SOLUTION OF LARGE LINEAR EQUATION SYSTEMS AND APPLICATION TO SWITCHING SYSTEMS

by

Rudolf S c h e h r e r

Dipl.-Ing., Scientific Assistant

Institute for Switching and Data Technics

University of Stuttgart

Federal Republic of Germany

Iranian-German Scientific Seminar

Pahlavi-University Shiraz, Iran

April 19 - 23 th, 1969

CONTENTS

I

INTRODUCTION

Many problems in physics and engineering lead to linear equation systems. In many non-linear cases, in which an exact solution is very complicated or not practicable at all, a linearisation yields good approximate results. Therefore very often the solvability of engineering problems depends on the solution of large sets of linear equations. Of course, the upper limit of the rank of an equation system to be solved is heavily influenced by the properties of the computer used, in particular by the storage capacity and the computing speed. However, the method applied and the equation system itself play a very important part, too, in this question.

II

ELIMINATION METHODS

In Fig. 1 a set of linear equations is shown. The

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ \dots\ + a_{1n}x_n - b_1 &= 0 \\
a_{21}x_1 + a_{22}x_2 + \ \dots\ + a_{2n}x_n - b_2 &= 0 \\
\cdots\cdots \qquad\qquad & \cdots\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \ \dots\ + a_{nn}x_n - b_n &= 0
\end{aligned}
$$

Fig. 1

Set of linear equations

classical approach is the well-known solution of such systems by means of elimination, the so-called GAUSS - JORDAN - algorithm. In this method, equations are successively combined in such a way, that in the new equations obtained one coefficient becomes zero. Thus, finally a set of equations with a triangular matrix (as shown in Fig. 2) is obtained, from which the unknowns

$$
\begin{aligned}
c_{11}x_1 + c_{12}x_2 + c_{13}x_3 + \ \ldots \ + c_{1n}x_n - d_1 &= 0 \\
c_{22}x_2 + c_{23}x_3 + \ \ldots \ + c_{2n}x_n - d_2 &= 0 \\
c_{33}x_3 + \ \ldots \ + c_{3n}x_n - d_3 &= 0 \\
\ldots\ldots & \\
c_{nn}x_n - d_n &= 0
\end{aligned}
$$

Fig. 2

Set of linear equations with triangular matrix

can easily be evaluated in turn. This elimination method is known in various modifications which are adapted to special computing facilities or special sorts of equations.

Whereas formerly by manual computation it was only possible to solve systems with up to about ten unknowns (with a reasonable amount of computing work), this limit can now be exceeded by far with the aid of electronic computers. The maximum possible number of unknowns depends mainly on the storage capacity of the computer. For a set of n linear equations in general $n \cdot (n+1) \approx n^2$ storage locations are necessary.

Thus, on a modern computer as e.g. the Control Data CDC 6600 with a core store of 64000 words and an addition time of 1 μs for 12 digits it is possible to solve systems with about 200 unknowns, regarding the fact that programs, monitor etc. must also be stored. Applying the

7 million words magnetic disk memory of this computer, it would be possible to deal with systems of up to 2500 equations. This is, however, not possible because of the immense computing time in the magnitude of more than a hundred hours in this case, caused by the poor speed of the magnetic disk memory.

The nunber of 200 unknowns doesn't seem to be too bad, but in many cases this is not sufficient.

In a modified elimination process, the matrix is subdivided into several submatrices as shown in Fig. 3 .

$$M = \begin{array}{|c|c|c|} \hline M_{11} & M_{12} & M_{13} \\ \hline M_{21} & M_{22} & M_{23} \\ \hline M_{31} & M_{32} & M_{33} \\ \hline \end{array}$$

Fig. 3

Subdivided matrix

This procedure is more complicated. Usually the maximum rank can be enlarged by a factor two or three with this method. Larger systems can also be solved, but only with long computing times. The largest general equation system solved on the CDC 6600 up to now had 600 unknowns.

In special cases the storage requirement of a matrix can be smaller. A very important special kind of matrices are the so-called band matrices. In such a matrix (as shown in Fig. 4) only the elements on the diagonal and

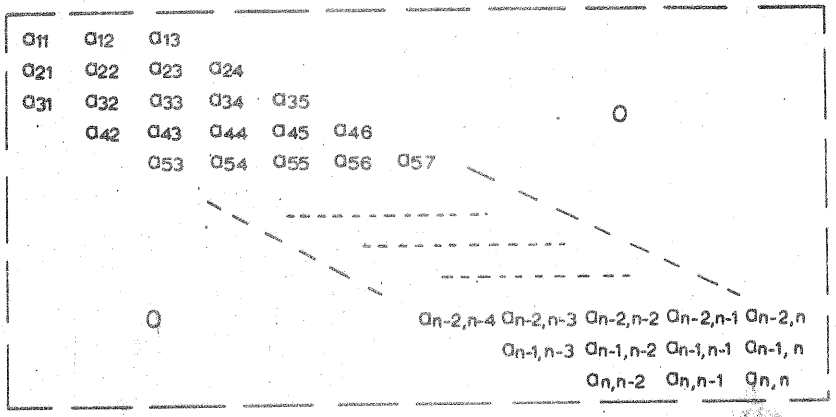

Fig. 4

Band matrix

near the diagonal are different from zero. In a band matrix the zero elements are not changed during the elimination process. Therefore it is not necessary to store them, and the storage requirement can be reduced. Furthermore, the elimination process is shortened in case of band matrices. This means a reduction of computing time, too. Thus, in case of band matrices the rank may have higher values, in particular if the number of terms in each equation is small.

A further, considerable enlargement of the rank can be obtained with band matrices by applying the method of sub-matrices. Then, the maximum rank of a set of equations is mainly determined by the computing time available.

The largest sets of equations with a band matrix evaluated on the CDC 6600 up to now had 6000 unknowns; in each equation were not more than 600 terms. For these systems a computing time of 5 up to 15 hours was necessary.

From this it can be seen, that the use of elimination methods can be extended to fairly large sets of equations, but only on the expense of a large amount of programming work and with long computing times. Furthermore, elimination methods are sensitive to rounding errors, in particular if a matrix is very large or ill-conditioned. Of course, rounding errors can be restricted by double or multiple precision storage, but in case of large sets of equations there will usually be not enough storage capacity available.

III

ITERATIVE METHODS

III.1 GAUSS - SEIDEL - method

A very well known iterative method is the so-called GAUSS - SEIDEL - algorithm.

This method starts with a set of approximate values $x_i^{(o)}$ for the unknowns x_i. (The subscripts indicate the number of the unknown, the superscripts the number of the iteration cycle.) These initial values for the unknowns are chosen according to empirical knowledge, e.g. $x_i^{(o)} = 0$ or $x_i^{(o)} = 1$. In the GAUSS - SEIDEL - method, these initial approximate values are improved successively as follows:

If these approximate values are inserted for instance in the first equation, this equation will in general be not fulfilled, i.e., the left side has the value ("error") e_1 instead of 0. Now, the numerical value of the first unknown $x_i^{(o)}$ is altered in such a way that the left side of this equation becomes 0. For this purpose, the increment Δx_1

must be

$$\Delta x_1 = - \frac{e_1}{a_{11}}$$

Similarly, the approximate values are inserted in the second equation, and the second unknown is altered in such a way that this equation is fulfilled, etc. Here, for the unknown x_1 the new, improved value is taken already. In general, the increment Δx_i which must be added to the value $x_i^{(0)}$ is

$$\Delta x_i = - \frac{e_i}{a_{ii}}$$

where e_i means the deviation of the left side of equation number i from zero. Finally, when the last unknown has been altered by means of the last equation, the first cycle is completed and a new set of approximate values $x_i^{(1)}$ exists. In the following cycles these values can be further improved. The process is stopped, when the increments Δx_i are altogether less than a given bound ε ,

$$\sum_{i=1}^{n} |\Delta x_i| < \varepsilon$$

where ε is e.g. equal to 10^{-5}.

For the convergence of this method the matrix elements in the leading diagonal must dominate the other elements. There are several convergence criteria. E.g. the method is convergent, if in each equation the absolute value of the coefficient of the diagonal is greater than the absolute values of the other coefficients together:

$$|a_{ii}| > \sum_{\substack{k=1 \\ k \neq i}}^{n} |a_{ik}|$$

This is, however, a sufficient condition only, and not a necessary one; that means, the process can converge, even if this condition is not fulfilled. There are also conditions which are more critical, but these are rather complicated and can usually not be checked before starting the iteration process. Thus, sometimes the question of convergence can only be solved by trial: in this case, during the iteration process it is observed whether it converges or not.

If this iteration method does not converge for a certain matrix, then by a method of GAUSS the matrix could be altered in such a way that the iteration process converges. This alteration, however, needs more computing work than the whole solution of the equations by elimination and is therefore usually not of interest.

A disadvantage of the GAUSS - SEIDEL - method is, that in many cases it converges rather slowly. Therefore several modifications with better convergence properties have been developed.

One of these modified methods is the so-called successive overrelaxation method. The next section deals with this method.

III.2 Successive Overrelaxation Method

In the GAUSS - SEIDEL - method, the corrections for the approximate x-values were

$$\Delta x_i = - \frac{e_i}{a_{ii}}$$

This correction is chosen such, that the error e_i is just compensated. In the overrelaxation method, these corrections are multiplied with a factor ω:

$$\Delta x_i = - \omega \cdot \frac{e_i}{a_{ii}}$$

i. e., the errors e_i are not just compensated, but over-compensated.

The convergence criteria for the overrelaxation method are the same as for the GAUSS - SEIDEL - method. If, however, the relaxation factor ω is properly chosen, the overrelaxation method may converge considerably faster as compared with the GAUSS - SEIDEL - method.

In general, it is important to use a suitable ω factor. strictly speaking, the optimum ω factor can be different for each equation and for each iteration cycle. For reasons of simplicity, however, usually a constant ω factor is applied. The optimum ω factor can be found by trial only, except in extremely small equation systems.

The way to find an ω factor which is optimal or nearly optimal could run as follows. The method is carried out with several ω values up to a certain checking point, for instance up to 8 cycles. Then, the method is carried on and finished with the best of these ω values.

When some experiences with this method are available, it is in many cases possible to find empirical approximations for the optimum ω factor with respect to type and magnitude of the equation system.

Generally, ω has values between 0 and 2

$$0 < \omega \leq 2$$

The GAUSS - SEIDEL - method is a special case of the overrelaxation method with $\omega = 1$. Sometimes the optimum ω factor can be less than 1. If an $\omega < 1$ is used, the method is called "underrelaxation".

III.3 Advantages of Iterative Methods

Concerning the advantages of iterative methods it must be pointed out, that they are very well suited for equation systems with a sparse matrix, i. e. a matrix in which a considerable number of elements are zero.

As the matrix is not changed in iterative methods, the zero elements exist through the whole process. This effects a decrease of computing time.

Another advantage of a sparse matrix is that the zero elements need not be stored if an iterative method is applied. It is then sufficient to store the non-zero elements and their subscripts which indicate the position of these elements in the matrix. By this kind of storage it is often possible to solve larger sets of equations.

In large equation systems very often the matrix elements are not written on punched cards but evaluated by the computer according to a special program.

Because the matrix elements are not changed during the iteration process, they can be evaluated whenever they are needed in the iteration procedure and do not need to be stored at all. The only thing that must be stored is the set of unknowns. This is a most important advantage of iterative methods.

The storage necessary is therefore restricted to n locations only (instead of n^2). This enables a considerable enlargement of the maximum rank. With the Telefunken medium speed computer TR 4 of the University Stuttgart with an addition time of 5 μs the number of unknowns can be raised to about 16 000, with the CDC 6600 computer the rank could be raised to about 40 000. These are fairly high values.

A further very important advantage of iterative methods is their insensibility against rounding errors. This is very important.

Theoretically elimination methods yield exact results, whereas iterative methods give approximate results only. If, however, these methods are carried out in practice on a computer with a limited number of digits, e. g. 12 or 15 digits per value, the situation is entirely different. Then the iterative methods can yield results of high precision, whereas the precision of results according to elimination methods can be severely impaired by rounding errors.

IV

APPLICATION TO MODERN SWITCHING NETWORKS

IV.1 Overflow Systems

As a first example let us consider a so-called overflow system. Such a system is shown in Fig. 5. A Poisson traffic

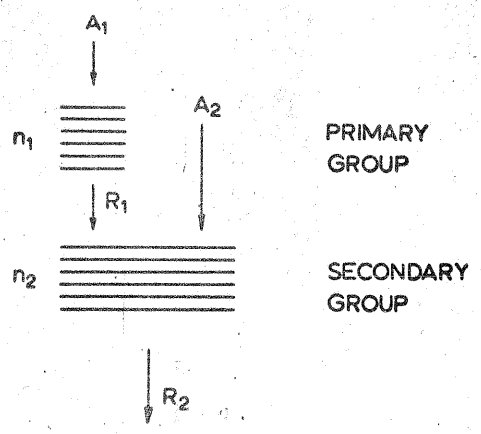

Fig. 5

Overflow system

A_1 is offered to a trunk group, the so-called primary group, with n_1 trunks. (A_1 is the number of calls arriving per unit of time). Calls which can not find a free trunk in this group are overflowing to another group, the so-called secondary group. This so-called overflow traffic R_1 and another Poisson traffic A_2 are offered to the secondary group, which consists of n_2 trunks.

This is a very simple example of an overflow system. In modern telephone systems with alternate routing facilities, overflow systems are used to a very large extent. This has also been pointed out in the paper of U. Herzog ("Problems in Teletraffic Theory").

For an exact computation of the amount of traffic which can not be handled by the secondary group, a set of linear equations must be solved.

In the primary group either no trunk, 1 trunk, 2 trunks etc. or all n_1 trunks can be occupied. Thus (n_1+1) different states are possible in the primary group. For each of these states, in the secondary group there are (n_2+1) possible states respectively. In total there are $(n_1+1).(n_2+1)$ different states.

In this system all calls can reach any free trunk of the primary group or secondary group respectively. Thus, for calculating the probability of loss, the pattern or situation of occupied trunks is irrelevant. Therefore it is sufficient to distinguish between the $(n_1+1).(n_2+1)$ different states of occupation.

Each of these states of occupation occurs with a certain probability. For calculating the probability of loss, the probabilities of all these $(n_1+1).(n_2+1)$ states must be evaluated. Therefore this problem leads to a set of $(n_1+1).(n_2+1)$ equations.

In the investigation of such overflow systems the successive overrelaxation method proved to be very well suited for the solution of the linear equation systems occuring.

As we had to compute values for quite a number of systems, it was well worth finding out the optimum ω factor, which is mainly dependent of the rank for systems of this kind. In Fig. 6 the optimum ω factor is

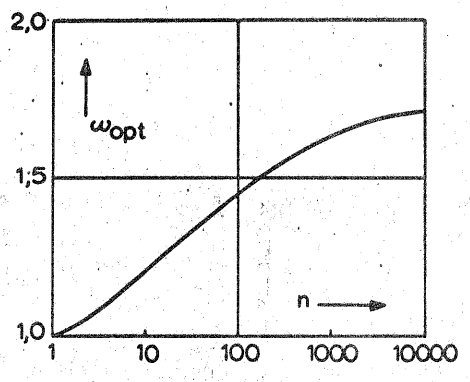

Fig. 6

The optimum ω factor as a function of the number of unknowns

shown as a function of the number of unknowns.

The largest system of this kind investigated up to now yielded a set of 10 000 equations. In each equation there were 5 terms at most, the others being zero. With an ω factor of 1.715 the solution of this set of equations took 85 minutes on the computer TR 4, giving results true to 7 digits.

IV.2 Groups with Limited Access

In a second example let us consider a trunk group with limited access, a so- called grading, as shown in Fig. 7 . In this group there are N trunks, but all calls

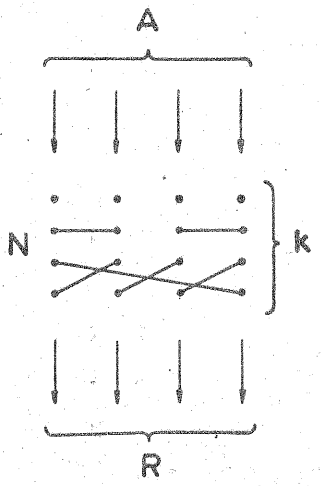

Fig. 7

Group with limited access
(grading)

have access to k trunks only. In this example each call in one of the 4 subgroups has access to 4 trunks only. A Poisson traffic A is offered equally to the various subgroups of this trunk group. The traffic R, which can not be handled, is to be evaluated.

For the exact calculation of such a group with limited access it is not sufficient to consider the number of trunks which are occupied in a certain state, because each of the subgroups, e.g. subgroup number 1 , have access to 4 trunks of the 10 only. Therefore it is essential to distinguish the various possible patterns in which these occupied trunks are situated, i.e., to consider which of the individual trunks are occupied.

The pattern of occupation can be regarded as a snapshot of the momentary state of the system.

As there are N trunks, each of which can be either free or occupied, there are 2^N possible patterns. Thus the investigation of such a group leads to a set of 2^N linear equations.

In groups with special symmetries the rank can be restricted. The patterns can then be classified into groups of patterns with the same probability, and it is sufficient to calculate the probabilities for these groups only instead of all patterns. But this special case shall not be considered here.

The investigation of such groups with limited access is rather an old problem of traffic theory. As, however, the number of equations increases tremendously with increasing number of trunks, the magnitude of such groups which could be investigated without the use of electronic computers was limited to about 3 trunks, corresponding to 2^3 or 8 equations.

When the first computers became available, it was possible to investigate groups with up to 9 trunks, corresponding to 512 equations.

On the TR 4 computer, it is now possible to investigate groups with up to 14 trunks, corresponding to a set of $2^{14} = 16\,384$ equations. With the overrelaxation method this system needed a computing time of 95 minutes with an ω factor of 1.33 .

For such groups with limited access the optimum ω factor is - similarly to the first example - mainly a function of the rank, and thus of the number of trunks.

In Fig. 8 the number of iteration cycles is shown as a function of the relaxation factor ω for a prescribed

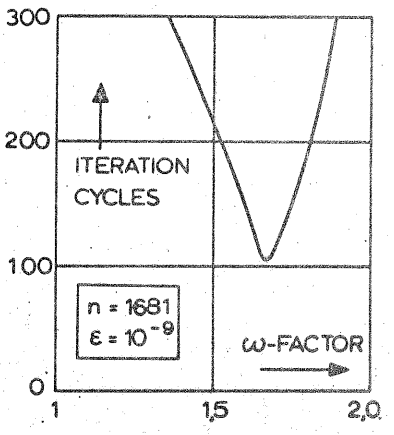

Fig. 8

The number of iteration cycles
as a function of the ω factor

ε value. From this diagram it can be seen that the computing time can increase considerably, e.g. by a factor 3 or more, if the ω factor is not properly chosen. (This diagram corresponds to a set of 1 681 equations).

V

CONCLUSION

In solving sets of equations with special properties, it is often useful to apply iterative methods, in particular the overrelaxation method, instead of the classical elimination process. It is possible to save computing time and storage room, which is essentially. Therefore very large sets of equations can be solved. As examples, applications to modern telephone switching systems have been presented. From these examples it can be seen that sets of 10 000 and more equations can be solved easily by means of the overrelaxation method.