21 [3].—J. K. Reid, Editor, *Large Sparse Sets of Linear Equations*, Academic Press, London-New York, 1971, ix + 283 pp., Price $16—.

Donald J. Rose, Ralph A. Willoughby, Editors, *Sparse Matrices and their Applications*, Plenum Publishing Corporation, New York-London, 1972, xii + 215 pp., 26 cm. Price $11.50.

We have here the proceedings of two conferences, one in Oxford in 1971, the other at the IBM Research Center in Yorktown Heights in 1972. They are more important than most sets of proceedings because, for a while, they will be *the* books in this new area called sparse matrix technology. Although attempts to exploit the zero elements in a matrix date back more then twenty years, the first explicit move to coordinate these efforts was Willoughby's first sparse matrix conference in 1968. Both of the books under review have been carefully edited and there is little overlap between them. They make a good introduction to the subject.

Let us begin with a few words about the subject itself. What are all these research workers trying to do? Mostly, they are trying to solve $Ax = b$ and to update A^{-1} after modifying A. Amazing. Can people still find something new to say on these corny old subjects? The answer is yes, and it is interesting to see how this comes about.

In the beginning came the existence of a solution, when A^{-1} existed, and a formula for the solution in terms of the data (Cramer's rule). Then there was Gaussian elimination for those unfortunates who actually had to find a solution. The advent of digital computers after World War II led people to apply elimination automatically to big 30×30 matrices and this brought on a new worry, namely its stability and then a detailed discussion of minor variations. During the 50's came the first analyses of iterative methods, and a full understanding of the stability question for triangular factorization. However, the difficulties in finding a proper scaling both for equations and unknowns were not yet appreciated. George Forsythe wrote a paper entitled "Solving linear equations can be interesting" which sought to explain this flurry of activity to the rest of the mathematical community. Yet few of the issues confronted in these conferences were discussed in that article. What more could there be to say on the subject?

A basic error to be avoided is the assumption that a numerical analyst is simply devoting his talents to help users save money. What if a computation that used to take 1 hour now takes 10 seconds? The answer is simple. Advances in computers and numerical methods have made *infinite* improvements; they have permitted the accomplishment of tasks that were previously infeasible. That *is* significant. It is this pressure to solve bigger and more complex problems that has led people to return again and again to look in ever increasing detail at such basic tools as a linear equations solver. Once standard algorithms were in good shape, the limitations of storage and execution time came to the fore.

A matrix of order one thousand has a million elements and most computers have fast, random access memories of fewer than seventy thousand cells. If few of the elements are zero, then it is necessary to use a much slower back up storage and one is faced with the problem of how to reorganize the calculation. Moreover, a machine which can perform a million multiplications in a second would still take more than five hours to solve a single system by Gaussian elimination. In many applications, the system is to be solved hundreds if not thousands of times and the project would not be feasible. However, most elements of the big matrices which arise in practice are zero. The problem becomes tractable if the conventional storage of the matrix (one cell for each element) is abandoned and zero elements are not stored.

Band matrices are easy to deal with. In such matrices, the (i, j) element vanishes if $|i - j|$ exceeds some number w which is small compared with the order. More difficult are those matrices in which the few nonzero elements are scattered randomly in the matrix. What is the most efficient way of representing such an array in the memory? This is a nontrivial problem in what is called data structures in computer science departments. How does the array representation affect the implementation of triangular factorization or iterative techniques?

So the reader unfamiliar with sparse matrices will begin to see that the new problems lie less in the domain of traditional numerical linear algebra and more in the realm of software, computer architecture, and graph theory. Naturally, much of the pioneering work has been done by those who had real problems to solve. It is perhaps worth mentioning that most of these large matrices are themselves generated in the computer by other programs and the results used immediately in yet other programs, no part of the process necessarily being seen by the user.

In the Oxford conference, there were papers showing the particular form of sparseness that arises in linear programming, in the static analysis of stresses and strains in structures, in geodesy, in high voltage power systems and in other network flows. There is a review of direct and iterative methods and, inevitably, some of these techniques are described in almost every paper. Mercifully, matrix notation is now universal and this itself indicates how the study of numerical methods has matured. Typical points which are taken up are (i) what is the best way of representing an inverse as a product of elementary factors (the PFI versus the EFI, for those in the trade), (ii) how should the equations be ordered to minimize bandwidth (let the academics worry about stability?), (iii) can standard iterative techniques, like over-relaxation or clever adaptations of the conjugate gradient method, compete with these specialized direct methods? In addition, there was a paper showing the utility of graph theory for discussing sparseness structure and the effect of pivot selection (that is the ordering of the rows and/or columns) on the fill-in of previously zero elements as the elimination proceeds. Another paper describes a programming system, written in FORTRAN, to facilitate the management of the linked list structure used to store the matrix.

A nice feature of the book is the inclusion of the discussions which followed the presentation of each paper.

The second conference, at Yorktown Heights, took advantage of the work done at the previous two conferences and omitted surveys of the major fields of application. The first chapter, by the editors, is a very useful discussion of the whole conference including comments on all of the papers. These were divided into the following categories: Computational Circuit Design, Linear Programming (problems with 7000 rows have been solved), Partial Differential Equations (particularly finite element matrices, whose orders can approach 100,000, and the generation, storage, and fetching of their elements), Special Topics (applications to Photogrammetry, Data Base Systems), Combinatorics and Graph Theory (optimal orderings, minimization of bandwidth), and a Bibliography.

The role of storage recurs throughout the book; what different "levels" of memory are available, how are they related, how can each hierarchy be best exploited?

The work of Gustavson and his colleagues is described here. They began in 1966 with the clever idea of a program whose output is not the solution to $Ax = b$ but instead a loop-free machine language code to compute x taking explicit advantage of the sparseness structure of A. This is dramatically efficient for the common case in which many problems are to be solved but all sharing the same sparseness structure. The personalized compiler has arrived.

B. N. P.

22 [5, 13.20].—F. Bauer, P. Garabedian & D. Korn, *Supercritical Wing Sections*, Springer-Verlag, Berlin, Heidelberg, New York, 1972, v + 211 pp., 26 cm. Price $6.40.

When aircraft fly at high subsonic velocities, an increasing aerodynamic resistance or drag is experienced as the speed of sound is approached. Initially, it was thought that the drag was unbounded at Mach 1 and, hence, that a "sonic barrier" had been discovered. Now we know that the barrier is only a local maximum, that supersonic flight is possible on one side, and that supercritical subsonic flight is possible on the other side. In the latter region, local regions of supersonic flow appear on wings and bodies, and special designs are required to avoid strong shock waves and accompanying adverse effects. As stated in the preface of *Supercritical Wing Sections* "The purpose of this report is to make available to the engineering public mathematical methods for the design of supercritical wings." These are the methods which have been used by the authors to design and analyze two-dimensional shock free supercritical airfoils. The publication, written in the style of a technical report, is divided into three parts giving the mathematical theory, a users' manual for the listed computer programs, and calculated results and examples.

Part I reviews the theoretical formulations with primary emphasis on the airfoil design techniques using the method of complex characteristics. In this ingenious method, the mixed elliptic-hyperbolic partial differential equation is made linear by a hodograph transformation, and, then, all variables are analytically continued from real 2-space to complex 4-space. The resulting equation is purely hyperbolic with distinct eigenvalues except at the sonic locus. The problem posed is of the inverse variety, that is, to find suitable initial data which yield a solution with a desirable body shape and pressure distribution, with the correct flow at infinity and which is shock free (no external limit lines). Selecting the proper initial data is complicated, and a general form is deduced based on knowledge of the incompressible flow past an ellipse and on experience. The singular part of the solution may be written as an integral formula, and the remainder is obtained by numerical integration of linear nonhomogeneous characteristic equations.

Two other theoretical methods which are used are briefly discussed in Part I. An integral boundary layer theory due to Nash and MacDonald is incorporated in the design method to predict the separation point and to make a displacement thickness correction for the airfoil shape. Solutions for off-design conditions are