HISTORY OF GAUSSIAN ELIMINATION

Can you think of a mathematical concept which is so simple on the surface that it is taught to junior high school students, which is so practical in nature that it is used thousands or perhaps hundreds of thousands of times each and every day, which is so old that it dates back to at least 200 BC, which has been the source of continued study and research for more than 2000 years, but which is still not completely understood? It is not clear how many concepts satisfy these conditions, but there is at least one:

> The numerical determination of the solution to a system of n linear algebraic equations in n unknowns by means of what we commonly refer to as *Gaussian elimination*.

With the proliferation of small inexpensive personal computers and hand-held calculators, there currently is an increased awareness concerning numerical procedures. Among all numerical techniques, those devised to solve linear systems have an extremely rich history. Much of this is due to some of the seeming paradoxes encountered. It is indeed a shock the first time you sit down at a small personal computer (or even a large non-personal one) and enter your own program to solve a system of equations only to find that either the solution is so far from being correct that it is nonsensical or that no solution at all is produced when it is known that one exists. You check your program for logical errors but you find none. You go back to your text book that discussed linear systems[1] to see if you made a theoretical error. Again you find none. You may even go so far as to question the electronic circuitry, but to no avail. So why did the numbers come out all wrong? You have just realized that what had seemed so simple in theory is in fact not simple after all. It is precisely this scenario that has captured so much interest for such a long period of time.

Let's start at the beginning, or at least as far back as recorded history will take us. The earliest recorded analysis of linear systems of equations is found in the ancient Chinese book *Chiu-chang Suan-shu (Nine Chapters on Arithmetic)* estimated to have been compiled some time around 200 B.C. In the beginning of Chapter VIII of the *Nine Chapters on Arithmetic* there appears a problem of the following form.

> Three sheafs of a good crop, two sheafs of a mediocre crop, and one sheaf of a bad crop are sold for 39 dou. Two sheafs of good, three mediocre, and one bad are sold for 34 dou; and one good, two mediocre, and three bad are sold for 26 dou. What is the price received for each sheaf of a good crop, each sheaf of a mediocre crop, and each sheaf of a bad crop?

Today we would formulate this problem as three equations in three unknowns by writing

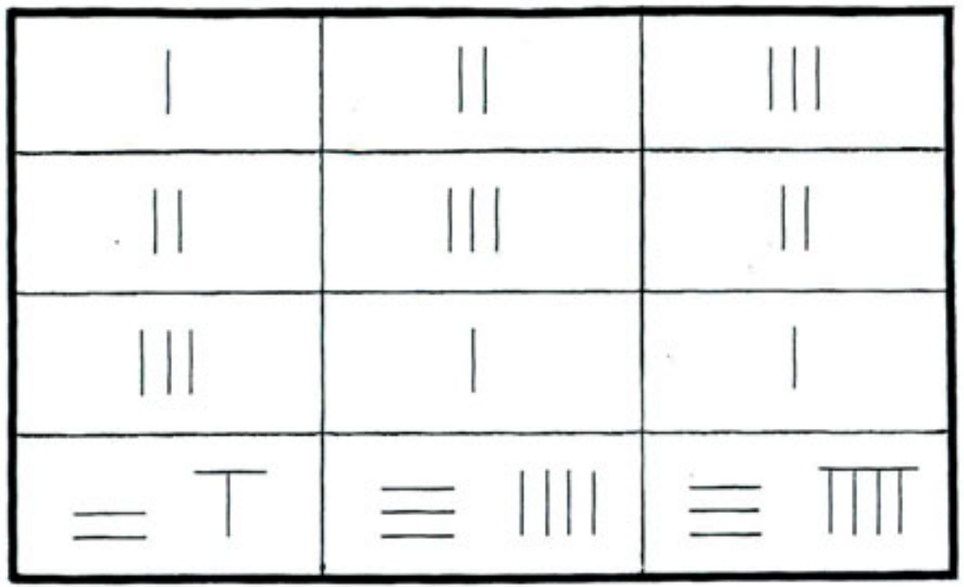
$$\begin{aligned} 3x + 2y + z &= 39 \\ 2x + 3y + z &= 34 \\ x + 2y + 3z &= 26 \end{aligned}$$

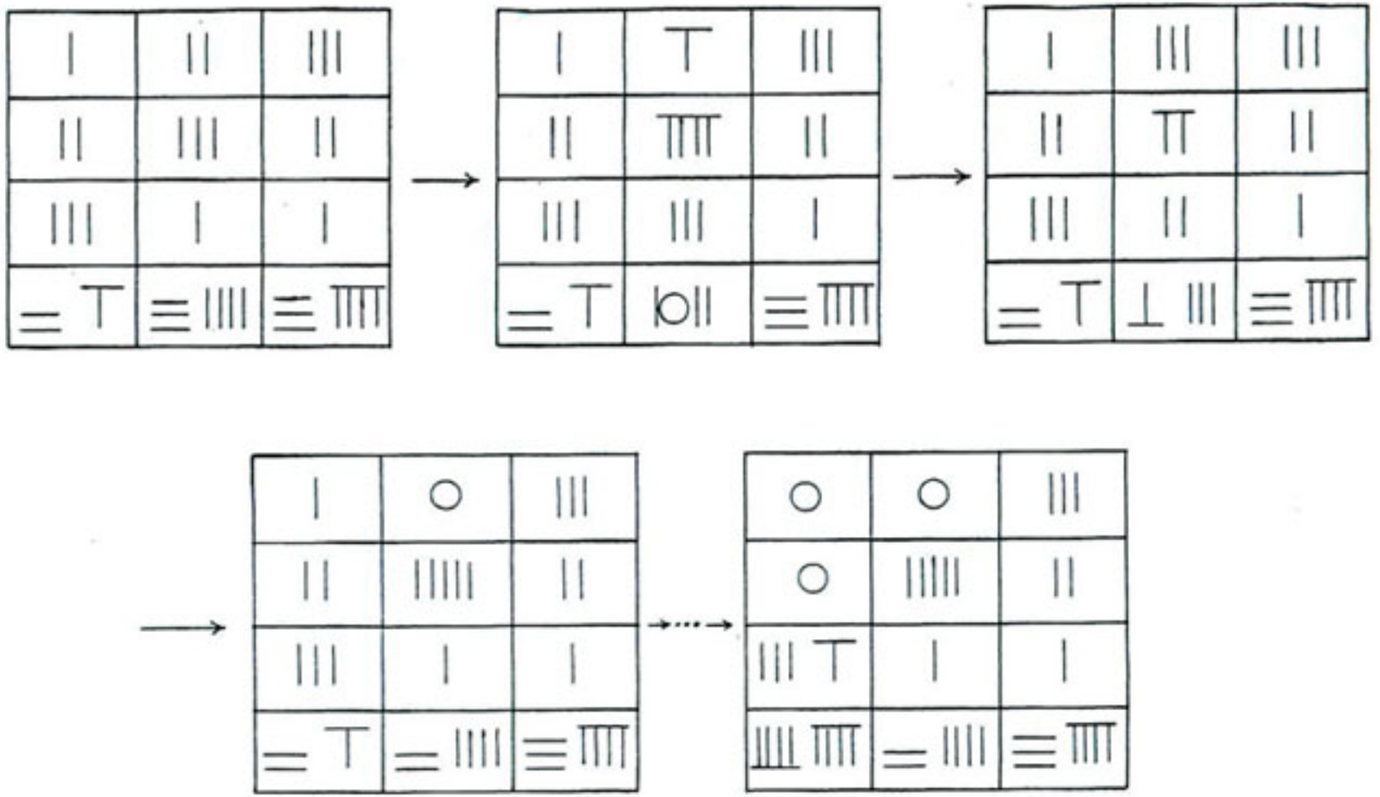
where x, y, and z represent the price for one sheaf of a good, mediocre, and bad crop, respectively.

There are many techniques that can be employed to extract a solution, however, the ancient Chinese saw right to the heart of the problem and realized that the most efficient technique is by using a rectangular array of colored bamboo rods placed in columns on a "counting board" as shown above.

The digits 1,2,3,4,5,6,7,8,9 were represented by the rod patterns $|$, $||$, $|||$, $||||$, $|||||$, $\top$, $\top|$, $\top||$, $\top|||$, $\top||||$ as well as the rod patterns

$-$, $=$, $\equiv$, $\equiv\!-$, $\perp$, $\perp\!-$, $\perp\!=$, $\perp\!\equiv$, $\perp\!\equiv\!-$. The two sets of rod patterns were used in alternate positions. Negative numbers were denoted with black rods while positive numbers were denoted with red rods.[2] For example, $\perp\top\!\!\top\ =\ ||$ represents 6822. An empty position represents zero so that ambiguities can arise unless one physically puts something down to represent 0. By manipulating the columns of this array of rods according to prescribed "rules of thumb" successive steps in the computation were performed.





CHINESE COUNTING BOARD USED IN GAUSSIAN ELIMINATION

Translate the rods to numerals and you will see that the end result is the equivalent system

$$\begin{aligned} 3x + 2y + z &= 39 \\ 5y + z &= 24 \\ 36z &= 99 \end{aligned}$$

which is easily solved by a backward substitution process.

This ancient concept is the essence of what is now generally known as the method of *Gaussian elimination.*[3]

Gaussian elimination as practiced today differs from the Chinese only in the sense that we now write our equations in rows rather than columns. The Chinese recognized the elimination method as a systematic way of transforming one system into a simpler but equivalent[4] system by successively eliminating unknowns to arrive at a system which is easily solvable. The elimination process relies on three simple operations by which to transform a system to an equivalent system:

(1) Interchange any two equations. (This says that it should make no difference in what order the equations are listed. Theoretically this is true but in numerical computations this can be grossly false. We will return to this point later.)

(2) Replace any equation by a nonzero multiple of itself.

(3) Replace any equation by a combination of itself plus a multiple of any other equation.

In the sequel, the discussion will be limited to the most common case of a square nonsingular system:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

At each step of the basic algorithm, the strategy is to focus on one position along the diagonal of the coefficients on the left hand side (this position is called the *pivotal position*), and then to eliminate all terms below the pivotal position using the three elementary operations. The coefficient in the pivotal position is called the *pivot*. Only nonzero numbers are allowed to be pivots but since interchanges are possible, there can be many choices for pivots on each step. The object is to create a *triangular* system

$$\begin{aligned} t_{11}x_1 + t_{12}x_2 + \ldots + t_{1n}x_n &= c_1 \\ t_{22}x_2 + \ldots + t_{2n}x_n &= c_2 \\ &\ldots \\ t_{nn}x_n &= c_n \end{aligned}$$

which can be easily solved by the *back substitution process:*

$$x_n = t_{nn}^{-1}c_n$$

followed by

$$x_i = t_{ii}^{-1}\left(c_i - \sum_{k=i-1}^{n} t_{ik}x_k\right) \text{ for } i = n-1, n-2, \ldots, 1.$$

We follow the ancient Chinese in that we don't explicitly write the symbols "x", "y", "z", and "=" since the only manipulations performed are on the numbers themselves. For example, the Chinese example might be solved in a contemporary elementary text as follows. (The pivotal element is always circled.)

$$\left[\begin{array}{ccc|c} \textcircled{3} & 2 & 1 & 39 \\ 2 & 3 & 1 & 34 \\ 1 & 2 & 3 & 26 \end{array}\right] \longrightarrow \left[\begin{array}{ccc|c} \textcircled{1} & 2 & 3 & 26 \\ 2 & 3 & 1 & 34 \\ 3 & 2 & 1 & 39 \end{array}\right]$$

$$\longrightarrow \begin{matrix} (R_2 - 2R_1) \\ (R_3 - 3R_1) \end{matrix} \left[\begin{array}{ccc|c} 1 & 2 & 3 & 26 \\ 0 & \textcircled{-1} & -5 & -18 \\ 0 & -4 & -8 & -39 \end{array}\right] \longrightarrow (R_3 - 4R_2) \left[\begin{array}{ccc|c} 1 & 2 & 3 & 26 \\ 0 & -1 & -5 & -18 \\ 0 & 0 & \textcircled{12} & 33 \end{array}\right]$$

Back substitution yields: $z = 33/12 = 11/4$, $y = (-1)(-18 + 5z) = 17/4$, $x = 26 - 2y - 3z = 37/4$.

In their use of array manipulation the ancient Chinese possessed the seed which might have germinated into a genuine theory of matrices. Unfortunately, in the year 213 B.C., emperor

Shih Hoang-ti ordered that "all books be burned and all scholars be buried." It is presumed that the emperor wanted all knowledge and written records to begin with him and his regime. The edict was carried out and it will never be known how much knowledge may have been lost. The *Nine Chapters* was compiled on the basis of fragments which survived.

More than a millennium had to pass before further progress was documented. The Chinese counting board and its use in solving linear systems eventually found its way to Japan. Seki Kowa (1642-1708), whom the Japanese consider the greatest mathematician that their country has produced, carried forward the Chinese principles involving "rule of thumb" elimination methods on arrays of numbers. His understanding of the elementary operations used in the Chinese elimination process led him to formulate the concept of what we now call the determinant[5] sometime before the year 1683. In addition to inventing the determinant, Seki Kowa also anticipated the fundamental concepts and array operations which today form the basis of the algebra of matrices.

About the same time, (somewhere between 1678 and 1693) Gottfried W. Leibniz (1646-1716) from Germany was independently developing his own notion of the determinant and applications of array manipulation to solve linear systems of equations. It appears that Leibniz's early work dealt with only three equations in three unknowns whereas Seki Kowa gave a general treatment for n equations in n unknowns. It seems that Seki Kowa and Gottfried Leibniz (although not in the same form or notations) both developed what later became known as "Cramer's Rule". These men also had something else in common. Their ideas concerning solving linear systems were never adopted by the mathematical community of the time and their discoveries quickly faded into oblivion.

In 1750 Gabriel Cramer (1704-1752) from Geneva, Switzerland rediscovered and published the rule that now bears his name. Cramer's work was responsible for a revived interest in determinants and systems of equations. Cramer's rule met with instant success and quickly found its way into the textbooks and classrooms of Europe. Sir Thomas Muir reports that one passed or failed the exams in the schools of public service in France according to one's knowledge and understanding of Cramer's rule.

The determinant went on to be the subject of great study and much was written on the subject.[6] There is some irony here. The basic elimination method for solving a linear system helped to foster the once heralded and much studied concept of the determinant. The determinant was considered to be an important concept and quite applicable in dealing with linear systems. Cramer's rule was once considered not only a standard method but in fact a reasonable way of solving a nonsingular system.[7] Two hundred years ago, very few people attempted to solve systems of magnitude much greater than (5 x 5). Moreover, the electronic computer that you can purchase in your local shopping center today is beyond anything that mathematicians of even the early part of our century could have dreamed of in their wildest fantasies. Things have indeed changed. Except for a few simple theoretical properties, the role of the determinant in contemporary mathematics is very minor whereas the original concept of Gaussian elimination (with some modern modifications) is now considered to be one of the most important algorithms of applied mathematics.

No one seems to know who first introduced Gaussian elimination in Europe, but it probably was not Carl F. Gauss (1777-1855). It has been reported that Gauss himself referred to the general elimination method as being "commonly known". Gauss was known for performing incredible computational feats and the elimination method which bears his name was certainly contained in his bag of tricks and he made extensive use of it. Gauss modified the basic elimination method to suit his specialized purposes and developed a practical algorithm for the positive definite systems that arise from the "normal equations" associated with the linear least squares problem. Herman Goldstine in *A History of Numerical Analysis From the 16th Through the 19th Century* claims that in his work on reducing quadratic forms,[8] Gauss, in effect, was the first to use elimination techniques in order to produce the factorization of a positive definite matrix into LDL^T where L is lower triangular with 1's on the diagonal and D is a diagonal matrix with positive diagonal entries.[9] This is now known as the Cholesky decomposition. The importance of matrix factorizations in solving linear systems will be discussed later in this article.

Only minor modifications occurred during the 19th century. One of these modifications carries the name "Gauss-Jordan method". In this method, the coefficients above the pivot, as well as those below, are eliminated at each step, thereby circumventing the need for the back substitution process. As was the case with determinants, the Gauss-Jordan process proved to be a step backwards. That is, it is much less efficient than basic Gaussian elimination with back substitution. It is easy to show that the Gauss-Jordan method requires approximately $(n^3/2)$ operations[10] whereas Gaussian elimination with back substitution requires only around $(n^3/3)$ operations. This means that the Gauss-Jordan method requires about *50% more effort* than Gaussian elimination with back substitution. For small order systems, these results may not be significant.

Curiously, it seems that the Gauss-Jordan method is also inappropriately named, just as the basic Gaussian elimination method may be a misnomer. Camille Jordan (1838-1922), from France, was a great mathematician and contributed much to linear algebra, but it appears that he had little to do with the Gauss-Jordan method because the process was only first described in the third edition (but not the first or second) of one of his texts which was prepared after his death. A. S. Householder in *The Theory of Matrices in Numerical Analysis* credits this algorithm to B. J. Clasen in 1888.

By the middle of the nineteenth century, German mathematicians stood head and shoulders above other nationalities in analysis and geometry. Algebra, on the other hand, was not a German monopoly. British mathematicians such as Arthur Cayley (1821-1895), Augustus DeMorgan (1806-1871), James J. Sylvester (1814-1897), George Boole (1815-1864), together with some others had a profound impact.

Although array manipulation in the form of determinants had long been in practice, matrix[11] theory did not evolve along with the theory of determinants. Because of this, Morris Kline in *Mathematical Thought From Ancient to Modern Times* says that ". . . the subject of matrices was well developed before it was created". It was Cayley's work with linear transformations that motivated him to single out the matrix as a separate entity (distinct from the notion of a determinant) and to define algebraic operations between matrices. In an 1855 paper, Cayley introduced some basic ideas which were presented mainly to simplify notations involved with linear transformations. In 1857 (Phil. Trans. London, vol. 148, pp. 17-37, Coll. Works, vol. 2, pp. 475-496) Cayley expanded on his original ideas and wrote *A Memoir On the Theory of Matrices* which laid the foundations for modern matrix algebra. This is generally credited for being the birth of the subject.

Cayley used the composition of two transformations, each defined by a linear system of equations, to formulate the notion of the *matrix product* as follows. Let T_1 and T_2 be transformations carrying (x,y) into (u,v) and (u,v) into (x'y'), respectively, which are defined by

$$T_1: \begin{aligned} ax + by &= u \\ cx + dy &= v \end{aligned} \qquad \text{and} \qquad T_2: \begin{aligned} Au + Bv &= x' \\ Cu + Dv &= y'. \end{aligned}$$

The composition of T_1 followed by T_2 maps (x,y) into (x'y') as

$$T_2T_1: \begin{aligned} (Aa + Bc)x + (Ab + Bd)y &= x' \\ (Ca + Dc)x + (Cb + Dd)y &= y' \end{aligned}$$

Cayley observed that if T_1 and T_2 are represented by retaining only the array of coefficients, that is as matrices

$$[T_1] = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{and} \quad [T_2] = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

then the effect of the composition T_2T_1 could be represented by the matrix

$$[T_2T_1] = \left[\begin{array}{c|c} Aa + Bc & Ab + Bd \\ \hline Ca + Dc & Cb + Dd \end{array}\right]$$

and that this matrix can be composed from a simple combination of matrices $[T_1]$ and $[T_2]$.

All of the elementary properties such as noncommutativity for matrix multiplication, matrix addition, scalar multiplication, singularity (the inability to reverse the effect of a transformation), the natural connection with determinants, along with many other basic concepts were realized by Cayley in his original works.[12] The fact that matrices provided a simple and concrete example of a noncommutative algebra led to a greater interest in algebra and has been credited by Bourbaki for being one of the chief factors in the development of an abstract view of algebra.

Matrix algebra provided a convenient notation for the solution of a nonsingular system of equations. The system can be represented in the notation $Ax = b$ and the solution by $x = A^{-1}b$. In Cayley's 1857 paper, Cramer's rule was translated into the familiar determinantal formula

$$A^{-1} = \text{adj}(A)/\det(A).$$

Once again, the newer developments were a step in the wrong direction as far as determining solutions for linear systems are concerned. In some older texts authors became carried away with the advantages of matrix notation and it was not uncommon to find the suggestion that in order to solve the linear system $Ax = b$, one should compute A^{-1} and then form the product $A^{-1}b$.[13] This was not always explicitly advocated but it sometimes was implicit. In the text *Computational Methods For Linear Algebra* by D. K. Faddeev and V. N. Faddeeva it is reported that in structural mechanics

there is a method developed to solve a linear system which is based on the computation of so-called "influence numbers". This method is nothing more than a means for constructing the inverse of the coefficient matrix and the "influence numbers" are nothing more than the entries of the inverse matrix. This, of course, is absurd as far as practical computation is concerned. For an (n x n) matrix A, it takes about n^3 operations just to compute A^{-1} using efficient elimination methods and then another n^2 operations to form the product $A^{-1}b$ whereas Gaussian elimination with back substitution requires only $n^3/3$ operations. Thus Gaussian elimination with back substitution is at least *three times more efficient* than computing the inverse of A.

Although the matrix notation offered no advantage in solving a linear system when used in the manner described above, it was eventually put to good use in providing a concise formulation for Gaussian elimination in terms of matrix products. Consider the system Ax = b and triangularize A by Gaussian elimination. Each elementary operation has an associated "elementary matrix" and these elementary matrices are each nonsingular and triangular. The process of reducing A to an upper triangular form can be accomplished by a sequence of matrix multiplications using the sequence of elementary matrices associated with each of the steps of Gaussian elimination. If there are no zero pivots so that no row interchanges are necessary, then each elementary matrix in this sequence will be lower triangular. Consider a 3 x 3 example.

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \xrightarrow[R_3 - (c_1/a_1)R_1]{R_2 - (b_1/a_1)R_1} \begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & u_1 & u_2 \\ 0 & v_1 & v_2 \end{bmatrix} \xrightarrow{R_3 - (v_1/u_1)R_2} \begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & u_1 & u_2 \\ 0 & 0 & w_3 \end{bmatrix}$$

Denote the multipliers of this elimination process by

$$(b_1/a_1) = m_{21},\ (c_1/a_1) = m_{31} \text{ and } (v_1/u_1) = m_{32}.$$

Using elementary matrices, this can be concisely written as

$$E_{32}E_{31}E_{21}A = U$$

where U is the upper triangular matrix with which back substitution will be performed. The diagonal elements of U are the pivots. The product of these three elementary matrices is

$$P = E_{32}E_{32}E_{21} = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ \hline -m_{21} & 1 & 0 \\ \hline m_{32}m_{21} - m_{31} & -m_{32} & 1 \end{array}\right]$$

Let $L = P^{-1}$ and notice that

$$L = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix}$$

Matrix L has 1's on the diagonal and contains, *in exactly the right positions*, the multipliers of the elimination process. That is, it isn't really necessary to do any matrix multiplications or inversions in order to obtain L. It is directly obtained, entry by entry, as the elimination process evolves.

What one now has is a factorization of A (provided A has no zero pivots)[14] into A = LU where L is lower triangular with 1's on the diagonal and below the diagonal are the multipliers (m_{ij}) of row j which is subtracted from row i while Gaussian elimination is being applied to A. Matrix U is upper triangular, containing the pivots on its diagonal, and is the end result of Gaussian elimination applied to A. The LU factorization as just described is unique.

One can go one step further and factor U as $U = D\tilde{U}$ where D is the diagonal matrix whose diagonal entries are those of U (i.e., the pivots) and $\tilde{U}$ is upper triangular with 1's on its diagonal. This is commonly referred to as the LDU factorization. (The tilde notation is dropped.) The LU or LDU factorization provides a compact matrix formulation for the solution of Ax = b using Gaussian elimination with back substitution.

> For Ax = b, first factor A as A = LU
> Now solve Ly = b for y and then solve Ux = y for x.

This process is the exact equivalent (in terms of arithmetic operations) of applying Gaussian elimination with back substitution to the augmented matrix [A |b].

One of the advantages which the LU (or LDU) decomposition affords is compactness. By start-

ing with the augmented matrix

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ & & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

it is easy to systematically perform the LDU decomposition in such a way so as to sequentially overwrite the initial array until it has been transformed into

$$\left[\begin{array}{cccc|c} u_{11} & u_{12} & \cdots & u_{1n} & y_1 \\ m_{21} & u_{22} & \cdots & u_{2n} & y_2 \\ & & \cdots & & \\ m_{n1} & m_{n2} & \cdots & u_{nn} & y_n \end{array}\right]$$

thereby circumventing the necessity to write down or to keep track of each of the successive arrays associated with the implementation of Gaussian elimination.

Another big advantage afforded by the LU approach to Gaussian elimination is that the reduction of the coefficient matrix A and the solution of any system Ax = b can be completely separated This means that once the LU factors for A have been obtained, they can be used to solve Ax = b *for any number of different* b's.

Although it is not clear who first formulated the LU or LDU factorization, it was probably known from the early stages.[15] Charles Hermite (1822-1901) from France introduced in 1851 what is now known as the *Hermite Normal Form* which embodied all of the basic aspects of an LU type of factorization, including the utilization of elementary matrices, but in a much more abstract setting. P. S. Dwyer in his text *Linear Computations* credits the Polish mathematician T. Banachiewicz (during the 1930's) as being perhaps the first to realize that "the basic problem (in extracting a numerical solution using Gaussian elimination) is really one of matrix factorization."

In the era of the slide rule and then of the desk machine (mechanical) calculator, the advantages of the compact LU factorization methods were realized and several efficient variants along with detailed tabular schemes were devised. Three important methods emerged and each can be described in terms of an LDU factorization (although they were not all originally formulated in this way).

The first important decomposition is the basic LU factorization described in some detail earlier in this article. It is the result of Gaussian elimination applied to matrix A where the diagonal matrix is subsumed in the upper triangular factor as A = L(DU). This method was first systematically organized (but not as matrix factorization) for practical use in geodetic and least squares calculations[16] by the American M. H. Doolittle in 1878 and is sometimes called the "Gauss-Doolittle method" or simply the "Doolittle method". Doolittle designed his method to apply only to symmetric systems. It was later extended to nonsymmetric systems and became known as the "extended Doolittle method".

The second and extremely important variant that is very common in practice today concerns the problem in which the matrix A is positive definite.[17] In this case, $U = L^T$ and the diagonal entries of D are positive so that part of D can be subsumed by L while part can be subsumed by L^T to yield.

$$A = (LD^{1/2})(D^{1/2}L^T).$$

Although the basic idea may have been implicit in some work of Gauss, it is now known as the *square root method* or the *Cholesky decomposition* named for Andre-Louis Cholesky (1875-1918) from France. Cholesky was a young major in the French military attached to the Geodesic Section of the Geographic Service. Around 1905 Cholesky became involved with the problem of adjusting the triangularization grid of France. The numerical procedures for dealing with the resulting least squares computations were not yet entirely agreed on. For the solution of the normal equations, he invented the scheme which now bears his name. However, Cholesky apparently never published his methods so that his techniques were carried forward by word of mouth until an obscure note by another French officer appeared in 1924. Shortly after 1913, France became involved in war and the promising career of Major Cholesky ended abruptly in battle on August 31, 1918.

The Cholesky decomposition has also been traced to Issai Schur in 1917 and was published by T. Banachiewicz in Poland in 1937. Paul S. Dwyer developed and promoted the Cholesky method (he called it the "square root method") in America in the mid-1940's. It is somewhat ironic that the Cholesky decomposition has found so much popularity in recent times. Dwyer has speculated that due to the natural structure of the method, it is a logical algorithm to derive and that it pro-

bably was discovered, and discarded, several times prior to Cholesky or Schur. The necessity to extract square roots would have rendered it almost useless for practical work in the days preceeding calculating machines and other computing aids. It no doubt would have been labeled by utilitarians of earlier times as another futile attempt by a "theorist" who lacked an appreciation of "practical numerical techniques". This is a vivid illustration, among countless other examples, of how the pure and seemingly impractical mathematics of one era becomes the practical and highly applicable mathematics of a future era.

The third important variation is a factorization in which the diagonal matrix **D** becomes subsumed in the lower factor as **A = (LD)U**. This algorithm was also published by T. Banachiewicz of Poland (mentioned earlier) in 1938. It was independently developed and published in America by P. D. Crout in 1941 and is now known as the "Crout method". Doolittle designed his original method to apply only to symmetric systems whereas Crout's method could be used for nonsymmetric problems. Crout was associated with the Marchant Calculating Machine Company and his methods were designed specifically for use on the old "Marchant mechanical monsters" and some other similar antiques. Court's versions of Gaussian elimination is an efficient organization of data and ordering of computations which results in reduced storage requirements.[18]

Even with the compact and efficient elimination schemes just discussed, the problem of solving say a (30 x 30) system using the manual mechanical calculators of the 1930's and 40's was still an enormous task which existed only at the outer limits of practicality. During the late 1930's a physicist named John V. Atanasoff at Iowa State College became interested in electronic computing machines and he recognized the high speeds which might be attained by electronic methods. He also realized that the correct mode of computation was by digital means as opposed to analog methods. Atanasoff and a colleague, Clifford Berry, designed and built an electronic machine specifically to execute ordinary Gaussian elimination to solve a linear system. The coefficients were stored in capacitors located on a rotating cylinder which made one revolution per second. Intermediate results of the elimination process were punched and stored on cards. The machine was designed to solve systems up to (30 x 30). Herman Goldstine in *The Computer From Pascal To von Neumann* says that the Atanasoff and Berry linear equation solver was probably the first use of vacuum tubes in conjunction with digital computation. A prototype was apparently working early in 1940. Because it was premature[19] in its engineering conception, the machine never became a serious computational tool. However, Goldstine credits the linear equation solver and the ideas embodied therein as having influenced the entire subsequent development of electronic computers. One could say that Gaussian elimination was present in the delivery room and assisted in giving birth to the electronic computer.

The onset of World War II accelerated the interest and development of electronic digital computers. In the early days, the computations of artillery-trajectory tables was a laborious process accomplished by many women using manual mechanical calculators. In 1938, the United States established the Ballistics Research Laboratory and John von Neumann (1903-1957) was brought together with Herman Goldstine (1913-) and a few others. Together with their colleagues, von Neumann and Goldstine developed the first digital electronic computer in which both the program and the data resided in the computer's memory.

In 1949 the first new stored-program digital computer went into operation and von Neumann and Goldstine (along with other mathematicians) directed their attention toward understanding the cumulative effect of rounding in computations carried out on their new machines. Attention focused on solving a square system of linear equations using Gaussian elimination. Herman Goldstine later said

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

Goldstine was referring to the paper by J. von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order,* Bull. Amer. Math. Soc., 53(1947), pp. 1021-1099) which became a cornerstone on which modern numerical analysis was built. Because of its practical significance, von Neumann and Goldstine focused on the problem of factoring a positive definite matrix A into $A = LDL^T$ (i.e., the Cholesky decomposition discussed earlier) by using elimination methods.

Earlier, in 1943, H. Hotelling examined the same problem and showed that if A is (n x n) and if L and D are computed and used to produce a computed inverse matrix, X, then $||AX - I||_2$ was bounded above by a factor which contained the term 4^n. This result helped to reinforce the prevelant pessimistic attitude concerning the use of Gaussian elimination. It was believed that in the elimination process, an error at any stage would affect all succeeding results and become greatly magnified. Thus Gaussian elimination should be expected to be an unstable algorithm. It was felt that

certain iterative methods should be preferred. In fact, von Neumann himself seemed to be of this opinion before he considered the problem in depth.

However, the 1947 paper of von Neumann and Goldstine immediately generated some confidence in Gaussian elimination (and the Cholesky decomposition). They proved that if A is not too near to a singular matrix, then Gaussian elimination and the Cholesky factors will produce a computed inverse, X which satisfies

$$||AX - I||_2 \leq 14.24(M/m)n^2b^{-t}$$

when M and m are the largest and smallest eigenvalues of A, respectively, and the computer works with t digits in the base b. This result was a tremendous improvement since the bound contains only a quadratic factor of n as opposed to the exponential factor of Hotelling.

Although the work of von Neumann and Goldstine was a milestone, there were some shortcomings. The hardware for floating-point arithmetic was not available in the first generation of computers so that fixed-point[20] arithmetic had to be used and thus the error analysis was based on fixed-point arithmetic. In truth, floating-point analysis is simpler and provides more satisfactory results. However, it was many years later before this became fully realized. Furthermore, the emphasis was on what is now referred to as *forward analysis* in which bounds for the difference between a computed solution and the exact solution are determined. In the early error analysis of elimination methods, the effects of rounding were not completely separated from the effects of ill-conditioning nor was the use of different pivoting strategies fully understood. More will be said about some of these ideas later in this article.

While American mathematicians were grappling with these problems, a parallel development was occurring at about the same time in England. Alan M. Turing, a noted English logician and mathematician, formulated the now famous concept of an abstract computer (the Turing machine) and was instrumental in developing the English computer called the ACE.[21] Like his American counterparts, Turing became interested in producing numerical solutions for linear systems using Gaussian elimination.

One of Turing's contributions was to help clarify the notion of ill-conditioning. Although this concept (along with the term "ill-condition") had been in common use, it was often used rather nebulously. In 1948, Turing introduced the term *condition number for a nonsingular matrix.* This is defined to be the number

$$c(A) = ||A||_2 ||A^{-1}||_2$$

and is used to gauge the inherent sensitivity of the problem itself to small perturbations.[22] For example, consider the system

$$.835x + .667y = .168$$

$$.333x + .266y = .067$$

in which the exact solution is

$$x = 1 \text{ and } y = -1.$$

The condition number of the coefficient matrix is about 1.3×10^6. This relatively large condition number signals that there exist small perturbations to which the system will be quite sensitive. Indeed, if $b_2 = .067$ is only slightly perturbed to $\tilde{b}_2 = .066$, then the exact solution changes drama-

$$\tilde{x} = -666 \text{ and } \tilde{y} = 834.$$

This is an example of an ill-conditioned system. The sensitivity exhibited is intrinsic to the system itself and is not due to any sort of unstable computational scheme. If the exact solution is sensitive to small changes in the coefficients, then any computed solution cannot be less so. Thus there does not exist an algorithm or any kind of "numerical trick" which will produce a solution that is not sensitive to roundoff (or any other source of error). If one attempts to solve this system using Gaussian elimination with 5-digit arithmetic (i.e., round each calculation to 5 significant digits), then the algorithm completely breaks down and fails to produce a solution due to the emergence of a zero pivot. However, this is *not* an indictment of Gaussian elimination. It seems that this point was not universally appreciated in the early days of implementing Gaussian elimination on electronic computers

Another such point is the effect of ill-conditioning on the residual. That is, suppose that x_c is a computed solution for a system $Ax = b$ and it is demonstrated that the residual $r = Ax_c - b$ is small in some sense (i.e., Ax_c is very close to b). Does this guarantee that x_c is close to the exact solution? Surprisingly, the answer is a resounding "NO!" when A is ill-conditioned. For example, in the ill-conditioned system given above, suppose that somehow you compute a solution and obtain

$$\tilde{x} = -666 \text{ and } \tilde{y} = 834.$$

If you attempt to "check the error" in your computed solution by substituting back into the original system and you form the residuals, then you find (even using exact arithmetic) that

$$.835x + .667\tilde{y} - .168 = 0$$

$$.333\tilde{y} + .266\tilde{y} - .067 = -.001.$$

That is, the computed solution $(\tilde{x},\tilde{y}) = (-666, 834)$ *satisfies exactly* the first equation and comes *very close* to satisfying the second. On the surface, this might seem to suggest that your computed solution should be very close to the exact solution. (In fact, you may be seduced into believing that you are within .001 of the true solution.) Obviously, this is nowhere close to being true since the exact solution is

$$x = 1 \text{ and } y = 1.^{23}$$

During the 1950's, it seems that some of these points were not always put in proper focus and this may have accounted for some of the undue pessimism which once surrounded the use of Gaussian elimination.

In theory, the order in which the equations of a linear system are listed is irrelevant. In practice, the order can "make all the difference in the world". An extremely accurate or a very inaccurate result can be produced by Gaussian elimination depending on how the equations are ordered.[24] For example, consider solving the following system using 3-digit arithmetic.

$$-10^{-4}x + y = 1$$

$$x + y = 2$$

Applying Gaussian elimination to the equations in this order yields

$$\left[\begin{array}{cc|c} -10^4 & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \xrightarrow[(R_2+10^4R_1)]{} \left[\begin{array}{cc|c} -10^4 & 1 & 1 \\ 0 & 10^4 & 10^4 \end{array}\right]$$

because $1 + 10^4 = 10{,}001$ and $2 + 10^4 = 10{,}002$ both get rounded to 10^4. Back substitution now produces

$$x = 0 \text{ and } y = 1.$$

Now interchange the two equations and apply Gaussian elimination to obtain

$$\left[\begin{array}{cc|c} -10^4 & 1 & 1 \\ 1 & 1 & 2 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ -10^{-4} & 1 & 1 \end{array}\right] \xrightarrow[(R_2+10^{-4}R_1)]{} \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right]$$

because $1 + 10^{-4} = 1.0001$ and $1 + 2x10^{-4} = 1.0002$ both get rounded to 1. This time, back substitution produces

$$x = 1 \text{ and } y = 1,$$

The exact solution of the system is

$$x = 1/1.0001 \text{ and } y = 1.0002/1.0001$$

which when rounded to 3 significant digits yields

$$x = 1 \text{ and } y = 1.$$

Thus leaving the equations in original order produces a very *inaccurate* computed solution whereas the result of an interchange gives a very *accurate* computed solution. This is due in part to the size of the pivotal element used in the elimination process.

This discovery led to a modification which today is so standard that it is almost always assumed to be part of the meaning of the phrase "Gaussian elimination" as it is currently used. This modification is simply to search each coefficient on and below the pivotal position for the number of maximal magnitude and then (if necessary) perform the appropriate row interchange to bring this coefficient of maximal magnitude into the pivotal position.

This strategy of "pivoting for size" was evidently known to von Neumann and Goldstine in 1947 (as well as to Turing in 1948) but their use of it in factoring a positive definite matrix was needed only because their analysis was for fixed-point arithmetic. For floating-point computations, the use of pivoting is not necessary when dealing with a positive definite matrix. This was not realized until some years later and led to confusion with regard to the use of pivoting strategies in Gaussian elimination.

In 1961, James H. Wilkinson[25] from England introduced the now familiar term *partial pivoting* for the pivoting strategy described above and his work demonstrated that its use in the elimination process is of fundamental importance for problems of a general nature and that floating-point

arithmetic doesn't remove the need for it. In the same paper, Wilkinson also introduced the now well known term *complete pivoting*. In this pivoting strategy, one searches at each step the entries below and/or to the right of the pivotal position for the number of maximum magnitude. If necessary, the appropriate row *as well as column* interchange is performed in order to bring the coefficient of maximum magnitude into the pivotal position.

Before 1961, there was some uncertainty concerning the use of partial pivoting versus complete pivoting in Gaussian elimination. Complete pivoting can be proven mathematically to be "safe" in the sense that errors are never unreasonably magnified in comparison with the size of the matrix. Experience seemed to indicate the same to hold for partial pivoting as well. However, no mathematical proof could be found. It was important to resolve this point because the cost of partial pivoting is very small while the cost of complete pivoting is relatively large. In his 1961 paper, Wilkinson presented a pathological example of a well-conditioned matrix to show that partial pivoting is *not* always "safe" and that it can fail where complete pivoting succeeds. In spite of this fact, Gaussian elimination *with partial pivoting* has become the preferred algorithm of today for solving a general linear system which will fit entirely into the computer's memory. John R. Rice in his text *Matrix Computations and Mathematical Software* sums up the feeling of today's practitioner with the following statement:

> "Why should I pay twice the cost just to protect myself from a situation so rare that it took numerical analysts years to find an example of it?"

The work of Wilkinson during the 1960's had a profound impact not only concerning Gaussian elimination but on all analysis of matrix computations and much of numerical analysis in general. Wilkinson provided a complete new level of understanding of Gaussian elimination and put its usage on a firm foundation. Perhaps one of Wilkinson's most important contributions was his development and promotion of the idea of *backward error analysis* as opposed to forward error analysis which was in vogue during the 1940's and 50's.

Recall that the object of a forward analysis is to produce error bounds for the differences between the exact solution and a computed solution. That is, one tries to answer the question "How close can we expect a computed solution to be to the exact solution?" With backward analysis, one tries to answer the question "What system did I really solve (i.e., obtain the exact solution of) and how close is this sytem to the one I originally tried to solve?" If $Ax = b$ is the original system and x_c is a computed solution, then x_c will be the exact solution for another system $(A + E)x = b$. The object of a backward analysis is to produce bounds on the size of the entries of E. To complete the analysis, one then must perform a perturbation analysis in order to determine the effects on the solution of $Ax = b$ when A undergoes a perturbation from A to $(A + E)$. Backward analysis has the advantage of being able to separate the effects of rounding in using a particular algorithm from the effects of ill-conditioning which are inherent in the problem itself. Furthermore, because rounding errors are reflected back into the original matrix, they are placed on the same footing as errors that may be present in the original data, which usually must also be taken into account. To illustrate the idea of backward analysis in a simple example, consider the system.

$$-10^{-4}x + y = 1$$
$$x + y = 2$$

discussed earlier. With 3-digit arithmetic and no pivoting, Gaussian elimination produces the computed solution $\tilde{x} = 0$ and $\tilde{y} = 1$ which is the *exact solution* of another system

$$-10^{-4}u + v = 1$$
$$u \quad = 0$$

which is *quite different from the original system* and hence shouldn't be expected to produce an exact solution which is close to the exact solution of the original system. This *is* an indictment against the algorithm used. When partial pivoting is incorporated, the computed solution becomes $\tilde{x} = 1$ and $\tilde{y} = 1$, which is the exact solution of

$$v = 1$$
$$u + v = 2$$

which is *very close to the original system.* For this example, backward analysis shows that Gaussian elimination with partial pivoting acted in a very "stable" way whereas without pivoting, the algorithm was "unstable".

Although Wilkinson, in his work of the early 1960's, was the one responsible for developing backward error analysis and providing a more complete and solid understanding of the numerical analysis of Gaussian elimination, he has said that the notion of backward analysis was implicit in some of the 1947 work of von Neumann and Goldstine but that it was concealed in their tedious presentation. He also attributed the work of Turing in 1948 with containing some implicit backward analysis. However, Wilkinson credits Wallace Givens as "the true father of backward error analysis". Givens presented in 1954 some work on analyzing the computations of eigenvalues of tridiagonal matrices in which he emphasised the use of backward analysis.

This takes us up to problems still under investigation. A seemingly simple aspect of Gaussian elimination which is still not completely understood concerns the problem of *scaling*. From the early days of automatic computation it was known that even a good algorithm (such as Gaussian elimination with partial pivoting) can fail if there are numbers of different magnitude involved. For example, using 3-digit arithmetic and partial pivoting on the system

$$-10x + 10^5 y = 10^5$$

$$x + \quad y = 2$$

produces a computed solution of $x = 0$ and $y = 1$, which is not very accurate since the exact solution is $x = 1/1.0001$ and $y = 1.0002/1.0001$. The trouble stems from the fact that the first equation contains coefficients which are extremely larger than those of the second equation. If we rescale the system to insure that the coefficient of maximum magnitude in each row is 1 (i.e., multiply the first equation by 10^{-5}), then the result is the system given earlier in this article. For the rescaled system, we have already seen that Gaussian elimination with partial pivoting produces a very accurate computed solution.

In his 1961 paper, Wilkinson suggests that in order for pivoting strategies to be most effective, all rows and columns of the coefficient matrix should have comparable norms. The ideal case is when each row and column of the coefficient matrix has the same norm.[26] For a system $Ax = b$, scaling (or equilibrating) A is usually attempted by replacing A by D_1AD_2 where the D's are nonsingular diagonal matrices. (Pre-multiplication by a nonsingular diagonal matrix simply multiplies each equation by a nonzero constant and, theoretically, doesn't alter the solution. Post-multiplication by a diagonal matrix has the effect of multiplying the columns of A by constants, which is equivalent to changing the units of each of the unknowns.) Unfortunately, it is not always possible to exactly equilibrate A by using "diagonal scalings". Furthermore, there is at present no known strategy for determining "optimal" scaling matrices. There is still some research being conducted on the problem of diagonal scalings.

As computer architecture became more sophisticated, it became possible to attempt to solve larger and larger problems. During the 1970's a great deal of emphasis was placed on solving very large systems in which the coefficient matrix usually had some special structure. In addition to being large, very frequently the problems of interest had a very sparse distribution of nonzeros. Applying Gaussian elimination to these large sparse systems opened up new areas of research because when one applies an elimination technique to this kind of matrix, one wants to preserve the sparsity. That is, you don't want to create a matrix which is mostly nonzero from one which originally was mostly full of zeros, especially if the size is quite large. Many new techniques (such as graph theory) have recently been brought to bear on this type of problem and there is currently still a great deal of research activity surrounding the very old and very simple idea of Gaussian elimination.

As long as technology keeps evolving so as to produce even better types of computing machines and as long as mathematicians are active in searching out new frontiers, Gaussian elimination will also continue to evolve and always be a source of interest for those trying to "just solve a linear system of equations." ⊡

NOTES

[1]It was probably written under the philosophy that "Just knowing the theory is sufficient for classroom purposes." After all, why clutter virgin minds with the nasty facts of practical life?

[2]"Being in the red" obviously had a different connotation than it has today.

[3]It is curious that the name of Gauss is associated with this concept since elimination methods were apparently used hundreds of years before Gauss. More will be later said about this.

[4]Two systems are equivalent if they possess the same solution set.

[5]The word "determinant" seems to have originated with Gauss but in a somewhat different context. It appears that it was Augustin-Louis Cauchy (1789-1857) who later popularized this terminology in Europe for what otherwise was described as "alternating symmetric functions".

[6]See **The Theory of Determinants In The Historical Order of Development**, Vols, I, II, III, and IV, by Sir Thomas Muir.

[7]Unfortunately, there still exist a few isolated elementary texts and some instructors that teach Cramer's rule as a tool to numerically solve a linear system. This is, of course, almost impossible for all but the simplest textbook problem. Using Cramer's rule and the n! terms in the expansion of an (n x n) determinant on a computer capable of a million multiplications and additions each second, running day and night for a whole century, one would not have enough time to solve even a (15 x 15) system. (This even neglects the effect of errors attributed to roundoff.)

[8]**Theoria Motus corporum coelestium in sectionibus conicis solem ambientium**, 1809.

[9]Of course, Gauss didn't phrase his discussions or his results in the modern matrix notation which we have used here because matrix multiplication had not yet been defined. Gauss prescribed individual formulas for each term of his process and didn't place the terms in any sort of array.

[10]An operation is a multiplication or a division. Because additions and subtractions are so much faster to perform in automatic computation, they are usually not counted when trying to gauge the efficiency of an algorithm.

[11]The word **matrix** was first used by Sylvester in 1850 but not with today's meaning. He invented the word to refer to a rectangular array because he couldn't use the word **determinant**. He was concerned only with determinants which could be formed from the numbers of a rectangular array. Sir Thomas Muir said the use of the word matrix was "entirely uncalled for . . . to take the place already satisfactorily occupied by the word array".

[12]Cayley was not the first to compose linear transformations. In fact, an example of such a composition appears in the **Disquisitiones Arithmeticae** of Gauss in 1801. However, Gauss failed to make the connection between the composition of linear transformations and composition of the associated matrices.

[13]Unfortunately, there still exist a few isolated cases of textbooks and instructors who teach this to their students.

[14]If each leading principal minor of A is nonzero, this is guaranteed.

[15]G. W. Stewart in his text **Introduction to Matrix Computations** credits Alan Turing from England for introducing the terminology "LDU decomposition" in 1948. However, it seems clear that the ideas surround triangular factorizations existed and were used long before 1948.

[16]A truly incredible amount of mathematics, both pure and applied, has been spawned by the basic necessity to perform geodetic and least squares calculations.

[17]Extensions to the case where A is symmetric but not positive definite are also considered in this context.

[18]Crout's method allows inner products to be accumulated in double precision. With the basic Gaussian elimination as described earlier, this would double the storage requirements.

[19]Goldstine uses Atanasoff to make the point that there seems to be an optimal time for discovery as well as an optimal period for perfection. If an idea is put forth too early, it frequently fails to survive due to inadequacies in existing knowledge and/or technology and the inability to make connections with relevancies of the moment. We have noted this phenomenon previously in the discussion of the discovery of Cramer's rule and the Cholesky decomposition of a positive definite matrix. If one proposes an idea too early, it can become lost because it can't be properly perceived whereas if one proposed exactly the same idea, but waits too long, it can be considered "obvious" or "trivial".

[20]For **fixed point** computation, the computer operates only on numbers for which the radix point is in a "fixed" location. The most common representation was in the form

$$\pm .d_1 d_2 \cdots d_t$$

with the radix point always "fixed" at the far left hand side of a sequence of base b (usually b = 2) integers d_i. It was the programmer's job to properly scale the numbers in the input and output phases (similar to how one used a slide rule). Most modern computers perform computations using **floating-point numbers**. A t-digit base-b floating point number has the form

$$(\pm .d_1 d_2 \cdots d_t) \times (b^e)$$

where the base b, the exponent e, and the digits d_i are integers. The base b is usually chosen from the set [2,4,8,10, 16] and the digits satisfy $0 \leq d_i \leq b-1$ with $d_1 \neq 0$. The range of e ($-m \leq e \leq M$) and the value of t vary greatly among different machines. Thus a floating-point number is an adaptation of the familiar scientific notation.

[21]ACE is the acronym for **Automatic Computing Engine** which was named in recognition of Charles Babbage's "Analytical Engine" of 1834 which was never completed.

[22]The term (M/m) in the von Neumann-Goldstine error bound given earlier is the condition number for a symmetric matrix, but von Neumann and Goldstine did not explicitly single out this number by name nor did they emphasize its relationship to the inherent sensitivity of the problem.

[23]This is always a shock to a student seeing this illustrated for the first time because it indeed is a counter to a novice's intuition. Unfortunately, many elementary textbooks and instructors still teach their students that they can always "check" the accuracy of their solution by substituting it back into the original system and see how close it comes to solving the system. With electronic hand calculators and personal computers finding their way into the high school (and even lower level) classrooms of today, it becomes important that these "fine points" no longer be regarded as "details not worth spending time with" when teaching students how to solve a linear system.

[24]This is a glaring example of how a very simple and solid truth, in theory, can degenerate into a complete falsehood when implemented (and thus slightly compromised) in practice. It is again unfortunate that this very important distinction between the theory and practice of basic Gaussian elimination has not reached most of the elementary textbooks (as well as many instructors).

[25]"Error analysis of direct methods of matrix inversion", J. Assoc., Comp., Mach., 8(1961) pp. 281-330.

[26]Such matrices are termed "equilibrated", a word Wilkinson says was invented by F. L. Bauer.