3 & + & x_4 & = & 9 & & \\
 & - & 3x_2 & + & x_3 & + & 4x_4 & = & 3 & & \\
 & & & & 4x_3 & + & 2x_4 & = & 8 & & \\
 & & & & & & 2x_4 & = & 4 & \longrightarrow & \color{red}{x_4 = 2}
\end{array}
$$

Refreshing Backward Substitution from High-School with Newton

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

Refreshing Backward Substitution from High-School with Newton

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

Refreshing Backward Substitution from High-School with Newton

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

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

Comments on High-School-G-E (...developed by Newton)

- My son told me one year ago (he was 16), after learning to run GE (by hand) on small systems of equations, that
  1. GE is boring;
  2. GE takes long long time (in part because it requires to write down a lot of equations);
  3. It is easy to commit errors that spoil the whole solution.
- He was right!!
- **GE elimination as established by Newton is not efficient to solve by hand or via mechanical or electronic calculators large (18×18) systems of equations**.
- This is the reason why Gauss and others developed methods for organizing the operations of GE in better ways that avoid the need of writing equations during the Professional Elimination Period.

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

$$\Downarrow$$

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

$$\Downarrow$$

$$\boxed{Ax = b}$$

From High-School to modern GE: Matrix Factorization (I)

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

$$\Downarrow$$

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

$$\Downarrow$$

$$Ax = b$$

From High-School to modern GE: Matrix Factorization (I)

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

$$\Downarrow$$

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

$$\Downarrow$$

$$Ax = b$$

From High-School to modern GE: Matrix Factorization (I)

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

$$\Downarrow$$

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

$$\Downarrow$$

$$Ax = b$$

From High-School to modern GE: Matrix Factorization (I)

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

$$\Downarrow$$

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

$$\Downarrow$$

$$\boxed{A\,x = b}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

  (let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \qquad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \to & eq(2) - (-2) \times eq(1) \\ eq(3) & \to & eq(3) - \ \ (3) \times eq(1) \\ eq(4) & \to & eq(4) - \ \ (1) \times eq(1) \\ eq(3) & \to & eq(3) - (-4) \times eq(2) \\ eq(4) & \to & eq(4) - \ \ (2) \times eq(2) \\ eq(4) & \to & eq(4) - (-7) \times eq(3) \end{array} \qquad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Recall the last system produced by GE

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

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{aligned} eq(2) &\rightarrow eq(2) - (-2) \times eq(1) \\ eq(3) &\rightarrow eq(3) - \ (3) \times eq(1) \\ eq(4) &\rightarrow eq(4) - \ (1) \times eq(1) \end{aligned}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array}$$

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

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{aligned} eq(2) &\rightarrow eq(2) - (-2) \times eq(1) \\ eq(3) &\rightarrow eq(3) - (3) \times eq(1) \\ eq(4) &\rightarrow eq(4) - (1) \times eq(1) \\ eq(3) &\rightarrow eq(3) - (-4) \times eq(2) \\ eq(4) &\rightarrow eq(4) - (2) \times eq(2) \\ eq(4) &\rightarrow eq(4) - (-7) \times eq(3) \end{aligned} \qquad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \to & eq(2) - (-2) \times eq(1) \\ eq(3) & \to & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \to & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \to & eq(3) - (-4) \times eq(2) \\ eq(4) & \to & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \to & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b)

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & \boxed{eq(2) - (-2) \times eq(1)} \\ eq(3) & \rightarrow & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \boxed{-2} & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \to & eq(2) - (-2) \times eq(1) \\ eq(3) & \to & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \to & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \to & eq(3) - (-4) \times eq(2) \\ eq(4) & \to & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \to & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (3) \times eq(1) \\ eq(4) & \rightarrow & \boxed{eq(4) - (1) \times eq(1)} \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ \boxed{1} & 2 & -7 & 1 \end{bmatrix}$$

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - (1) \times eq(1) \\ eq(3) & \rightarrow & \boxed{eq(3) - (-4) \times eq(2)} \\ eq(4) & \rightarrow & eq(4) - (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & \boxed{-4} & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

(let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - (1) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

From High-School to modern GE: Matrix Factorization (II)

- The *replacement operations on equations* that we performed translate to *replacement operations on rows* of

$$A = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \quad \text{to get} \quad U = \begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

  (let us forget for a while the vector b).

- These replacement operation have been

$$\begin{array}{lcl} eq(2) & \rightarrow & eq(2) - (-2) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - \quad (3) \times eq(1) \\ eq(4) & \rightarrow & eq(4) - \quad (1) \times eq(1) \\ eq(3) & \rightarrow & eq(3) - (-4) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - \quad (2) \times eq(2) \\ eq(4) & \rightarrow & eq(4) - (-7) \times eq(3) \end{array} \quad \text{construt} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}$$

And now a miracle!!

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- This is the famous **LU Factorization** of a matrix.
- It was introduced first by von Neumann and Goldstine in their celebrated **"Numerical inverting of matrices of high order"** (*Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*).
- It was also introduced (later and not independently) by Turing in his 1948 landmark paper with its current name.
- Efficient methods for computing **Triangular matrix factorizations** were considered among of the Top Ten Algorithms of 20th Century.

And now a miracle!!

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- This is the famous **LU Factorization** of a matrix.
- It was introduced first by von Neumann and Goldstine in their celebrated **"Numerical inverting of matrices of high order"** (*Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*).
- It was also introduced (later and not independently) by Turing in his 1948 landmark paper with its current name.
- Efficient methods for computing **Triangular matrix factorizations** were considered among of the Top Ten Algorithms of 20th Century.

And now a miracle!!

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- This is the famous **LU Factorization** of a matrix.
- It was introduced first by von Neumann and Goldstine in their celebrated **"Numerical inverting of matrices of high order"** (*Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*).
- It was also introduced (later and not independently) by Turing in his 1948 landmark paper with its current name.
- Efficient methods for computing **Triangular matrix factorizations** were considered among of the Top Ten Algorithms of 20th Century.

And now a miracle!!

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- This is the famous **LU Factorization** of a matrix.
- It was introduced first by von Neumann and Goldstine in their celebrated **"Numerical inverting of matrices of high order"** (*Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*).
- It was also introduced (later and not independently) by Turing in his 1948 landmark paper with its current name.
- Efficient methods for computing **Triangular matrix factorizations** were considered among of the Top Ten Algorithms of 20th Century.

And now a miracle!!

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- This is the famous **LU Factorization** of a matrix.
- It was introduced first by von Neumann and Goldstine in their celebrated **"Numerical inverting of matrices of high order"** (*Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*).
- It was also introduced (later and not independently) by Turing in his 1948 landmark paper with its current name.
- Efficient methods for computing **Triangular matrix factorizations** were considered among of the Top Ten Algorithms of 20th Century.

Modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A

$$A = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$Ly = b$$

- Step 3: Solve via backward substitution the upper triangular system

$$Ux = y$$

This approach was suggested first by Turing in his 1948-paper.

Modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A

$$A = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$Ly = b$$

- Step 3: Solve via backward substitution the upper triangular system

$$Ux = y$$

This approach was suggested first by Turing in his 1948-paper.

Modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A

$$A = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$Ly = b$$

- Step 3: Solve via backward substitution the upper triangular system

$$Ux = y$$

This approach was suggested first by Turing in his 1948-paper.

Modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A

$$A = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$Ly = b$$

- Step 3: Solve via backward substitution the upper triangular system

$$Ux = y$$

This approach was suggested first by Turing in his 1948-paper.

Modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A

$$A = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$Ly = b$$

- Step 3: Solve via backward substitution the upper triangular system

$$Ux = y$$

This approach was suggested first by Turing in his 1948-paper.

Algorithm for computing LU factorization of a matrix

INPUT: $A \in \mathbb{R}^{n \times n}$
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

Cost: $2n^3/3$ operations.

Computing LU factorization is easy on a computer

Algorithm for computing LU factorization of a matrix

INPUT: $A \in \mathbb{R}^{n\times n}$
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

No longer boring...It is fascinating!!

Cost: $2n^3/3$ operations.

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$(A \equiv) A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$(A \equiv) A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$(A \equiv) A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge “errors” when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$A^{(1)} = \begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix} \longrightarrow \begin{bmatrix} 6 & 21 & -3 & -11 \\ -4 & -9 & 3 & 2 \\ 2 & 3 & -1 & 1 \\ 2 & -3 & -27 & -3 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge “errors” when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge “errors” when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge “errors” when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

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

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge "errors" when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$U \equiv A^{(4)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & -12 & -5 \\ 0 & 0 & 0 & 1/15 \end{bmatrix}$$

Modern GE: Partial Pivoting (I)

The LU factorization (GE) algorithm explained above may produce huge “errors” when it is implemented on a computer.

In practice, it is necessary to permute the rows of A, equivalently the equations, as follows

$$U \equiv A^{(4)} = \begin{bmatrix} 6 & 21 & -3 & -11 \\ 0 & -10 & -26 & 2/3 \\ 0 & 0 & -12 & -5 \\ 0 & 0 & 0 & 1/15 \end{bmatrix}$$

This strategy is known as partial pivoting and yields

Modern GE: Partial Pivoting (II)

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

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- Row permutations of a matrix change the L and U factors in a nontrivial way and this indicates why error analysis of GE is extremely difficult.
- Partial pivoting produces an L_P factor with entries of absolute values less than or equal to one (fundamental property).

Modern GE: Partial Pivoting (II)

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

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- Row permutations of a matrix change the L and U factors in a nontrivial way and this indicates why error analysis of GE is extremely difficult.
- Partial pivoting produces an L_P factor with entries of absolute values less than or equal to one (fundamental property).

Modern GE: Partial Pivoting (II)

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

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- Row permutations of a matrix change the L and U factors in a nontrivial way and this indicates why error analysis of GE is extremely difficult.
- Partial pivoting produces an L_P factor with entries of absolute values less than or equal to one (fundamental property).

Modern GE: Partial Pivoting (II)

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

$$\underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ -4 & -9 & 3 & 2 \\ 6 & 21 & -3 & -11 \\ 2 & -3 & -27 & -3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 3 & -4 & 1 & 0 \\ 1 & 2 & -7 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 & 1 \\ 0 & -3 & 1 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 2 \end{bmatrix}}_{U}$$

- Row permutations of a matrix change the L and U factors in a nontrivial way and this indicates why error analysis of GE is extremely difficult.
- Partial pivoting produces an L_P factor with entries of absolute values less than or equal to one (fundamental property).

The definitive modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A with **partial pivoting**

$$PA = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$L\,y = Pb$$

- Step 3: Solve via backward substitution the upper triangular system

$$U\,x = y$$

The definitive modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A with **partial pivoting**

$$PA = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$L\,y = Pb$$

- Step 3: Solve via backward substitution the upper triangular system

$$U\,x = y$$

The definitive modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A with **partial pivoting**

$$PA = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$L\,y = Pb$$

- Step 3: Solve via backward substitution the upper triangular system

$$U\,x = y$$

The definitive modern GE: Three steps for the solution

Starting from

$$Ax = b$$

modern GE performs three steps.

- Step 1: Compute LU factorization of A with **partial pivoting**

$$PA = LU$$

- Step 2: Solve via forward substitution the lower triangular system

$$L\,y = Pb$$

- Step 3: Solve via backward substitution the upper triangular system

$$U\,x = y$$

Outline

Three very famous papers on error analysis of GE in the 1940s (I)

- Harold Hotelling, **"Some new methods in matrix calculation"**, *The Annals of Mathematical Statistics, 14 (1943), pp. 1-34*.
- John von Neumann and Herman Goldstine, **"Numerical inverting of matrices of high order"**, *Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*.
- Alan Turing, **"Rounding off errors in matrix processes"**, *Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308*.

The paper by von Neumann and Goldstine is often considered as the first paper of modern Numerical Analysis, where "modern" has the sense of "analyzing methods to be used on digital, electronic, programmable computers".

Three very famous papers on error analysis of GE in the 1940s (I)

- Harold Hotelling, **"Some new methods in matrix calculation"**, *The Annals of Mathematical Statistics, 14 (1943), pp. 1-34*.
- John von Neumann and Herman Goldstine, **"Numerical inverting of matrices of high order"**, *Bulletin of the American Mathematical Society, 53 (1947), pp. 1021-1099*.
- Alan Turing, **"Rounding off errors in matrix processes"**, *Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308*.

The paper by von Neumann and Goldstine is often considered as the first paper of modern Numerical Analysis, where "modern" has the sense of "analyzing methods to be used on digital, electronic, programmable computers".

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by "hand" and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling's results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by “hand” and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling’s results motivated von Neumann, Goldstine, and Turing.

- They are motivated by the specific question: Will the best known method for solving linear systems by "hand" and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling's results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by "hand" and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling's results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by “hand” and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling's results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by “hand” and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling’s results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (II)

- They are motivated by the specific question: Will the best known method for solving linear systems by “hand” and/or by electrical/mechanical calculators be accurate on modern computers?
- **The three papers were written before modern computers existed,**
- but projects for constructing computers got underway during this period.
- They are motivated by the general question: New computers will offer a huge power of computation but, will the numerical methods used up to now be accurate and efficient on modern computers?
- Hotelling obtained error bounds for GE that increase exponentially with the size of the matrix ($\sim 4^n$). This would make GE useless in practice even for very small matrices and led to general pessimism on GE.
- Hotelling’s results motivated von Neumann, Goldstine, and Turing.

Three very famous papers on error analysis of GE in the 1940s (III)

- The three papers were written by top researchers, who considered the problem very important from an applied point of view and from a fundamental point of view, since
- GE was the first numerical algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established.
- None of the papers solved completely the problem. It was too formidable even for Von Neumann or Turing.
- The analysis of GE accepted nowadays came with Wilkinson (1961).
- A complete rigorous solution still remains as an open problem.

Three very famous papers on error analysis of GE in the 1940s (III)

- The three papers were written by top researchers, who considered the problem very important from an applied point of view and from a fundamental point of view, since
- GE was the first numerical algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established.
- None of the papers solved completely the problem. It was too formidable even for Von Neumann or Turing.
- The analysis of GE accepted nowadays came with Wilkinson (1961).
- A complete rigorous solution still remains as an open problem.

Three very famous papers on error analysis of GE in the 1940s (III)

- The three papers were written by top researchers, who considered the problem very important from an applied point of view and from a fundamental point of view, since
- GE was the first numerical algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established.
- None of the papers solved completely the problem. It was too formidable even for Von Neumann or Turing.
- The analysis of GE accepted nowadays came with Wilkinson (1961).
- A complete rigorous solution still remains as an open problem.

- The three papers were written by top researchers, who considered the problem very important from an applied point of view and from a fundamental point of view, since
- GE was the first numerical algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established.
- None of the papers solved completely the problem. It was too formidable even for Von Neumann or Turing.
- The analysis of GE accepted nowadays came with Wilkinson (1961).
- A complete rigorous solution still remains as an open problem.

Three very famous papers on error analysis of GE in the 1940s (III)

- The three papers were written by top researchers, who considered the problem very important from an applied point of view and from a fundamental point of view, since
- GE was the first numerical algorithm to be subjected to rounding error analysis and the fundamentals of rounding error analysis had to be established.
- None of the papers solved completely the problem. It was too formidable even for Von Neumann or Turing.
- The analysis of GE accepted nowadays came with Wilkinson (1961).
- A complete rigorous solution still remains as an open problem.

A few words on the Authors (I)

John von Neumann
(1903-1957)

Alan Turing
(1912-1954)

Two of the most important Mathematicians of the History

Harold Hotelling
(1895-1973)

He was a mathematical statistician and an influential economic theorist. Associate Prof. of Maths. at Stanford (1927-31), faculty of Columbia (1931-46), and Prof. of Mathematical Statistics at University of North Carolina (1946-1895). He received the North Carolina Award for contributions to science (1972).

He introduced Hotelling's T-square distribution and canonical correlation analysis in Statistics.

He made pioneering studies of non-convexity in economics.

Herman Goldstine
(1913-2004)

He was a mathematician (PhD in Maths, U. Chicago, 1936) and computer scientist. He was awarded the USA National Medal of Science (1983).

He joined the Army in WWII and he persuaded USA Army to build ENIAC (Electronic Numerical Integrator And Computer): the first electronic computer starting to work in 1946 up to 1955.

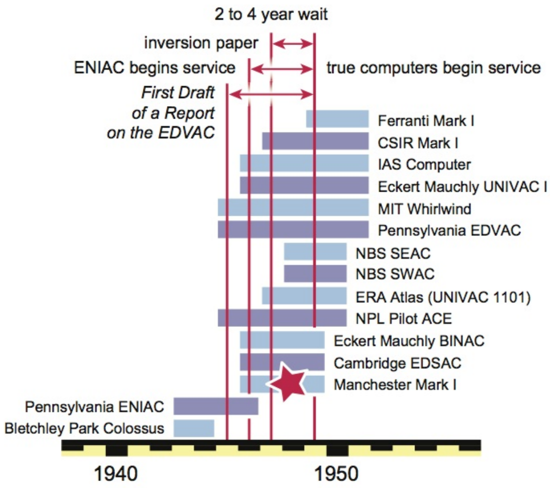
He was program manager of ENIAC.

Herman Goldstine (1913-2004)

ENIAC was thousand of times faster than electro-mechanical machines. It was programmable, but no way existed to issue orders at electronic speed (modern programs), so ENIAC had to be configured with patch cords and rotary switches for each task.

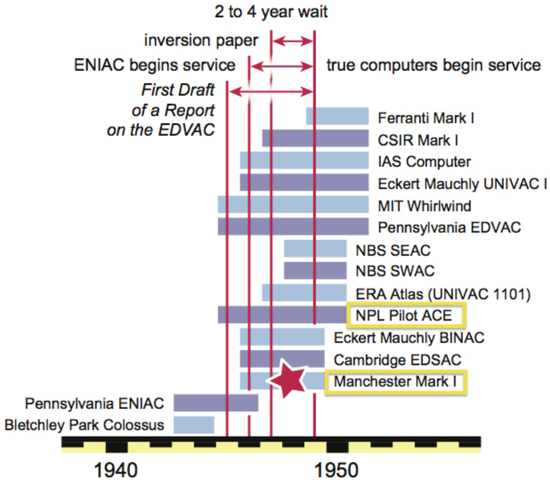
Goldstine involved von Neumann (1944) in planning ENIAC's successor resulting in the famous von Neumann's 1945 report *"First Draft of a Report on the EDVAC"* on how to build a modern computer, and in a long and fruitful collaboration.

Some projects in 1940-50's for constructing modern computers (I)



from J. F. Grcar, *John von Neumann's Analysis of Gaussian Elimination and the Origins of Modern Numerical Analysis*, SIAM Review, 53 (2011), pp. 607-682.

Some projects in 1940-50's for constructing modern computers (I)



from J. F. Grcar, *John von Neumann's Analysis of Gaussian Elimination and the Origins of Modern Numerical Analysis*, SIAM Review, 53 (2011), pp. 607-682.

Some projects in 1940-50's for constructing modern computers (II)

- **Turing** was involved in two of these projects

1. **NPL Pilot ACE** (National Physical Laboratory Pilot Automatic Computing Engine, England). Turing worked at NPL from 1945-1948 and in this period he became interest in rounding errors in GE.

2. **Manchester Mark I** (University of Manchester, England). This was the first digital, electronic, programmable computer that worked in the world in April, 1949.

Some projects in 1940-50's for constructing modern computers (II)

- **Turing** was involved in two of these projects

1. **NPL Pilot ACE** (National Physical Laboratory Pilot Automatic Computing Engine, England). Turing worked at NPL from 1945-1948 and in this period he became interest in rounding errors in GE.

2. **Manchester Mark I** (University of Manchester, England). This was the first digital, electronic, programmable computer that worked in the world in April, 1949.

- **Turing** was involved in two of these projects
  1. **NPL Pilot ACE** (National Physical Laboratory Pilot Automatic Computing Engine, England). Turing worked at NPL from 1945-1948 and in this period he became interest in rounding errors in GE.
  2. **Manchester Mark I** (University of Manchester, England). This was the first digital, electronic, programmable computer that worked in the world in April, 1949.

- **Von Neumann** led the **Computer project** at the **Institute of Advanced Studies at Princeton (USA)** and **Goldstine** was also working there. In this period they became interested in rounding errors in GE.

Outline

Lloyd N. Trefethen.
Numerical analyst from Oxford University.
Current SIAM President

"...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..."

from L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM 1997.

Numbers and arithmetic in current computers (I)

- Computers can only represent a finite subset of the real numbers, which is called the set of **floating point** numbers, denoted by $\mathbb{F}$. This fact produces errors.
- $\mathbb{F}$ is not closed under basic arithmetic operations $(+, -, \times, /)$, but when they are performed on a computer, they must give another number of $\mathbb{F}$. This fact produces further errors.
- These two facts are encapsulated into the **axioms of rounding error analysis.**

Numbers and arithmetic in current computers (I)

- Computers can only represent a finite subset of the real numbers, which is called the set of **floating point** numbers, denoted by $\mathbb{F}$. This fact produces errors.
- $\mathbb{F}$ is not closed under basic arithmetic operations $(+, -, \times, /)$, but when they are performed on a computer, they must give another number of $\mathbb{F}$. This fact produces further errors.
- These two facts are encapsulated into the **axioms of rounding error analysis.**

Numbers and arithmetic in current computers (I)

- Computers can only represent a finite subset of the real numbers, which is called the set of **floating point** numbers, denoted by $\mathbb{F}$. This fact produces errors.
- $\mathbb{F}$ is not closed under basic arithmetic operations $(+, -, \times, /)$, but when they are performed on a computer, they must give another number of $\mathbb{F}$. This fact produces further errors.
- These two facts are encapsulated into the **axioms of rounding error analysis.**

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!!!** ($\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$ in IEEE double precision).

Axiom 2. 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, that may not be in $\mathbb{F}$.

Numbers and arithmetic in current computers (II)

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!!!** ($\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$ in IEEE double precision).

Axiom 2. 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, that may not be in $\mathbb{F}$.

Numbers and arithmetic in current computers (III)

- Axioms of floating point arithmetic were introduced by Wilkinson (1960),
- but the **original idea of establishing simple axioms** for rounding error analysis **goes back to von Neumann and Goldstine's 1947-paper** (axioms for fixed point arithmetic).
- The analysis in Turing's 1948-paper does not include axioms for rounding errors and is very far from current error analyses.
- Algorithms on a computer are combinations of (many) $\{+, -, \times, /\}$ operations. The idea is to combine via the axioms the errors in all these operations to produce a final relative error in the computed magnitude.

Numbers and arithmetic in current computers (III)

- Axioms of floating point arithmetic were introduced by Wilkinson (1960),
- but the **original idea of establishing simple axioms** for rounding error analysis **goes back to von Neumann and Goldstine's 1947-paper** (axioms for fixed point arithmetic).
- The analysis in Turing's 1948-paper does not include axioms for rounding errors and is very far from current error analyses.
- Algorithms on a computer are combinations of (many) $\{+, -, \times, /\}$ operations. The idea is to combine via the axioms the errors in all these operations to produce a final relative error in the computed magnitude.

- Axioms of floating point arithmetic were introduced by Wilkinson (1960),
- but the **original idea of establishing simple axioms** for rounding error analysis **goes back to von Neumann and Goldstine's 1947-paper** (axioms for fixed point arithmetic).
- The analysis in Turing's 1948-paper does not include axioms for rounding errors and is very far from current error analyses.
- Algorithms on a computer are combinations of (many) $\{+, -, \times, /\}$ operations. The idea is to combine via the axioms the errors in all these operations to produce a final relative error in the computed magnitude.

Numbers and arithmetic in current computers (III)

- Axioms of floating point arithmetic were introduced by Wilkinson (1960),
- but the **original idea of establishing simple axioms** for rounding error analysis **goes back to von Neumann and Goldstine's 1947-paper** (axioms for fixed point arithmetic).
- The analysis in Turing's 1948-paper does not include axioms for rounding errors and is very far from current error analyses.
- Algorithms on a computer are combinations of (many) $\{+, -, \times, /\}$ operations. The idea is to combine via the axioms the errors in all these operations to produce a final relative error in the computed magnitude.

We offer a simplified approach that pays attention only to the **fundamental feature of GE which motivates that Hotelling found an error bound that increases exponentially with n**.

INPUT: $A \in \mathbb{R}^{n\times n}$
OUTPUT: L stored in strictly lower triangular part of A
U stored in upper triangular part of A

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

Direct error analysis of GE produces exponential bounds in n (I)

We offer a simplified approach that pays attention only to the **fundamental feature of GE which motivates that Hotelling found an error bound that increases exponentially with n.**

INPUT: $A \in \mathbb{R}^{n \times n}$
OUTPUT: L stored in strictly lower triangular part of A
U stored in upper triangular part of A

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

We offer a simplified approach that pays attention only to the **fundamental feature of GE which motivates that Hotelling found an error bound that increases exponentially with n.**

INPUT: $A^{(1)} = A \in \mathbb{R}^{n\times n}$
OUTPUT: L stored in strictly lower triangular part of A
U stored in upper triangular part of A

for $k = 1 : n-1$
for $i = k+1 : n$
for $j = k+1 : n$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}$$

end
end
end

We offer a simplified approach that pays attention only to the **fundamental feature of GE which motivated that Hotelling found an error bound that increases exponentially with n**.

INPUT: $A^{(1)} = A \in \mathbb{R}^{n\times n}$
OUTPUT: L stored in strictly lower triangular part of A
U stored in upper triangular part of A

for $k = 1 : n-1$
  for $i = k+1 : n$
    for $j = k+1 : n$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \boxed{\frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}}$$

    end
  end
end

Direct error analysis of GE produces exponential bounds in n (II)

$$\boxed{\frac{a_{ik}^{(k)}\, a_{kj}^{(k)}}{a_{kk}^{(k)}}}$$

- Assume that at stage k of GE the computed entries $\widehat{a}_{pq}^{(k)}$ satisfy

$$\left| \frac{\widehat{a}_{pq}^{(k)} - a_{pq}^{(k)}}{a_{pq}^{(k)}} \right| \leq \mathbf{e_k}, \quad \text{for all } k \leq p, q \leq n,$$

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!!

- Then, as we learnt when we were very young

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

where 2nd-order terms are discarded.

$$\boxed{\frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}}$$

- Assume that at stage k of GE the computed entries $\widehat{a}_{pq}^{(k)}$ satisfy

$$\left| \frac{\widehat{a}_{pq}^{(k)} - a_{pq}^{(k)}}{a_{pq}^{(k)}} \right| \leq \mathbf{e_k}, \quad \text{for all } k \leq p, q \leq n,$$

i.e., $\mathrm{e_k}$ is an upper bound on **the maximum relative error at the kth stage of GE**. This is what we want to determine!!

- Then, as we learnt when we were very young

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

where 2nd-order terms are discarded.

Direct error analysis of GE produces exponential bounds in n (II)

- **and there are still more errors in computing!!**

$$\widehat{a}_{ij}^{(k+1)} = \text{computed}\left(\widehat{a}_{ij}^{(k)} - \frac{\widehat{a}_{ik}^{(k)}\widehat{a}_{kj}^{(k)}}{\widehat{a}_{kk}^{(k)}}\right).$$

- Therefore, a bound on the **maximum relative error in $(k+1)$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 $n \times n$ matrices

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

Direct error analysis of GE produces exponential bounds in n (II)

- **and there are still more errors in computing!!**

$$\widehat{a}_{ij}^{(k+1)} = \text{computed}\left(\widehat{a}_{ij}^{(k)} - \frac{\widehat{a}_{ik}^{(k)}\widehat{a}_{kj}^{(k)}}{\widehat{a}_{kk}^{(k)}}\right).$$

- Therefore, a bound on the **maximum relative error in $(k+1)$th stage of GE**, i.e., $\mathbf{e_{k+1}}$ satisfies

$$\boxed{\mathbf{e_{k+1}} \gtrsim 3\mathbf{e_k},}$$

- and, since $\mathbf{e_1} = \mathbf{u} \approx 10^{-16}$ and **GE performs $(n-1)$ stage transitions** for $n \times n$ matrices

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

Direct error analysis of GE produces exponential bounds in n (II)

- **and there are still more errors in computing!!**

$$\widehat{a}_{ij}^{(k+1)} = \text{computed}\left(\widehat{a}_{ij}^{(k)} - \frac{\widehat{a}_{ik}^{(k)}\widehat{a}_{kj}^{(k)}}{\widehat{a}_{kk}^{(k)}}\right).$$

- Therefore, a bound on the **maximum relative error in $(k+1)$th stage of GE**, i.e., $\mathbf{e_{k+1}}$ satisfies

$$\boxed{\mathbf{e_{k+1}} \gtrsim 3\mathbf{e_k},}$$

- and, since $\mathbf{e_1} = \mathbf{u} \approx 10^{-16}$ and **GE performs $(n-1)$ stage transitions** for $n \times n$ matrices

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

$\mathbf{n}$	$\mathbf{e_n} \approx 3^{n-1} \cdot 10^{-16}$
10	$2 \cdot 10^{-12}$
20	$1.2 \cdot 10^{-7}$
30	$6.9 \cdot 10^{-3}$
40	$4.1 \cdot 10^{+2}$
50	$2.4 \cdot 10^{+7}$

Direct error analysis of GE produces exponential bounds in n (IV)

- This huge error bound led Hotelling (1943) to state: *"The rapidity with which this increases with n is a caution against relying on...elimination method..."* This created a general pessimism with respect the use of GE in practice.
- Much more sophisticated rounding error analyses were needed to restore the confidence in GE. Starting with von Neumann and Goldstine's (1947) and Turing's (1948) papers, the analysis accepted today was given by Wilkinson in 1961 in the key paper
- James Wilkinson, **"Error analysis of direct methods of matrix inversion"**, Journal of the Association for Computing Machinery, 8 (1961), pp. 281-330.

Direct error analysis of GE produces exponential bounds in n (IV)

- This huge error bound led Hotelling (1943) to state: *"The rapidity with which this increases with n is a caution against relying on...elimination method..."* This created a general pessimism with respect the use of GE in practice.
- Much more sophisticated rounding error analyses were needed to restore the confidence in GE. Starting with von Neumann and Goldstine's (1947) and Turing's (1948) papers, the analysis accepted today was given by Wilkinson in 1961 in the key paper
- James Wilkinson, **"Error analysis of direct methods of matrix inversion"**, Journal of the Association for Computing Machinery, 8 (1961), pp. 281-330.

- This huge error bound led Hotelling (1943) to state: *"The rapidity with which this increases with n is a caution against relying on...elimination method..."* This created a general pessimism with respect the use of GE in practice.
- Much more sophisticated rounding error analyses were needed to restore the confidence in GE. Starting with von Neumann and Goldstine's (1947) and Turing's (1948) papers, the analysis accepted today was given by Wilkinson in 1961 in the key paper
- James Wilkinson, **"Error analysis of direct methods of matrix inversion"**, Journal of the Association for Computing Machinery, 8 (1961), pp. 281-330.

A few words on James Wilkinson

James Hardy Wilkinson (1919-1986)

A Cambridge-trained English mathematician. **He worked as Turing's assistant at NPL (1946-48). He is considered as the founder of modern rounding error analysis of algorithms** by using systematically **backward error analysis**.

Backward Errors in Gaussian elimination: Rigorous result

Theorem (Wilkinson, 1961)

Let $A \in \mathbb{R}^{n\times n}$ be **any nonsingular matrix**, *let $b \in \mathbb{R}^n$, and let*

$$\widehat{x}$$

be the approximate solution of

$$Ax = b$$

computed by **GE with partial pivoting in a computer with unit roundoff $\mathbf{u}$**. *Then*

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

where

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

is the **growth factor of Gaussian elimination**. *Here $A^{(1)} := A, A^{(2)}, \ldots, A^{(n)}$ are the matrices appearing in the Gaussian elimination process.*

Backward Errors in Gaussian elimination in plain words

Theorem (Wilkinson, 1961)

*Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by GE with partial pivoting (**GEPP**) in a computer **with unit roundoff** $\mathbf{u}$. Then*

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho_n},$$

i.e., **the computed solution is the exact solution of a nearby linear system** *(if ρ_n is not large!!).*

This is an instance of the "mantra" that every numerical analyst working in 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"**.

Backward Errors in Gaussian elimination in plain words

Theorem (Wilkinson, 1961)

*Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by GE with partial pivoting (***GEPP***) in a computer* **with unit roundoff $\mathbf{u}$**. *Then*

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

i.e., **the computed solution is the exact solution of a nearby linear system** *(if ρ_n is not large!!).*

This is an instance of the "mantra" that every numerical analyst working in 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"**.

... but in GEPP the backward error depends on the growth factor...

Example (Growth factor)

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

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

$$\rho = \frac{11.76}{10} = 1.1760$$

... but in GEPP the backward error depends on the growth factor...

Example (Growth factor)

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

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

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

Lemma (Wilkinson 1961)

Let $A \in \mathbb{R}^{n \times n}$ be a 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 some matrices.*

...and the backward error of GEPP can be as large as...

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff $\mathbf{u}$**. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n \leq 3 \cdot n^3 \cdot 2^{(n-1)} \cdot \mathbf{u}.$$

The right-most bound is larger than 1, and then useless, for very small matrices since $\mathbf{u} = 2^{-53} \approx 10^{-16}$.

Key comment: *"Despite of the fact that* **GEPP** *does not guarantee tiny backward errors,* **it is the standard algorithm for solving in modern computers linear systems of equations** *and also despite of the fact that there are other (more expensive) algorithms that guarantee always tiny backward errors."*

...and the backward error of GEPP can be as large as...

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff** $\mathbf{u}$. Then

$$(A + \Delta A)\widehat{x} = b, \quad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n \leq 3 \cdot n^3 \cdot 2^{(n-1)} \cdot \mathbf{u}.$$

The right-most bound is larger than 1, and then useless, for very small matrices since $\mathbf{u} = 2^{-53} \approx 10^{-16}$.

Key comment: *"Despite of the fact that* **GEPP** *does not guarantee tiny backward errors,* **it is the standard algorithm for solving in modern computers linear systems of equations** *and also despite of the fact that there are other (more expensive) algorithms that guarantee always tiny backward errors."*

One of the major unsolved problems in Numerical Analysis

"...the growth factor is almost invariable found to be small ($\rho_n \leq 50$). ***Explaining this fact remains one of the major unsolved problems in Numerical Analysis****."*

from N. Higham, "Accuracy and Stability of Numerical Algorithms", (SIAM, 2002).

Turing, Von Neumann-Goldstine, and backward error analysis

- Some modern texts and papers indicate that Von Neumann & Goldstine and Turing introduced in their papers backward error analysis.
- In my opinion, this is not completely true. Von Neumann & Goldstine, and Turing mention backward errors (without the name), but in a rather marginal way, and do not realize the importance of this concept.
- **Backward error analysis is today one of the most fundamental and powerful ideas in Numerical Analysis.**

- Some modern texts and papers indicate that Von Neumann & Goldstine and Turing introduced in their papers backward error analysis.
- In my opinion, this is not completely true. Von Neumann & Goldstine, and Turing mention backward errors (without the name), but in a rather marginal way, and do not realize the importance of this concept.
- **Backward error analysis is today one of the most fundamental and powerful ideas in Numerical Analysis.**

Turing, Von Neumann-Goldstine, and backward error analysis

- Some modern texts and papers indicate that Von Neumann & Goldstine and Turing introduced in their papers backward error analysis.
- In my opinion, this is not completely true. Von Neumann & Goldstine, and Turing mention backward errors (without the name), but in a rather marginal way, and do not realize the importance of this concept.
- **Backward error analysis is today one of the most fundamental and powerful ideas in Numerical Analysis.**

From backward to forward errors: The condition number

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff** $\mathbf{u}$. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

Bounding the difference between the **exact solution**, x, and the **computed solution**, $\widehat{x}$, becomes a mathematical problem of **perturbation theory**.

Theorem (Wilkinson, 1963)

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\|\Delta A\|_\infty}{\|A\|_\infty}$$

$$\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$$

Definition (The (very famous!!!) condition number of a matrix)

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

From backward to forward errors: The condition number

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff** $\mathbf{u}$. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

Bounding the difference between the **exact solution**, x, and the **computed solution**, $\widehat{x}$, becomes a mathematical problem of **perturbation theory**.

Theorem (Wilkinson, 1963)

$$\begin{aligned} \frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \\ &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n) \end{aligned}$$

Definition (The (very famous!!!) condition number of a matrix)

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

From backward to forward errors: The condition number

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff $\mathbf{u}$**. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

Bounding the difference between the **exact solution**, x, and the **computed solution**, $\widehat{x}$, becomes a mathematical problem of **perturbation theory**.

Theorem (Wilkinson, 1963)

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

Definition (The (very famous!!!) condition number of a matrix)

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

From backward to forward errors: The condition number

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff $\mathbf{u}$**. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

Bounding the difference between the **exact solution**, x, and the **computed solution**, $\widehat{x}$, becomes a mathematical problem of **perturbation theory**.

Theorem (Wilkinson, 1963)

$$\begin{aligned} \frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \\ &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n) \end{aligned}$$

Definition (The (very famous!!!) condition number of a matrix)

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

From backward to forward errors: The condition number

Let $\widehat{x}$ be the approximate solution of $Ax = b$ computed by **GEPP** in a computer **with unit roundoff** $\mathbf{u}$. Then

$$(A + \Delta A)\widehat{x} = b, \qquad \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n,$$

Bounding the difference between the **exact solution**, x, and the **computed solution**, $\widehat{x}$, becomes a mathematical problem of **perturbation theory**.

Theorem (Wilkinson, 1963)

$$\begin{aligned} \frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \\ &\lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n) \end{aligned}$$

Definition (The (very famous!!!) condition number of a matrix)

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

Who discovered "the condition number"? (personal opinion)

- Turing in his 1948 paper gives an "unusual definition" of condition number of a matrix and shows in an "unusual way" its relationship with the variation of the solution of a linear system under perturbations of the data. The essentials are here!!
- Turing gave the name "condition number".
- Von Neumann and Goldstine in their 1947 paper use the condition number in their error bounds, but they do not show any perturbation inequality involving the condition number.
- First rigorous perturbation results on condition numbers: Bauer (1959) (for inverses) and Wilkinson (1963) (for linear systems).

Who discovered "the condition number"? (personal opinion)

- Turing in his 1948 paper gives an "unusual definition" of condition number of a matrix and shows in an "unusual way" its relationship with the variation of the solution of a linear system under perturbations of the data. The essentials are here!!
- Turing gave the name "condition number".
- Von Neumann and Goldstine in their 1947 paper use the condition number in their error bounds, but they do not show any perturbation inequality involving the condition number.
- First rigorous perturbation results on condition numbers: Bauer (1959) (for inverses) and Wilkinson (1963) (for linear systems).

Who discovered "the condition number"? (personal opinion)

- Turing in his 1948 paper gives an "unusual definition" of condition number of a matrix and shows in an "unusual way" its relationship with the variation of the solution of a linear system under perturbations of the data. The essentials are here!!
- Turing gave the name "condition number".
- Von Neumann and Goldstine in their 1947 paper use the condition number in their error bounds, but they do not show any perturbation inequality involving the condition number.
- First rigorous perturbation results on condition numbers: Bauer (1959) (for inverses) and Wilkinson (1963) (for linear systems).

Who discovered “the condition number”? (personal opinion)

- Turing in his 1948 paper gives an “unusual definition” of condition number of a matrix and shows in an “unusual way” its relationship with the variation of the solution of a linear system under perturbations of the data. The essentials are here!!
- Turing gave the name “condition number”.
- Von Neumann and Goldstine in their 1947 paper use the condition number in their error bounds, but they do not show any perturbation inequality involving the condition number.
- First rigorous perturbation results on condition numbers: Bauer (1959) (for inverses) and Wilkinson (1963) (for linear systems).

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \qquad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x}$ and $\kappa(A) \approx 1.3 \cdot 10^{16}$

"Huge errors in the solution are diabolically correlated to give tiny residuals."

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \quad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x}$ and $\kappa(A) \approx 1.3 \cdot 10^{16}$

"Huge errors in the solution are diabolically correlated to give tiny residuals."

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \quad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x}$ and $\kappa(A) \approx 1.6 \cdot 10^{16}$.

"Huge errors in the solution are diabolically correlated to give tiny residuals."

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \quad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x} \qquad$ and $\qquad \kappa(A) \approx 1.6 \cdot 10^{16}$.

"Huge errors in the solution are diabolically correlated to give tiny residuals."

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \quad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x} \qquad$ and $\qquad \kappa(A) \approx 1.6 \cdot 10^{16}$.

"Huge errors in the solution are diabolically correlated to give tiny residuals."

Example backward vs forward errors

Computed by GEPP, $\widehat{x}$, and exact, x, solutions of $Ax = b$ satisfy

- $(A + \Delta A)\widehat{x} = b, \quad \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n, \qquad (\mathbf{u} \approx 10^{-16})$
- $\dfrac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \|A\|_\infty \, \|A^{-1}\|_\infty \dfrac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq \|A\|_\infty \, \|A^{-1}\|_\infty \, (3 \cdot n^3 \cdot \mathbf{u} \cdot \boldsymbol{\rho}_n)$

Example

$$A = \begin{bmatrix} 10^{16} & -10^8/5 & 1/10 \\ 10^{16}/3 & 10^8 & -1/10 \\ 10^{16}/3 & -10^8/5 & 1 \end{bmatrix} \quad \text{and} \quad b = A \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} = 0.14 \quad \text{and} \quad \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty} = 1.3 \cdot 10^{-16}$$

Explanation:

- $A\widehat{x} - b = -(\Delta A)\widehat{x} \qquad$ and $\qquad \kappa(A) \approx 1.6 \cdot 10^{16}$.

“Huge errors in the solution are diabolically correlated to give tiny residuals.”

Outline

Why did Turing write this paper?

- At NPL, Turing and collaborators (Fox, Goodwin, and Wilkinson) were asked to solve a linear system of 18 equations.
- Turing's collaborators used GE with complete pivoting on desk electronic calculators.
- Turing thought that it would be a failure, because, as a consequence of Hotelling's bounds, he shared the general pessimism existing in mid 1940's on GE, but
- the relative residual for the computed solution was

$$\frac{\|A\widehat{x} - b\|_\infty}{\|b\|_\infty} \approx \textbf{unit roundoff}$$

  for the computed solution $\widehat{x}$, and
- **Turing believed based on a few numerical tests that GE was the right method of solution!!**. Although

Why did Turing write this paper?

- At NPL, Turing and collaborators (Fox, Goodwin, and Wilkinson) were asked to solve a linear system of 18 equations.
- Turing's collaborators used GE with complete pivoting on desk electronic calculators.
- Turing thought that it would be a failure, because, as a consequence of Hotelling's bounds, he shared the general pessimism existing in mid 1940's on GE, but
- the relative residual for the computed solution was

$$\frac{\|A\widehat{x} - b\|_\infty}{\|b\|_\infty} \approx \textbf{unit roundoff}$$

for the computed solution $\widehat{x}$, and

- **Turing believed based on a few numerical tests that GE was the right method of solution!!**. Although

Why did Turing write this paper?

- At NPL, Turing and collaborators (Fox, Goodwin, and Wilkinson) were asked to solve a linear system of 18 equations.
- Turing's collaborators used GE with complete pivoting on desk electronic calculators.
- Turing thought that it would be a failure, because, as a consequence of Hotelling's bounds, he shared the general pessimism existing in mid 1940's on GE, but
- the relative residual for the computed solution was

$$\frac{\|A\widehat{x} - b\|_\infty}{\|b\|_\infty} \approx \textbf{unit roundoff}$$

  for the computed solution $\widehat{x}$, and
- **Turing believed based on a few numerical tests that GE was the right method of solution!!**. Although

Why did Turing write this paper?

- At NPL, Turing and collaborators (Fox, Goodwin, and Wilkinson) were asked to solve a linear system of 18 equations.
- Turing's collaborators used GE with complete pivoting on desk electronic calculators.
- Turing thought that it would be a failure, because, as a consequence of Hotelling's bounds, he shared the general pessimism existing in mid 1940's on GE, but
- the relative residual for the computed solution was

$$\frac{\|A\widehat{x} - b\|_\infty}{\|b\|_\infty} \approx \textbf{unit roundoff}$$

for the computed solution $\widehat{x}$, and

- **Turing believed based on a few numerical tests that GE was the right method of solution!!**. Although

- At NPL, Turing and collaborators (Fox, Goodwin, and Wilkinson) were asked to solve a linear system of 18 equations.
- Turing's collaborators used GE with complete pivoting on desk electronic calculators.
- Turing thought that it would be a failure, because, as a consequence of Hotelling's bounds, he shared the general pessimism existing in mid 1940's on GE, but
- the relative residual for the computed solution was

$$\frac{\|A\widehat{x} - b\|_\infty}{\|b\|_\infty} \approx \textbf{unit roundoff}$$

for the computed solution $\widehat{x}$, and

- **Turing believed based on a few numerical tests that GE was the right method of solution!!**. Although

Turing was sure that GE can fail

The best known method for the solution of linear equations is Gauss's elimination method. This is the method almost universally taught in schools. It 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. 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

from first page in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

Turing's unique insight: although error bounds of GE with pivoting may increase exponentially with the size for some matrices, these matrices are very rare and GEPP can be used with confidence. This insight has influenced in depth Numerical Analysis.

Turing got this idea with the very limited numerical experience accumulated in the 40's and based on the non-optimal error bounds he proved.

Turing was sure that GE can fail

The best known method for the solution of linear equations is Gauss's elimination method. This is the method almost universally taught in schools. It 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. 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

from first page in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

Turing's unique insight: although error bounds of GE with pivoting may increase exponentially with the size for some matrices, these matrices are very rare and GEPP can be used with confidence. This insight has influenced in depth Numerical Analysis.

Turing got this idea with the very limited numerical experience accumulated in the 40's and based on the non-optimal error bounds he proved.

Turing was sure that GE can fail

> The best known method for the solution of linear equations is Gauss's elimination method. This is the method almost universally taught in schools. It 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. 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

from first page in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

Turing's unique insight: although error bounds of GE with pivoting may increase exponentially with the size for some matrices, these matrices are very rare and GEPP can be used with confidence. This insight has influenced in depth Numerical Analysis.

Turing got this idea with the very limited numerical experience accumulated in the 40's and based on the non-optimal error bounds he proved.

case where **A** is reduced to a unit matrix. We assume that in the calculation of each quantity

$$\mathbf{A}_{ij}^{(r-1)} - \frac{\mathbf{A}_{rj}^{(r-1)}\mathbf{A}_{ir}^{(r-1)}}{\mathbf{A}_{rr}^{(r-1)}},$$

an error of at most ϵ 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 $\mathbf{A}_{ir}^{(r-1)}/\mathbf{A}_{rr}^{(r-1)}$ will have to depend on the values of the $\mathbf{A}_{rj}^{(r-1)}$. Likewise, we

from p. 302 in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

No computer, neither present or past, can guarantee such an error bound a priori in finite precision arithmetic.

However, the results presented by Turing were fundamental in late 1940's for restoring the confidence in GE as the best numerical method for solving general systems of linear equations.

Turing's analysis is nonstandard

> case where **A** is reduced to a unit matrix. We assume that in the calculation of each quantity
>
> $$\mathrm{A}_{ij}^{(r-1)} - \frac{\mathrm{A}_{rj}^{(r-1)}\mathrm{A}_{ir}^{(r-1)}}{\mathrm{A}_{rr}^{(r-1)}},$$
>
> an error of at most ϵ 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 $\mathrm{A}_{ir}^{(r-1)}/\mathrm{A}_{rr}^{(r-1)}$ will have to depend on the values of the $\mathrm{A}_{rj}^{(r-1)}$. Likewise, we

from p. 302 in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

No computer, neither present or past, can guarantee such an error bound a priori in finite precision arithmetic.

However, the results presented by Turing were fundamental in late 1940's for restoring the confidence in GE as the best numerical method for solving general systems of linear equations.

Turing's analysis is nonstandard

case where **A** is reduced to a unit matrix. We assume that in the calculation of each quantity

$$\mathrm{A}_{ij}^{(r-1)} - \frac{\mathrm{A}_{rj}^{(r-1)}\mathrm{A}_{ir}^{(r-1)}}{\mathrm{A}_{rr}^{(r-1)}},$$

an error of at most ϵ 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 $\mathrm{A}_{ir}^{(r-1)}/\mathrm{A}_{rr}^{(r-1)}$ will have to depend on the values of the $\mathrm{A}_{rj}^{(r-1)}$. Likewise, we

from p. 302 in A. Turing, *"Rounding off errors in matrix processes"*, Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287-308.

No computer, neither present or past, can guarantee such an error bound a priori in finite precision arithmetic.

However, the results presented by Turing were fundamental in late 1940's for restoring the confidence in GE as the best numerical method for solving general systems of linear equations.

A few words on Turing's bound for $\|x - \widehat{x}\|_\infty \,/\, \|x\|_\infty$

- Turing's bounds are expressed in terms of the unknown quantity ϵ.
- With the unrealistic ideal assumption $\epsilon = \mathbf{u}\,\|A\|_\infty$,
- Turing's bound becomes a non-optimal bound of the type

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right)^2 (p(n) \cdot \mathbf{u}),$$

with $p(n)$ a polynomial in n that does not depend on the growth factor.

- A trivial change in the analysis would produce

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right) (p(n) \cdot \mathbf{u}),$$

which has the standard form, but does not include the growth factor.

A few words on Turing's bound for $\|x - \widehat{x}\|_\infty \,/\, \|x\|_\infty$

- Turing's bounds are expressed in terms of the unknown quantity ϵ.
- With the unrealistic ideal assumption $\epsilon = \mathbf{u}\,\|A\|_\infty$,
- Turing's bound becomes a non-optimal bound of the type

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right)^2 (p(n) \cdot \mathbf{u}),$$

  with $p(n)$ a polynomial in n that does not depend on the growth factor.
- A trivial change in the analysis would produce

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right) (p(n) \cdot \mathbf{u}),$$

  which has the standard form, but does not include the growth factor.

A few words on Turing's bound for $\|x - \widehat{x}\|_\infty \,/\, \|x\|_\infty$

- Turing's bounds are expressed in terms of the unknown quantity ϵ.
- With the unrealistic ideal assumption $\epsilon = \mathbf{u}\, \|A\|_\infty$,
- Turing's bound becomes a non-optimal bound of the type

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right)^2 (p(n) \cdot \mathbf{u}),$$

with $p(n)$ a polynomial in n that does not depend on the growth factor.

- A trivial change in the analysis would produce

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty \, \|A^{-1}\|_\infty\right) (p(n) \cdot \mathbf{u}),$$

which has the standard form, but does not include the growth factor.

A few words on Turing's bound for $\|x - \widehat{x}\|_\infty \,/\, \|x\|_\infty$

- Turing's bounds are expressed in terms of the unknown quantity ϵ.
- With the unrealistic ideal assumption $\epsilon = \mathbf{u}\,\|A\|_\infty$,
- Turing's bound becomes a non-optimal bound of the type

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty\,\|A^{-1}\|_\infty\right)^2 (p(n)\cdot \mathbf{u}),$$

  with $p(n)$ a polynomial in n that does not depend on the growth factor.
- A trivial change in the analysis would produce

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \lesssim \left(\|A\|_\infty\,\|A^{-1}\|_\infty\right) (p(n)\cdot \mathbf{u}),$$

  which has the standard form, but does not include the growth factor.

Outline

1. A very brief and simplified history of Gaussian elimination
2. Historical context of the paper by Alan Turing
3. Error bounds for Gaussian elimination
4. Remarks on Turing's 1948-paper
5. **Conclusions**

Conclusions

- Turing faced a very important and hard problem with a perspective that no other mathematician had used before.
- His conclusions on GEPP are still valid and were essential in late 1940's to adopt GE as the standard method for solving systems of equations on modern computers.
- A complete understanding of the behavior of rounding errors in GEPP remains today as an open problem.
- GE is the subject of considerable research activity today:
  1. Grigori, Demmel, Xiang (SIMAX 2011). CALU a communication avoiding optimal LU factorization with new pivoting strategy.
  2. D. , Molera (IMAJNA 2012). Widest class of structured matrices for which $\|A\|_\infty \|A^{-1}\|_\infty$ can be removed from the error bound.

Conclusions

- Turing faced a very important and hard problem with a perspective that no other mathematician had used before.
- His conclusions on GEPP are still valid and were essential in late 1940's to adopt GE as the standard method for solving systems of equations on modern computers.
- A complete understanding of the behavior of rounding errors in GEPP remains today as an open problem.
- GE is the subject of considerable research activity today:
  1. Grigori, Demmel, Xiang (SIMAX 2011). CALU a communication avoiding optimal LU factorization with new pivoting strategy.
  2. D. , Molera (IMAJNA 2012). Widest class of structured matrices for which $\|A\|_\infty \|A^{-1}\|_\infty$ can be removed from the error bound.

Conclusions

- Turing faced a very important and hard problem with a perspective that no other mathematician had used before.
- His conclusions on GEPP are still valid and were essential in late 1940's to adopt GE as the standard method for solving systems of equations on modern computers.
- A complete understanding of the behavior of rounding errors in GEPP remains today as an open problem.
- GE is the subject of considerable research activity today:
    1. Grigori, Demmel, Xiang (SIMAX 2011). CALU a communication avoiding optimal LU factorization with new pivoting strategy.
    2. D. , Molera (IMAJNA 2012). Widest class of structured matrices for which $\|A\|_\infty \|A^{-1}\|_\infty$ can be removed from the error bound.

Conclusions

- Turing faced a very important and hard problem with a perspective that no other mathematician had used before.
- His conclusions on GEPP are still valid and were essential in late 1940's to adopt GE as the standard method for solving systems of equations on modern computers.
- A complete understanding of the behavior of rounding errors in GEPP remains today as an open problem.
- GE is the subject of considerable research activity today:
  1. Grigori, Demmel, Xiang (SIMAX 2011). CALU a communication avoiding optimal LU factorization with new pivoting strategy.
  2. D. , Molera (IMAJNA 2012). Widest class of structured matrices for which $\|A\|_\infty \|A^{-1}\|_\infty$ can be removed from the error bound.