Contemporary Mathematics
Volume **754**, 2020
https://doi.org/10.1090/conm/754/15141

Iterative methods for linear systems of equations: A brief historical journey

Yousef Saad

Abstract. This paper presents a brief historical survey of iterative methods for solving linear systems of equations. The journey begins with Gauss who developed the first known method that can be termed iterative. The early 20th century saw good progress of these methods which were initially used to solve least-squares systems, and then linear systems arising from the discretization of partial differential equations. Then iterative methods received a big impetus in the 1950s - partly because of the development of computers. The survey does not attempt to be exhaustive. Rather, the aim is to bring out the way of thinking at specific periods of time and to highlight the major ideas that steered the field.

1. It all started with Gauss

A big part of the contributions of Carl Friedrich Gauss can be found in the voluminous exchanges he had with contemporary scientists. This correspondence has been preserved in a number of books, e.g., in the twelve "Werke" volumes gathered from 1863 to 1929 at the University of Göttingen[1]. There are also books specialized on specific correspondences. One of these is dedicated to the exchanges he had with Christian Ludwig Gerling [**67**]. Gerling was a student of Gauss under whose supervision he earned a PhD from the university of Göttingen in 1812. Gerling later became professor of mathematics, physics, and astronomy at the University of Marburg where he spent the rest of his life from 1817 and maintained a relation with Gauss until Gauss's death in 1855. We learn from [**73**] that there were 388 letters exchanged between the two from 1810 to 1854 (163 written by Gauss and 225 by Gerling).

It is in one of these letters that Gauss discusses his method of *indirect elimination* which he contrasted with the method of *direct elimination* or Gaussian elimination in today's terminology. Gauss wrote this letter to Gerling on Dec. 26th, 1823. Gerling was a specialist of geodesy and the exchange in this letter was about the application of the method of least-squares, which Gauss invented in the early 1800s, to geodesy. An English translation of this letter was published in 1951 by George Forsythe [**25**]. The historical context of this translation is interesting in itself. In the forward, Forsythe begins by stating that his specific aim was to

2010 *Mathematics Subject Classification.* Primary 65F10,65-03, 01A55.
This work was supported by NSF grants DMS 1521573 and DMS 1912048.
[1]https://gdz.sub.uni-goettingen.de/volumes/id/PPN235957348

find whether or not the reference in a book by Zurmühl on matrices [**88**] that mentioned the method of relaxation described by Southwell [**75**, **76**] is the same as the one given in "Dedekind's report on Gauss's Lectures" (see references in [**25**]) *"...It is believed by some computers*[2] *however, that Gauss's method is a different one, namely the related method given by Seidel* [**69**]. *In the interest of giving Gauss his proper credit as a proponent of relaxation, the following translation of a letter by Gauss is offered."*

Let us take a look at the content of the letter. After some manipulations of data, Gauss arrives at a least-squares system for determining angles formed by certain directions for 4 German cities. For this he states that he will describe an "indirect" method for solving the normal equations. The equations of the resulting least-squares system are then combined so that the entries of each column sum-up to zero. This results in a linear system which is written in the following form:

$$0 = +6 \quad + 67a - 13b - 28c \; - 26d \tag{1.1}$$

$$0 = -7558 \; - 13a + 69b - 50c \; - 6d \tag{1.2}$$

$$0 = -14604 - 28a - 50b + 156c - 78d \tag{1.3}$$

$$0 = +22156 - 26a - 6b \; - 78c \; + 110d \tag{1.4}$$

As can be verified all column sums are equal to zero. The preliminary step to produce a system of this type is a 'trick' which helps get a better convergence and provides what we may call today *an invariant*, i.e., a property that is always satisfied and can therefore be useful in detecting calculation errors. Gauss mentions that without this trick *"you lose the great convenience of always having as a control the sum of the absolute terms = 0"*. The absolute terms are the terms of the first column of the above system, which is the negative of what we now call the right-hand side.

In the iterative scheme proposed, the coordinates of the solution change and the right-hand side, which is in fact the residual vector, is updated each time. The mechanical nature of the procedure is at the same time simple and appealing. So for example, all coordinates of the solution are set to zero, and in the first step, he selects to modify the 4th coordinate because it would lead to the biggest decrease in the residual. The letter then shows the progress of the algorithm in the following table:

	$d = -201$	$b = +92$	$a = -60$	$c = +12$	$a = +5$	$b = -2$	$a = -1$
$+6$	$+5232$	$+4036$	$+16$	-320	$+15$	$+41$	-26
-7558	-6352	-4	$+776$	$+176$	$+111$	-27	-14
-14604	$+1074$	-3526	-1846	$+26$	-114	-14	$+14$
$+22156$	$+46$	-506	$+1054$	$+118$	-12	0	$+26$

The first column is just the initial residual (corresponding to the initial guess $x = 0$ in today's terminology). The largest entry is the 4th, and so Gauss selects to update d which now has the value $d = -201$ (obtained by making the last equation satisfied). Note that all values are rounded and this is one of the important attractions of this method. The second column shows the next modification to be added this time to b. In today's notation we would write something like $\Delta b = +92$. At the end of the 7 steps above, we have $a = -56, b = +90, c = 12, d = -201$. The

[2]In the 1950s a 'computer' often referred to someone who specialized in numerical computing, i.e., whose job was to carry out a calculation given a certain algorithm, or set of formulas.

corresponding residual is shown in the last column. Note here that in Step 6, Gauss changes b instead of c while in fact the the 3rd residual component (-114) has a bigger magnitude than the second (-111). We can only speculate that this is done for convenience since the 3rd equation has more big terms than the first and 111 and 114 are not too far apart. It turns out for this example, nothing changes after the 7 steps shown above: *"Insofar as I carry the calculation only to the nearest 2000-th of a second, I see that now there is nothing more to correct..."* and the final solution is displayed. Gauss ends the letter with the words: *"... Almost every evening I make a new edition of the tableau, wherever there is easy improvement. Against the monotony of the surveying business, this is always a pleasant entertainment; one can also see immediately whether anything doubtful has crept in, what still remains to be desired, etc. I recommend this method to you for imitation. You will hardly ever again eliminate directly, at least not when you have more than 2 unknowns. The indirect procedure can be done while half asleep, or while thinking about other things."*

Gauss recommends this iterative scheme *(indirect elimination)* over Gaussian elimination for systems of order > 2. We will contrast this with other recommendations later.

This appears to be the first known reference to a use of an iterative method for solving linear systems. Later, in 1845 Jacobi [**40**] developed a relaxation type method in which the latest modification is not immediately incorporated into the system. In that same paper he introduces a way of modifying a linear system by using what we now know as "Jacobi rotations" to annihilate large diagonal entries before performing relaxations. This is in order to speed-up convergence of the iteration, and so one can say that Jacobi introduced the first known form of preconditioning [3] That same technique that uses cleverly selected rotations, was advocated in a paper that appeared a year later [**41**] for solving symmetric eigenvalue problems.

Though this letter to Gerling dealt with a 4×4 linear system, Gauss solved bigger systems with the same method. Bodewig [**11**, p. 145] describes the state of the art with this method in its early stages as follows: *Gauss had systems of 20—30—40 unknowns, later still higher systems were met, for instance, in the triangulation of Saxony by NAGEL in 1890: 159 unknowns, H. BOLTZ at Potsdam 670 unknowns and in the present triangulation of Central Europe several thousands of unknowns.*

2. Solution by relaxation

The term *relaxation* which is in common use today to describe the general process first invented by Gauss, seems to have been introduced by Southwell [**75**, **76**]. Suppose we have a linear system of the form

$$Ax = b. \tag{2.1}$$

which can be viewed as a collection of equations:

$$a_i x = \beta_i, \quad i = 1, \cdots, n \tag{2.2}$$

[3]Note that this would not be called preconditioning in today's terminology as the condition number of the system does not change. It is a preprocessing step in which the matrix is explicitly modified so as to achieve better diagonal dominance. However, the idea of modifying a system to achieve better convergence is of the same essence as that of preconditioning.

where $a_i \in \mathcal{R}^{1\times n}$ is the i-th row of A and β_i the i-th component of b. We will denote by r the residual vector:

$$r = b - Ax. \tag{2.3}$$

The relaxation process is then as follows, where we set $r^{(new)} = b - Ax^{(new)}$:

Modify i-th component of x into $x_i^{(new)} := x_i + \delta$ so that: $r_i^{(new)} = 0$.

This means that we should have

$$a_i(x + \delta e_i) = \beta_i \quad \longrightarrow \quad \delta = \frac{r_i}{a_{ii}}.$$

This is done in a certain order. In the original approach by Gauss, the method is to select i to be the coordinate of the residual that has the largest entry in magnitude. The newly computed component of x is substituted into x and the new associated residual is then computed. In the Jacobi method, all n components are updated using the same r.

In 1874, Seidel [**69**] described a relaxation method that was again geared toward solving normal equations. His method can be viewed as a relaxation for the system $A^TAx = A^Tb$ of normal equations, and because of this specificity he was able to argue for convergence. He also mentions that unknowns need not be processed from 1 to n cyclically. Instead, convergence is improved by making each dominant residual into zero, which is the same scheme as the one initially proposed by Gauss. In the same paper Seidel also developed a block method whereby a few unknowns are processed at the same time. Often in the literature that followed, "relaxation" became synonymous with Seidel's method and the method was often called Seidel's method. It is now called the Gauss-Seidel method in an effort to give credit to Gauss who invented the non-cyclic variant almost 50 years before him. The fact that Seidel recommends against processing the unknown cyclically prompted Gorge Forsythe to remark that *"the Gauss-Seidel method was not known to Gauss and not recommended by Seidel"* according to Householder, see [**39**, p. 115]. In the same notes, Householder also mentions that Nekrasov, a Russian author (see [**39**] for the reference) defined the exact same method as Seidel in 1884 and that "the method is called Nekrasov's method in the Russian literature." In fact, to this day the method is referred to as the method of Seidel-Nekrasov by some Russian authors. Nekrasov analyzed the method theoretically [**53**] and the paper [**47**] shows a convergence result.

One of the main attractions of the cyclic version of Gauss-Seidel iteration is that it can easily programmed or "mechanized" as was said in the early days of computing. David Young recounts the following anecdote in [**86**]: *"Not too long after I began my work, Sir Richard Southwell visited Birkhoff at Harvard. One day when he, Birkhoff and I were together, I told him what I was trying to do. As near as I can recall, his words were* 'any attempt to mechanize relaxation methods would be a waste of time.' *This was somewhat discouraging, but my propensity of making numerical errors was so strong that I knew that I would never be able to solve significant problems except by machines. Thus, though discouraged, I continued to work."*

Relaxation-type methods can be written in the form of fixed point iterations and this makes it easy to analyze their convergence. Consider the decomposition $A = D - E - F$ where D is a diagonal matrix having the same (diagonal) entries as those of A, $-E$ is the strictly lower triangular part of A and $-F$ its strictly upper

triangular part. Then the method of Gauss-Seidel generates the iterates defined by

$$x^{(k+1)} = (D-E)^{-1}\left(Fx^{(k)} + b\right). \tag{2.4}$$

Here one can write $Ax = b$ as $(D-E)x = F+b$, using the *splitting* $A = (D-E) - F$, from which the above iteration follows. Similarly, the Jacobi iteration is of the form $x^{(k+1)} = D^{-1}[(E+F)x^{(k)} + b]$ and is based on the splitting: $A = D - (E+F)$.

In addition to the original paper by Seidel mentioned above, convergence of the standard Gauss-Seidel process was studied early on by several authors, see, e.g., [**83**]. A number of these results can be found in a chapter of Bodewig's book [**11**, Chap. 7] that starts with the warning: *"But, first, let us note that these theorems are more or less superfluous in practical computation. For the iteration methods will only be applied when the convergence is evident at first sight, that is, when the diagonal dominates strongly whereas in other cases convergence will be too slow even for modern computing machines so that it is better to apply a direct procedure (Gauss or variants)."*

3. Early 20th century

The early 20th century was marked by the beginning of the application of iterative methods to problems modeled by partial differential equations. Up to that period, iterative methods were mainly utilized to solve linear systems that originated from normal equations. On the other hand, a method proposed by Liebmann [**45**] was geared specifically toward solving discretized Poisson equations. The method is nothing but what we term today the Gauss-Seidel method, and for this reason the Gauss-Seidel iteration when applied to Partial Differential Equations was often called the Liebmann method. It is known as Nekrasov's method in the Russian literature [**53**].

In a remarkable paper published in 1910, Richardson [**58**] put together a number of techniques for solving simple PDEs (Laplace, Poisson, Bi-Harmonic, ..) by finite differences. He then describes an iterative scheme for solving the linear system that results from discretizing these equations. The PDEs addressed in the paper are all of a homogeneous type, e.g., Laplace, or $\Delta u = 0$, with boundary conditions. This results in a linear system that can be written as $Ax + b = 0$ where A acts on interior points only and b reflects the action of the discretized operator on the boundary points. Thus, for an arbitrary x, the vector $Ax+b$ represents the residual of the system under consideration. With this notational point in mind, the method introduced by Richardson in this paper can be written in the form:

$$x_{j+1} = x_j - \frac{1}{\alpha_j} A r_j \tag{3.1}$$

and results in a polynomial iteration scheme whose residual at step k satisfies

$$r_k = \left(I - \frac{A}{\alpha_k}\right) \cdots \left(I - \frac{A}{\alpha_2}\right) \cdots \left(I - \frac{A}{\alpha_1}\right) r_0. \tag{3.2}$$

Thus, r_k is of the form:

$$r_k = p_k(A) r_0 \tag{3.3}$$

where p_k is a polynomial of degree k satisfying the constraint $p_k(0) = 1$ that depends on the free coefficients $\alpha_1, \cdots \alpha_k$. This is what we would call a polynomial iteration today. What comes as a surprise is that Richardson identifies exactly the problem

he has to solve in order to get a small residual, namely to find a set of α_i's for which $p_k(t)$ deviates the least from zero, but then does not invoke Chebyshev's work to find the solution. He arrives at a certain solution *"by trial"* for a polynomial of degree 7. If we know that the eigenvalues are in an interval $[a, b]$ with $a > 0$, the best solution can be expressed in terms of a Chebyshev polynomial of the first kind, see, e.g., [**64**, Sec. 12.3]. Chebyshev introduced his polynomials in 1854, [**78**], or 56 years prior to Richardson's article, but his paper addressed completely different issues from those with which we are familiar today when analyzing convergence of certain algorithms or when defining iterative schemes such as the Chebyshev iteration. Equally surprising is the fact that Richardson does not seem to be aware of the work by Gauss [**25**] and Seidel [**69**] on iterative schemes. His work is truly original in that it defines a completely new method, the method of polynomial iteration, but misses Chebyshev acceleration as we know it today.

It was much later that the missing part was completed in the work of Shortley [**71**], Sheldon [**70**], and finally Golub and Varga [**37**] and von Neumann in an Appendix of [**10**]. This work also lead to a second-order Richardson iteration to accelerate the "basic" iteration $u^{(k+1)} = Gu^{(k)} + f$ which takes the following form:

$$u^{(k+1)} = \rho \left[\gamma(Gu^{(k)} + f) + (1 - \gamma)u^{(k)}\right] + (1 - \rho)u^{(k-1)}$$

where, unlike the Chebyshev method, the parameters ρ and γ are fixed throughout the iteration.

4. 1930s–1940s: Southwell

Iterative methods were popularized by a series of papers, e.g., [**1**, **17**, **28**, **38**, **75**], and books [**76**, **77**] by Richard Southwell and co-workers who put these methods to use for solving a wide range of problems in mechanical engineering and physics. A good survey of developments with relaxation methods with a summary of the problems successfully solved by these methods up to the late 1940s is given by Fox [**27**]. Southwell defined various refinement techniques to standard relaxation, including block-relaxation (called group relaxation [**75**]) for example. However, his biggest contribution was to put these techniques in perspective and to show their effectiveness for handling a large variety of realistic engineering and physical problems, thus avoiding the use of direct solution methods. Many of the problems tackled were challenging for that period.

5. The SOR era

Later toward the mid-20th century the observation was made that the convergence of a relaxation procedure could be significantly accelerated by including an *over-relaxation* parameter. In the language of the iteration (2.4) Over-relaxation (Young and others) is based on the splitting

$$\omega A = (D - \omega E) - (\omega F + (1 - \omega)D),$$

resulting in a scheme known as the Successive Over-Relaxation (SOR) method. The 1950s and early 1960s marked a productive era for iterative methods that saw an in-depth study of this class of techniques initiated by David Young and Stanley Frankel. In a 1950 article, Frankel [**29**] described the "Liebmann" method, which was just the cyclic relaxation process described by Seidel, along with an "extrapolated Liebmann method" which is nothing but the SOR scheme. He discusses the

parameter ω (called α in his paper) and obtains an optimal value for standard finite difference discretizations of the Laplacean. This particular topic received a rather comprehensive treatment by David Young in his PhD thesis [**84**] who generalized Frankel's work to matrices other than those narrowly targeted by Frankel's paper. The SOR method and its variants, became quite successful, especially with the advent of digital computing and they enjoyed a popularity that lasted until the 1980s when preconditioned Krylov methods started replacing them. Here is what Varga [**81**] said about the capabilities of these methods in the year 1960: *"As an example of the magnitude of problems that have been successfully solved by cyclic iterative methods, the Bettis Atomic Power Laboratory of the Westinghouse Electric Corporation had in daily use in 1960 a 2-dimensional program which would treat as a special case Laplacean-type matrix equations of order 20,000."*

He then adds in a footnote: (paraphrasing) that the program was written for the Philco-2000 computer which had 32,000 words of core storage (32Kwords!) and *"Even more staggering":* Bettis had a 3-D code which could treat coupled matrix equations of order 108,000. This reflects the capability of iterative methods and indeed of linear system solvers (direct methods could not handle such systems) at that point in time.

Up to the early 1980s, this was *the state of the art* in iterative methods. These methods are still used in some applications either as the main iterative solution method or in combination with recent techniques (e.g. as smoothers for multigrid or as preconditioners for Krylov methods).

What I call the SOR era culminated with the production of two major books that together give a complete view of the state of the art in iterative methods up to the late 1960s early 1970s. The first is by Richard Varga [**81**] which appeared in 1962 and the second by David Young [**85**] which appeared in 1971.

6. A turning point: The Forsythe article

In 1953, George Forsythe published a great survey article [**26**] in the Bulletin of the American Mathematical Society, with the title: *"Solving linear algebraic equations can be interesting"*. The paper is rather illuminating as much by the breadth of its content as by its vision. In it Forsythe mentions a new method, called the Conjugate Gradient method, that appeared on the horizon. *"It is my hope, on the other hand, to arouse the mathematician's interest by showing (sec. 2) the diversity of approaches to the solution of (1), and by mentioning (secs. 3 to 6) some problems associated with selected iterative methods.* The newest process on the roster, the method of conjugate gradients, is outlined in sec. 7. *Sec. 8 touches on the difficult general subject of errors and "condition," while a few words are hazarded in sec. 9 about the effect of machines on methods."* The title of the article is intriguing but what it is even more so when one reads the comment by the author, is that the title of the submitted manuscript was *"Solving linear systems is not trivial"*. We will probably never know the reason for the change, but it seems clear that in those days, solving linear systems of equations could be thought to be "trivial" from some angle. [4]

[4]When I was working for my PhD in France, I was once asked about the topic of my thesis and when I replied that it was about sparse matrix methods I was told "..but the problem of solving linear systems of equations is solved. Isn't that just tinkering?" Just like Young in [**86**], "though discouraged I continued to work."

George Forsthye (1917-1972) joined Stanford in 1957 (Mathematics) and founded the computer science department, one of the first in the nation, in 1965. Knuth discusses this era in [**43**] illustrating the fact that it was not an easy task to start a computer science department at the time and praising Forsythe's vision. George Forsythe is now considered one of the fathers of modern numerical analysis.

7. In brief: Chaotic iterations

In the early days of electronic computing many people started to see the potential of parallel processing. *Chaotic relaxation* was viewed as a way of exploiting this avenue. It is interesting to see how early this vision of parallelism emerged. Two papers introduced the term *"free steering"* for a relaxation method in which the components to be relaxed are chosen freely, one by Ostrowski in 1955 [**54**] and the other by Schechter in 1959 [**68**]. Both studied convergence for H-matrices. The article by Chazan and Miranker [**13**] in 1969 introduced the term *"chaotic relaxation"* which was adopted for a while until it was replaced later by the term *"asynchronous relaxation"*.

Here is a quote from this paper by Chazan and Miranker that explains the motivation and context of their work: *"The problem of chaotic relaxation was suggested to the authors by J. Rosenfeld who has conducted extensive numerical experiments with chaotic relaxation* [J. Rosenfeld (1967)]. *Rosenfeld found that the use of more processors decreased the computation time for solving the linear system. (...) The chaotic form of the relaxation eliminated considerable programming and computer time in coordinating the processors and the algorithm. His experiments also exhibited a diminution in the amount of overrelaxation allowable for convergence in a chaotic mode."* The article [**62**] by Rosenfeld, mentioned above, seems to be the first to actually implement chaotic iterations on a parallel machine and show their effectiveness and potential.

The Chazan-Miranker article prompted a group from the French school to study the convergence of chaotic iterations, leading to two articles that appeared at the same time: One by Miellou [**49**] and the other by Robert et al. [**59**]. A notable difference between the two papers is that the analysis by Miellou incorporated "delays" in the algorithm which take into account communication times for example. Other than this, both papers dealt with nonlinear equations and both exploited the tool of "vectorial pseudo-norms" (vector functions whose components are norms) which was quite popular at the time. Later, Miellou published a series of articles in the Comptes Rendus de l'Académie des Sciences (Proceedings of the French Academy of Sciences), see, e.g., [**51**, **52**]. The work by Miellou generated an important following in France, with papers that focused on convergence as well as parallel implementations, see, e.g., [**6**, **7**, **9**, **20**, **21**, **35**, **50**]. Some of the work done in France in those days was truly visionary. Discussions that I attended as a student in Grenoble could be tense sometimes, with one camp claiming that the methods were utopian. They were not necessarily utopian but certainly ahead of their time by a few decades. In fact this work has recently staged a strong come back with the advent of very large high-performance computers where communication is expensive, see, for example, [**2**, **14**, **22**, **32**, **46**] among many others.

8. Meanwhile, on the opposite camp

At this point, the reader may be led to believe that direct methods, or direct elimination methods using Gauss' terminology, were about to be abandoned as the success of iterative methods was spreading to more areas of engineering and science. In fact, quite the opposite happened. Developers of direct methods for sparse linear systems became very active starting in the 1960s and the whole area witnessed an amazing progression in the few decades that followed. Here, it is good to begin by mentioning the survey article by Iain Duff [**18**] which had over 600 references already in 1977. Major advances were made for general sparse matrices – as opposed to those matrices with regular structure that came from finite difference techniques applied to PDEs on simple regions. When a sparse linear system is solved by Gaussian elimination, some zero entries will become nonzero and because of the repetitive linear combination of rows the final matrix may loose sparsity completely. A new non-zero entry created by the process is called a "fill-in" and the number of fill-ins created depends enormously on the way the equations and rows are ordered. Then, a big part of the know-how in *sparse direct methods* is to try to find orderings that minimize fill-in.

The discovery of sparse matrix techniques began with the link made between graph theory and sparse Gaussian elimination by Seymour Parter [**55**] in 1961. This paper gave a model of the creation of fill-in that provided a better understanding of the process. Graphs played a major role thereafter but it took some time before a major push was made to exploit this link in the form of a theorem that guarantees the non-creation of fill-ins by judicious reordering [**60**,**61**]. One important feature of sparse direct methods that distinguishes them from iterative methods, is that they are rather complex to implement. Today, it takes man-years of effort to develop a good working code with all the optimized features that have been gathered over years of steady progress. In contrast, it would take a specialist a few days or weeks of work to develop a small set of preconditioners (e.g., of ILU-type) with one or two accelerators. This distinction has had an impact on available software. In particular, sparse direct solvers (SPARSPAK, YSMP, ..) were all commercial packages at the beginning.

A major contribution, and boost to the field, was made in 1981 by Alan George and Joseph Liu who published an outstanding book [**34**] that layed out all that had been learned on sparse direct solution techniques for solving symmetric linear systems up to that point. The book also included FORTRAN routines and this led to the first package, called SPARSPAK [**15**], for solving sparse symmetric positive definite linear systems[5].

The speed with which progress was made at the early stages of research on sparse direct solvers is astounding. Table 1, reproduced from [**33**], shows the evolution of the performance of the minimum degree algorithm, a reordering technique to reduce fill-in in Gaussian elimination from its inception to 1989. With each discovery, or new trick, there is a gain, often quite substantial, in performance, both in the reduction of the number of nonzero entries and the time of the procedure. Since 1989, many more new ingredients have been found that make sparse direct solvers hard to beat for certain types of problems.

[5]As was just mentioned SPARSPACK was a commercial package but the book included listings of the main routines.

TABLE 1. Evolution of the minimum degree algorithm up to 1989 according to [**33**]. Note that the methods are labeled backward with the most advanced one (Md # 1)at the top and the original one (Md # 6) at the bottom. The test matrix comes the discretization of a Laplacean on a 180 by 180 regular grid using a 9-point finite difference scheme. Its size is $n = 32,400$ and it has a total of $289,444$ non-zero entries.

Version	Minimum Degree Algorithm	Off-diagonal Factor Nonz	Ordering Time
Md#1	Final minimum degree	1,180,771	43.90
Md#2	Md# 1 without multiple elimination	1,374,837	57.38
Md#3	Md# 2 without element absorption	1,374,837	56.00
Md#4	Md# 3 without incomplete deg update	1,374,837	83.26
Md#5	Md# 4 without indistinguishable nodes	1,307,969	183.26
Md#6	Md# 5 without mass elimination	1,307,969	2289.44

The merits and disadvantages of direct and iterative methods have been compared since the earliest paper of Gauss, see Section 1. In his 1959 book [**11**, p. 173] Bodewig states that *"Compared with direct methods, iteration methods have the great disadvantage that, nearly always, they converge too slowly and, therefore, the number of operations is large"*. Then he continues that in fact *"For most systems the iteration does not converge at all. The methods for making convergent an arbitrary system are circumstantial."*

The only potential advantage of iterative methods over direct methods he saw was that *"Rounding errors cannot accumulate, for they are restricted to the last operation. So, without doubt, this is an advantage compared with direct methods. Yet this advantage costs probably more than it is worth."*

Later, David M. Young [**85**] states in the first chapter of his book (1971): *"The use of direct methods even for solving very large problems has received increased attention recently (see for example Angel, 1970). In some cases their use is quite appropriate. However,* there is the danger that if one does not properly apply iterative methods in some cases one will incorrectly conclude that they are not effective *and that direct methods must be used. It is hoped that this book will provide some guidance (...)"* A comparison from the opposite camp (George & Liu's book [**34**]) warns that: *"(...) Unless the question of which class of methods should be used is posed in a quite narrow and well defined context, it is either very complicated or impossible to answer."* The authors then give reference to Varga and Young and say that there are no books on direct solvers and *"In addition, there are situations where it can be shown quite convincingly that direct methods are far more desirable than any conceivable iterative scheme."* Surprisingly, this section of the book does not mention the relative ineffectiveness of direct solvers for large 3D problems (though this was clearly known by the authors at the time, see below).

The debate has somewhat diminished recently with the consensus that iterative methods are competitive for 3-D problems but that for 2-D problems the benefits may be outweighed by the lack of robustness of these methods for indefinite problems. The common argument that is given to illustrate this fact is to compare the result of one of the best orderings for regular grids in the 2-D and 3-D cases, as

illustrated in [**34**]. Consider a standard Poisson equation on an $n \times n$ regular grid in 2-D, and then on an $n \times n \times n$ regular grid in 3-D. We call N the size of the resulting system, so $N = n^d$ where d is the space dimension, i.e., $d = 2, 3$. The order of the cost is given by the following table:

	2-D	3-D
space (fill)	$O(N \log N)$	$O(N^{4/3})$
time (flops)	$O(N^{3/2})$	$O(N^2)$

The table shows a significant difference in complexity between the 2-D and the 3-D cases.

A widespread misconception is that *3-D problems are harder just because they are bigger*. In fact they are just *intrinsically* harder as is suggested in the above table. When I teach sparse matrix techniques in a numerical linear algebra course, I often give a demonstration in MATLAB to illustrate direct solution methods. I show a live illustration of using the *back-slash* operation [6] in MATLAB to solve a linear system involving a coefficient matrix that comes from a centered difference discretization of Poisson's equation on a 2-D or a 3-D mesh. The idea is to show that for the same size problem, e.g., 350×350 grid in 2-D versus $50 \times 50 \times 49$ grid in 3-D (leading to a problem of size $N = 122,500$ in each case), the 3-D problem takes much longer to solve. For this example it can take 11 seconds for the 3-D problem and 0.7 second for the 2-D problems on my laptop. What I also tell the audience is that in past years I was gradually increasing the size of these problems as times went down. A decade ago for example, I would have a demo with a problem of size approximately $N = 20,000$ if I wanted not to have students wait too long for the answer. Of course, this gain in speed reflects progress in both hardware and algorithms.

9. One-dimensional projection processes

The method of steepest descent was introduced by Cauchy in 1829 as a means of solving a nonlinear equation related to a problem of the approximation of an integral. A detailed account of the origin of the steepest descent method is given in [**56**] where we learn that Riemann, Nekrasov, and later Debye were also associated with the method. In 1945 Kantorovitch introduced the method in the form we know today for linear systems for SPD matrices:

$$\min_x \ f(x) \equiv \frac{1}{2} x^T A x - b^T x. \tag{9.1}$$

The gradient of the above function is $\nabla f(x) = Ax - b$ which is the negative of the residual $b - Ax$ and so the steepest descent method will just generate an iteration of the form

$$x_{k+1} = x_k + \omega_k r_k$$

where $r_k = b - Ax_k$ and ω_k is selected to minimize (9.1) at each step. Convergence can easily shown for matrices that are symmetric positive definite. Methods of this type are one-dimensional projection methods in the sense that they produce a new iterate $x^{(new)}$ from a current iterate x by a modification of the form $x^{(new)} = x + \delta$ where δ belongs to a subspace of dimension 1. In the case of the steepest descent

[6] In MATLAB a sparse linear system $Ax = b$ can be solved by the command $x = A\backslash b$. This back-slash operation will invoke a sparse direct solver to produce the answer.

method we can write $\delta = \alpha \nabla f(x) = \alpha(b - Ax)$ and it is easy to calculate α if we wish to minimize (9.1).

Simple projection methods of this type for solving linear systems were proposed earlier. For example, in a short paper [**42**], Kaczmarz described a method in which at each step δ is selected to be the vector $a_i = A^T e_i$, the i-th row of A written as a column vector. In this case,

$$x^{(new)} = x + \alpha a_i \quad \alpha = \frac{r_i}{\|a_i\|_2^2} \tag{9.2}$$

in which r_i is the i-th component of the current residual vector $b - Ax$. Equation (9.2) is written in a form that avoids clutter but we note that the indices i of the components that are modified are cycled from 1 to n and this is repeated until convergence. We can rewrite (9.2) as $x^{(new)} = x + \frac{e_i^T r}{\|A^T e_i\|_2^2} A^T e_i$. If x_* is the exact solution and we write $x^{(new)} = x + \alpha a_i$ then we have

$$\begin{aligned} \|x_* - x^{(new)}\|_2^2 &= \|(x_* - x) - \alpha a_i\|_2^2 \\ &= \|x_* - x\|_2^2 + \alpha^2 \|a_i\|_2^2 - 2\alpha \langle x_* - x, a_i \rangle. \end{aligned} \tag{9.3}$$

This is a quadratic function of α and the minimum is reached when

$$\alpha = \frac{\langle x_* - x, a_i \rangle}{\|a_i\|^2} = \frac{\langle x_* - x, A^T e_i \rangle}{\|a_i\|^2} = \frac{\langle A(x_* - x), e_i \rangle}{\|a_i\|^2} = \frac{\langle r, e_i \rangle}{\|a_i\|^2}$$

which is the choice made in the algorithm. In addition, Kaczmarz was able to show convergence. Indeed, with the optimal α, equation (9.3) yields

$$\|x_* - x^{(new)}\|_2^2 = \|x_* - x\|_2^2 - \frac{r_i^2}{\|a_i\|_2^2}, \tag{9.4}$$

showing that the error must decrease. From here there are a number of ways of showing convergence. The simplest is to observe that the norm of the error $\|x_* - x^{(j)}\|$ must have a limit and therefore (9.4) implies that each residual component r_i converges to zero, which in turn implies that $x^{(j)}$ converges to the solution. The method is motivated by a simple interpretation. The solution x is located at the intersection of the n hyperplanes represented by the equations $a_i x - b_i = 0$ and the algorithm projects the current iterate on one of these hyperplanes in succession, bringing it closer to the solution each time.

At almost the same time, in 1938, Cimmino [**16**] proposed a one-dimensional process which has some similarity with the Kaczmarz algorithm. Instead of projecting the solution onto the various hyperplanes, Cimmino generates n intermediate solutions each of which is a mirror image of the current iterate with respect to the hyperplanes. Once these are available then he takes their convex combination. Specifically, Cimmino defines intermediate iterates in the form

$$x^{(j)} = x + 2r_j a_j \tag{9.5}$$

where r_j is the j-th component of the residual $r = b - Ax$, and then takes as a new iterate a convex combination of these points:

$$x_{new} = \sum \mu_j x^{(j)}. \tag{9.6}$$

Details on this method and on the life and contributions of Cimmino can be found in Michele Benzi's article [**8**].

10. Krylov methods take off: The CG algorithm

One-dimensional projection methods and Richardson iteration are of the form $x_{k+1} = x_k + \beta_k d_k$, where d_k is a certain direction that is generated from the current iterate only.

It was Frankel who in 1950 had the idea to extend these to a *second-order* iteration of the form [**29**]

$$x_{k+1} = x_k + \beta_k d_k \quad \text{where} \quad d_k = r_k - \alpha_k d_{k-1}. \tag{10.1}$$

Frankel was inspired by the solution of time-dependent partial differential equations such as the heat equation which allowed him to add a parameter. We can recover the Chebyshev iteration by using constant coefficients α_k and β_k as we saw before. A method of the type represented by (10.1) with constant coefficients α_k, β_k was termed *semi-iterative* method. In his 1957 article Varga [**80**] uses this term for any polynomial method and mentions earlier work by Lanczos and Stiefel. The 1961 paper by Golub and Varga [**37**] explains how Chebyshev polynomials can be used effectively and stably.

The understanding and development of semi-iterative methods is deeply rooted in approximation theory. The residual of the approximation x_{k+1} obtained from Richardson type iteration of the form $x_{k+1} = x_k + \omega_k r_k$, can be shown to be equal to

$$r_{k+1} = (I - \omega_k A)(I - \omega_{k-1} A) \cdots (I - \omega_0 A) r_0 \equiv p_{k+1}(A) r_0$$

where p_{k+1} is a polynomial of degree $k+1$ satisfying the conditin $p_{k+1}(0) = 1$. One can therefore design effective iterative schemes by selecting polynomials of this type that are small on a set that contains the spectrum of A. Many papers adopted this approximation theory viewpoint. This is most apparent in Lanczos' work. Thus, the remarkable 3-term recurrence obtained by Lanczos to generate an orthogonal basis of the Krylov subspace is a consequence of the Stieljes procedure for generating orthogonal polynomials. Magnus Hestenes [UCLA] and Eduard Stiefel [ETH, Zürich] developed the method of Conjugate Gradient independently. The article [**66**] describes how the two authors discovered that they both developed the exact same method independently at the occasion of a conference held at UCLA in 1951, see also [**36**]. Lanczos developed another method that exploited what we now call Lanczos vectors, to obtain the solution from the Krylov subspace that has the smallest residual norm. His paper [**44**] appeared within 6 months of the one by Hesteness and Stiefel. The method developed by Lanczos is mathematically equivalent to what we would call the Minimal Residual method today, but it is implemented with the Lanczos procedure.

Though not perceived this way at the time, the conjugate gradient method was the single most important advance made in the 1950s. One of the main issues with Chebyshev semi-iterative methods is that they require fairly accurate estimates of extremal eigenvalues, since these define the interval in which the residual polynomial is minimized. The conjugate gradient method bypassed this drawback – but it was viewed as an unstable, direct method. Engeli et al. [**23**] were the first to view the method as an iterative process and indicated that this process can take $2n$ to $3n$ steps to "converge."

The method laid dormant until the early 1970s when a paper by John Reid [**57**] showed the practical interest of this iterative viewpoint when considering large sparse linear systems. With the advent of incomplete Cholesky preconditioners

developed by Meijerink and van der Vorst in 1977 [**48**], the method gained tremendous popularity and ICCG (Incomplete Cholesky Conjugate Gradient) became the de facto iterative solver for the general Symmetric Positive Definite case.

11. Krylov methods: The nonsymmetric period

Nonsymmetric linear systems were given less attention right from the early days of iterative methods. In his 1952 paper, Lanczos [**44**] discusses a method that is essentially equivalent to what we now call the BiCG algorithm and then drops the method by stating: *"... let us restrict our attention to the symmetric case (Normal equations.)"*.

However, the demand for nonsymmetric solvers started to strengthen when applications in aerospace engineering for example gained in importance. Thus, the success of the CG method led researchers to investigate Krylov subspace methods for the nonsymmetric case.

It was only in 1976 that Fletcher [**24**] introduced the *BiCG* method, which was based on the Lanczos process. BiCG uses two matrix-vector products: one with A and the other with A^T. However, the operations with A^T are only needed to generate the coefficients used for the projection and these operations were therefore viewed as wasteful. A number of methods later appeared whose goal was to avoid these products. The first of these was the Conjugate Gradient Squared (CGS) [**74**] developed by Sonneveld in 1984. Then came BiCGSTAB [**79**] in 1992, along with variants, e.g., [**72**], and QMR [**31**], TFQMR [**30**], QMRSTAB, [**12**] and several others. In the methods just listed no attempt was made to exploit orthogonal bases. In parallel to these efforts, a second class of methods was unraveled that exploited orthogonality, allowing in particular to extract the solution from the Krylov subspace that has the smallest residual norm. The first among these, ORTHOMIN, motivated by problems in reservoir simulation was introduced in 1976 by Vinsome [**82**]. A flurry of activity followed and a number of methods were introduced among which we can cite: the GCG and GCG-LS methods [**4**, **5**], ORTHODIR [**87**], FOM [**63**], GCR [**19**], and GMRES [**65**]. Though GMRES is mathematically equivalent to some of the other techniques developed, it has a few practical advantages both in terms of its memory usage and its numerical behavior and it has therefore been the preferred approach among this second class of methods. These contributions to accelerators for the nonsymmetric case are described in detail in the earlier paper [**66**] which covers the period of the twentieth century. In fact research on accelerators has been less active since 2000 while preconditioners have attracted continued attention. The same survey article [**66**] and others, e.g., [**3**], provide additional details on contributions to preconditioners.

12. Present and future

Modern numerical linear algebra started with the influence of George Forsythe and one could view his 1953 survey paper [**26**] as a sort of road-map. Since then, the field has changed directions several times, often to respond to new demands from applications. So the natural question to ask is *"what next?"* For iterative methods, research is still active in the area of preconditioners for some types of problems (Helmholtz, Maxwell, Structures,...), as well as in developing efficient parallel algorithms. For example, it was noted earlier that asynchronous iterations are back. On the other hand research on accelerators has subsided. Another observation is

that the fields of numerical analysis and numerical Linear Algebra are gradually disappearing from computer science graduate programs. This is unfortunate because some research topics fit better in computer science than in mathematics. Among these topics we can mention: sparse matrix techniques and sparse direct solvers, preconditioning methods, effective parallel solvers, and graph-based methods. Some of these topics may reappear in other areas, e.g. computational statistics, and machine learning, but if they are no longer represented in either computer science or mathematics, there will be a lack of students trained in them.

When trying to answer the questions "What next?" we need to remember that for the bigger part of the 20th century, solution techniques (iterative and direct) were aimed primarily at solving certain types of PDEs, and this was driven in part by demand in some engineering applications, most notably the aerospace, the automobile, and the semi-conductor industries. Therefore, a related question to ask is "What new demands are showing up at the horizon?" Currently, the answer to this question is without a doubt related to the emergence of data mining and machine learning. Conferences that used to bear the title "computational X" in the past are now often replacing this title, or augmenting it, by "machine learning X." Linear algebra is gradually addressing tasks that arise in the optimization problems and the computational statistics problems of machine learning. The new linear algebra specialist encounters such problems as evaluating matrix functions, computing and updating the SVD, fast low-rank approximation methods, random sampling methods, etc. An important new consideration in all of these topics is the pre-eminence of randomness and stochastic approaches. In this context, methods such as the conjugate gradient or GMRES, that are based on global optimality are not adapted to randomness and it may be time to look for alternatives or to reformulate them. There are opportunities also in adapting various techniques learned in linear algebra, and more broadly in numerical analysis, to solve various problems in machine learning. Thus, one can echo the title of Forsythe's 1953 paper [**26**] by saying that *"Solving matrix problems in machine learning can be interesting."*

Acknowledgments

The author is indebted to Michele Benzi and Daniel Szyld for their very careful reading of the manuscript and for making numerous suggestions for improvements.

References

[1] D. N. de G. Allen and Richard Southwell, *Relaxation methods applied to engineering problems. XIV. Plastic straining in two-dimensional stress-systems*, Philos. Trans. Roy. Soc. London. Ser. A. **242** (1950), 379–414, DOI 10.1098/rsta.1950.0005. MR0035197

[2] Hartwig Anzt, Stanimire Tomov, Jack Dongarra, and Vincent Heuveline, *A block-asynchronous relaxation method for graphics processing units*, Journal of Parallel and Distributed Computing **73** (2013), no. 12, 1613 – 1626, Heterogeneity in Parallel and Distributed Computing.

[3] O. Axelsson, *A survey of preconditioned iterative methods for linear systems of algebraic equations*, BIT **25** (1985), no. 1, 166–187, DOI 10.1007/BF01934996. MR785811

[4] O. Axelsson, *A generalized conjugate gradient, least square method*, Numer. Math. **51** (1987), no. 2, 209–227, DOI 10.1007/BF01396750. MR890033

[5] Owe Axelsson, *Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations*, Linear Algebra Appl. **29** (1980), 1–16, DOI 10.1016/0024-3795(80)90226-8. MR562745

[6] M. Bahi and J.-C. Miellou, *Contractive mappings with maximum norms: comparison of constants of contraction and application to asynchronous iterations*, Parallel Comput. **19** (1993), no. 5, 511–523, DOI 10.1016/0167-8191(93)90003-4. MR1221041

[7] Gérard M. Baudet, *Asynchronous iterative methods for multiprocessors*, J. Assoc. Comput. Mach. **25** (1978), no. 2, 226–244, DOI 10.1145/322063.322067. MR494894

[8] M. Benzi, *Gianfranco Cimmino's contributions to numerical mathematics*, Atti del Seminario di Analisi Matematica, Dipartimento di Matematica dell'Universita' di Bologna, pages 87-109, Tecnoprint, Bologna, 2005.

[9] Dimitri P. Bertsekas and Didier El Baz, *Distributed asynchronous relaxation methods for convex network flow problems*, SIAM J. Control Optim. **25** (1987), no. 1, 74–85, DOI 10.1137/0325006. MR872452

[10] A. Blair, N. Metropolis, J. von Neumann, A. H. Taub, and M. Tsingou, *A study of a numerical solution to a two-dimensional hydrodynamical problem*, Math. Tables Aids Comput. **13** (1959), 145–184. MR108885

[11] E. Bodewig, *Matrix calculus*, 2nd revised and enlarged edition, North-Holland Publishing Co., Amsterdam; Interscience Publishers, Inc., New York, 1959. MR0127517

[12] T. F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, and C. H. Tong, *A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems*, SIAM J. Sci. Comput. **15** (1994), no. 2, 338–347, DOI 10.1137/0915023. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992). MR1261457

[13] D. Chazan and W. Miranker, *Chaotic relaxation*, Linear Algebra and Appl. **2** (1969), 199–222, DOI 10.1016/0024-3795(69)90028-7. MR0251888

[14] E. Chow, H. Anzt, and J. Dongarra, *Asynchronous iterative algorithm for computing incomplete factorizations on GPUs*, High Performance Computing. ISC High Performance (Ludwig T. Kunkel J., ed.), Lecture Notes in Computer Science, vol 9137, Springer, 2015.

[15] E. Chu, A. George, J. W.-H. Liu, and E. G.-Y. Ng, *Users guide for SPARSPAK–A: Waterloo sparse linear equations package*, Tech. Report CS-84-36, University of Waterloo, Waterloo, Ontario, Canada, 1984.

[16] G. Cimmino, *Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari*, Ricerca Scientifica, II **9** (1938), 326–333.

[17] D. N. de G. Allen and R. V. Southwell, *Relaxation methods applied to determine the motion, in two dimensions, of a viscous fluid past a fixed cylinder*, Quart. J. Mech. Appl. Math. **8** (1955), 129–145, DOI 10.1093/qjmam/8.2.129. MR70367

[18] I. S. Duff, *A survey of sparse matrix research*, Proceedings of the IEEE, 65 (New York), Prentice Hall, 1977, pp. 500–535.

[19] Stanley C. Eisenstat, Howard C. Elman, and Martin H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal. **20** (1983), no. 2, 345–357, DOI 10.1137/0720023. MR694523

[20] Didier El Baz, *M-functions and parallel asynchronous algorithms*, SIAM J. Numer. Anal. **27** (1990), no. 1, 136–140, DOI 10.1137/0727008. MR1034924

[21] Mouhamed Nabih El Tarazi, *Some convergence results for asynchronous algorithms* (French, with English summary), Numer. Math. **39** (1982), no. 3, 325–340, DOI 10.1007/BF01407866. MR678738

[22] Nahid Emad, S.-A. Shahzadeh-Fazeli, and Jack Dongarra, *An asynchronous algorithm on the netsolve global computing system*, Future Generation Computer Systems **22** (2006), no. 3, 279 – 290.

[23] M. Engeli, Th. Ginsburg, H. Rutishauser, and E. Stiefel, *Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems*, Mitt. Inst. Angew. Math. Zürich. No. **8** (1959), 107. MR0145689

[24] R. Fletcher, *Conjugate gradient methods for indefinite systems*, Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, pp. 73–89. Lecture Notes in Math., Vol. 506. MR0461857

[25] G. E. Forsythe, *Gauss to Gerling on relaxation*, Mathematical Tables and Other Aids to Computation **5** (1951), 255–258.

[26] George E. Forsythe, *Solving linear algebraic equations can be interesting*, Bull. Amer. Math. Soc. **59** (1953), 299–329, DOI 10.1090/S0002-9904-1953-09718-X. MR56372

[27] L. Fox, *A short account of relaxation methods*, Quart. J. Mech. Appl. Math. **1** (1948), 253–280, DOI 10.1093/qjmam/1.1.253. MR29270

[28] L. Fox and R. V. Southwell, *Relaxation methods applied to engineering problems. VII A. Biharmonic analysis as applied to the flexure and extension of flat elastic plates*, Philos. Trans. Roy. Soc. London. Ser. A. **239** (1945), 419–460, DOI 10.1098/rsta.1945.0001. MR0014310

[29] Stanley P. Frankel, *Convergence rates of iterative treatments of partial differential equations*, Math. Tables Aids Comput. **4** (1950), 65–75. MR46149

[30] Roland W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput. **14** (1993), no. 2, 470–482, DOI 10.1137/0914029. MR1204242

[31] Roland W. Freund and Noël M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math. **60** (1991), no. 3, 315–339, DOI 10.1007/BF01385726. MR1137197

[32] Andreas Frommer and Daniel B. Szyld, *On asynchronous iterations*, J. Comput. Appl. Math. **123** (2000), no. 1-2, 201–216, DOI 10.1016/S0377-0427(00)00409-X. Numerical analysis 2000, Vol. III. Linear algebra. MR1798526

[33] Alan George and Joseph W. H. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Rev. **31** (1989), no. 1, 1–19, DOI 10.1137/1031001. MR986480

[34] Alan George and Joseph W. H. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981. Prentice-Hall Series in Computational Mathematics. MR646786

[35] L. Giraud and P. Spiteri, *Résolution parallèle de problèmes aux limites non linéaires* (French, with English summary), RAIRO Modél. Math. Anal. Numér. **25** (1991), no. 5, 579–606, DOI 10.1051/m2an/1991250505791. MR1111656

[36] Gene H. Golub and Dianne P. O'Leary, *Some history of the conjugate gradient and Lanczos algorithms: 1948–1976*, SIAM Rev. **31** (1989), no. 1, 50–102, DOI 10.1137/1031003. MR986482

[37] Gene H. Golub and Richard S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. II*, Numer. Math. **3** (1961), 157–168, DOI 10.1007/BF01386014. MR145679

[38] J. R. Green and R. V. Southwell, *Relaxation methods applied to engineering problems. IX. High-speed flow of compressible fluid through a two-dimensional nozzle*, Philos. Trans. Roy. Soc. London. Ser. A. **239** (1944), 367–386, DOI 10.1098/rsta.1944.0002. MR0010082

[39] Alston S. Householder, *The theory of matrices in numerical analysis*, Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, 1964. MR0175290

[40] C. G. J. Jacobi, *Ueber eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommende linearen Gleichungen*, Astronomische Nachrichten (1845), 297–306.

[41] C. G. J. Jacobi, *Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen* (German), J. Reine Angew. Math. **30** (1846), 51–94, DOI 10.1515/crll.1846.30.51. MR1578455

[42] S. Kaczmarz, *Approximate solution of systems of linear equations*, Internat. J. Control **57** (1993), no. 6, 1269–1271, DOI 10.1080/00207179308934446. Translated from the German. MR1220361

[43] Donald E. Knuth, *George Forsythe and the development of computer science*, Comm. ACM **15** (1972), no. 8, 721–726, DOI 10.1145/361532.361538. Collection of articles in honor of George E. Forsythe (ACM Student Competition Award Papers). MR0386953

[44] Cornelius Lanczos, *Solution of systems of linear equations by minimized-iterations*, J. Research Nat. Bur. Standards **49** (1952), 33–53. MR0051583

[45] H. Liebmann, *Die angenäherte harmonischer Funktionen und konformer Abbildungen (nach Ideen von Boltzmann und Jacobi)*, S. B. Math. Nat. Kl. Bayerischen Akad. Wiss. München (1918), 385–416.

[46] Frédéric Magoulès, Daniel B. Szyld, and Cédric Venet, *Asynchronous optimized Schwarz methods with and without overlap*, Numer. Math. **137** (2017), no. 1, 199–227, DOI 10.1007/s00211-017-0872-z. MR3679933

[47] R. Mehmke and P.A. Nekrasov, *Solutions of linear systems of equations by means of successive approximations*, Matematicheskii Sbornik **16** (1892), 437–459, German translation available in Math-Net.Ru.

[48] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. **31** (1977), no. 137, 148–162, DOI 10.2307/2005786. MR438681

[49] J. C. Miellou, *Algorithmes de rélaxation chaotique à retards*, Rev. Française Automat. Informat. Recherche Operationnelle Sér. **9** (1975), no. R-1, 55–82. MR0440904

[50] J. C. Miellou, D. El Baz, and P. Spiteri, *A new class of asynchronous iterative algorithms with order intervals*, Math. Comp. **67** (1998), no. 221, 237–255, DOI 10.1090/S0025-5718-98-00885-0. MR1432131

[51] Jean-Claude Miellou, *Itérations chaotiques à retards* (French), C. R. Acad. Sci. Paris Sér. A **278** (1974), 957–960. MR0362887

[52] Pierre Comte, *Itérations chaotiques à retards, étude de la convergence dans le cas d'un espace produit d'espaces vectoriellement normés* (French, with English summary), C. R. Acad. Sci. Paris Sér. A-B **281** (1975), no. 20, Aii, A863–A866. MR416012

[53] P. Nékrassov, *Détermination des inconnues par la méthode de moindres carrés dans le cas où le nombre d'inconnues est considérable*, Mat. Sb. **12** (1885), 189–204, In Russian. Available along a French version from http://mi.mathnet.ru/msb7148.

[54] Alexander Ostrowski, *Determinanten mit überwiegender Hauptdiagonale und die absolute Konvergenz von linearen Iterationsprozessen* (German), Comment. Math. Helv. **30** (1956), 175–210, DOI 10.1007/BF02564340. MR76433

[55] S. Parter, *The use of linear graphs in Gauss elimination*, SIAM Rev. **3** (1961), 119–130, DOI 10.1137/1003021. MR143349

[56] Svetlana S. Petrova and Alexander D. Solov′ev, *The origin of the method of steepest descent* (English, with English, French and Russian summaries), Historia Math. **24** (1997), no. 4, 361–375, DOI 10.1006/hmat.1996.2146. MR1488796

[57] J. K. Reid, *On the method of conjugate gradients for the solution of large sparse systems of linear equations*, Large sparse sets of linear equations (Proc. Conf., St. Catherine's Coll., Oxford, 1970), Academic Press, London, 1971, pp. 231–254. MR0341836

[58] L. F. Richardson, *The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses to a masonry dam*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **210** (1910), 307–357.

[59] F. Robert, M. Charnay, and F. Musy, *Itérations chaotiques série-parallèle pour des équations non-linéaires de point fixe* (French, with Czech summary), Apl. Mat. **20** (1975), 1–38. MR373272

[60] Donald J. Rose, R. Endre Tarjan, and George S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput. **5** (1976), no. 2, 266–283, DOI 10.1137/0205021. MR408312

[61] Donald J. Rose and Robert Endre Tarjan, *Algorithmic aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math. **34** (1978), no. 1, 176–197, DOI 10.1137/0134014. MR488988

[62] J. Rosenfeld, *A case study on programming for parallel processors*, Research Report RC 64, IBM, Watson Research Center, Yorktown Heights, New-York, 1967.

[63] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp. **37** (1981), no. 155, 105–126, DOI 10.2307/2007504. MR616364

[64] Yousef Saad, *Iterative methods for sparse linear systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003. MR1990645

[65] Youcef Saad and Martin H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **7** (1986), no. 3, 856–869, DOI 10.1137/0907058. MR848568

[66] Yousef Saad and Henk A. van der Vorst, *Iterative solution of linear systems in the 20th century*, J. Comput. Appl. Math. **123** (2000), no. 1-2, 1–33, DOI 10.1016/S0377-0427(00)00412-X. Numerical analysis 2000, Vol. III. Linear algebra. MR1798516

[67] David Eugene Smith, *Book Review: Briefwechsel zwischen Carl Friedrich Gauss und Christian Ludwig Gerling*, Bull. Amer. Math. Soc. **34** (1928), no. 5, 665–666, DOI 10.1090/S0002-9904-1928-04652-9. MR1561632

[68] Samuel Schechter, *Relaxation methods for linear equations*, Comm. Pure Appl. Math. **12** (1959), 313–335, DOI 10.1002/cpa.3160120208. MR107361

[69] Ludwig Seidel, *über ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineare Gleichungen uiberhaupt, durch successive Annäherung aufzulösen*, Akad. Wiss., Munich, mat.-nat. Abt. Abhandlungen **11** (1874), 81–108.

[70] John W. Sheldon, *On the numerical solution of elliptic difference equations*, Math. Tables Aids Comput. **9** (1955), 101–112. MR74929

[71] George Shortley, *Use of Tschebyscheff-polynomial operators in the numerical solution of boundary-value problems*, J. Appl. Phys. **24** (1953), 392–396. MR55042

[72] Gèrard L. G. Sleijpen and Diederik R. Fokkema, *BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electron. Trans. Numer. Anal. **1** (1993), no. Sept., 11–32 (electronic only). MR1234354

[73] David Eugene Smith, *Book Review: Briefwechsel zwischen Carl Friedrich Gauss und Christian Ludwig Gerling*, Bull. Amer. Math. Soc. **34** (1928), no. 5, 665–666, DOI 10.1090/S0002-9904-1928-04652-9. MR1561632

[74] Peter Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **10** (1989), no. 1, 36–52, DOI 10.1137/0910004. MR976160

[75] R. V. Southwell, *Stress calculation in frameworks by the method of systematic relaxation of constraints*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **151** (1935), 56–95.

[76] R. V. Southwell, *Relaxation methods in engineering science, a treatise on approximate computation*, Oxford University Press, 1940, 1943.

[77] R. V. Southwell, *Relaxation methods in theoretical physics, vol. 2*, Oxford Univ. Press, Oxford, 1946.

[78] P. L. Tchebycheff, *Theorie des mecanismes connus sous le nom de parallélogrammes (mémoires presentés a l'académie imperiale des sciences de St-Petersbourg par divers savants vii, 1854, pp. 539-568)*, Oeuvres de P. L. Tchebycheff (A. Markoff and N. Sonin, eds.), Academie imperiale des sciences, St-Petersbourg, 1899, pp. 125–143.

[79] H. A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput. **13** (1992), no. 2, 631–644, DOI 10.1137/0913035. MR1149111

[80] Richard S. Varga, *A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials*, J. Soc. Indust. Appl. Math. **5** (1957), 39–46. MR0090129

[81] Richard S. Varga, *Matrix iterative analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962. MR0158502

[82] P. K. W. Vinsome, *ORTHOMIN: an iterative method for solving sparse sets of simultaneous linear equations*, Proceedings of the Fourth Symposium on Resevoir Simulation, Society of Petroleum Engineers of AIME, 1976, pp. 149–159.

[83] R. von Mises and H. Pollaczek-Geiringer, *Praktische Verfahren der Gleichungsauflösung*, Zeitschrift für Angewandte Mathematik und Mechanik **9** (1929), 58–77.

[84] David M. Young, *ITERATIVE METHODS FOR SOLVING PARTIAL DIFFERENCE EQUATION OF ELLIPTIC TYPE*, ProQuest LLC, Ann Arbor, MI, 1950. Thesis (Ph.D.)–Harvard University. MR2938019

[85] David M. Young, *Iterative solution of large linear systems*, Academic Press, New York-London, 1971. MR0305568

[86] David M. Young, *A historical overview of iterative methods*, Comput. Phys. Comm. **53** (1989), no. 1-3, 1–17, DOI 10.1016/0010-4655(89)90145-8. Practical iterative methods for large scale computations (Minneapolis, MN, 1988). MR1004694

[87] David M. Young and Kang C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl. **34** (1980), 159–194, DOI 10.1016/0024-3795(80)90165-2. MR591431

[88] R. Zurmühl, *Matrizen*, Springer, Berlin, 1950.

Department of Computer Science & Engineering, University of Minnesota, Twin Cities, Minnesota 55455-0154

Email address: saad@umn.edu