55

SPARSE MATRICES and Their Applications

A volume in the IBM Research Symposia Series

Edited by Donald J. Rose

SPARSE MATRICES
AND THEIR APPLICATIONS

Proceedings of a Symposium on Sparse Matrices and Their Applications, held September 9-10, 1971, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, the National Science Foundation, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department.

Edited by Donald J. Rose
Department of Mathematics
University of Denver
Denver, Colorado

and

Ralph A. Willoughby
Mathematical Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York

PLENUM PRESS • NEW YORK–LONDON • 1972

Library of Congress Catalog Card Number 71-188917

ISBN 0-306-30587-9

A Division of Plenum Publishing Corporation
227 West 17th Street, New York, N.Y. 10011

United Kingdom edition published by Plenum Press, London
A Division of Plenum Publishing Corporation
Davis House (4th Floor), 8 Scrubs Lane, Harlesden, NW10, 6SE, London, England

Printed in the United of America

PREFACE

This book contains papers on sparse matrices and their applications which were presented at a Symposium held at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York on September 9-10, 1971. This is a very active field of research since efficient techniques for handling sparse matrix calculations are an important aspect of problem solving. In large scale problems, the feasibility of the calculation depends critically on the efficiency of the underlying sparse matrix algorithms.

An important feature of the conference and its proceedings is the cross-fertilization achieved among a broad spectrum of application areas, and among combinatorialists, numerical analysts, and computer scientists. The mathematical, programming, and data management features of these techniques provide a unifying theme which can benefit readers in many fields.

The introduction summarizes the major ideas in each paper. These ideas are interspersed with a brief survey of sparse matrix technology. An extensive unified bibliography is provided for the reader interested in more systematic information.

The editors wish to thank Robert K. Brayton for his many helpful suggestions as chairman of the organizing committee and Redmond O'Brien for his editorial and audio-visual assistance. We would also like to thank Mrs. Tiyo Asai and Mrs. Joyce Otis for their help during the conference and on the numerous typing jobs for the manuscript. A special thanks goes to William J. Turner for establishing the IBM Research Symposia Series with Plenum Press.

D. J. Rose
R. A. Willoughby

January 1972

CONTRIBUTORS

Victor R. Basili, Computer Science Center, University of Maryland, College Park, Maryland

James R. Bunch, Computer Science Department, Cornell University, Ithaca, New York

D. A. Calahan, Department of Electrical Engineering, The University of Michigan, Ann Arbor, Michigan

Elizabeth Cuthill, Naval Ship Research and Development Center, Washington, D. C.

Albert M. Erisman, Boeing Computer Services, Incorporated, Seattle, Washington

David J. Evans, Computing Laboratory, University of Sheffield, Sheffield, England

J. Alan George, Department of Applied Analysis and Computer Science, University of Waterloo, Waterloo, Ontario, Canada

Gary H. Glaser, DBA Systems, Incorporated, Melbourne, Florida

Fred G. Gustavson, Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, New York

G. L. Guymon, Institute of Water Resources, University of Alaska, College, Alaska

Gary D. Hachtel, Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, New York

Eli Hellerman, Bureau of the Census, Computer Applications Working Group, United States Bureau of Commerce, Suitland, Maryland

Gerhard H. Hoernes, Systems Development Division, International Business Machines Corporation, Poughkeepsie, New York

I. P. King, Water Resources Engineers, Incorporated, Walnut Creek, California

W. J. McCalla, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California

Charles K. Mesztenyi, Computer Science Center, University of Maryland, College Park, Maryland

Dennis C. Rarick, Management Science Systems, Rockville, Maryland

Werner C. Rheinboldt, Computer Science Center, University of Maryland, College Park, Maryland

Donald J. Rose, Department of Mathematics, University of Denver, Denver, Colorado

Michael S. Saliba, DBA Systems, Incorporated, Melbourne, Florida

John A. Tomlin, Scicon Limited, London, England

Olof B. Widlund, Courant Institute of Mathematical Sciences, New York, New York

Ralph A. Willoughby, Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, New York

CONTENTS

INTRODUCTION

COMPUTATIONAL CIRCUIT DESIGN

LINEAR PROGRAMMING

PARTIAL DIFFERENTIAL EQUATIONS

SPECIAL TOPICS

COMBINATORICS AND GRAPH THEORY

BIBLIOGRAPHY

INDEX

INTRODUCTION

SYMPOSIUM ON SPARSE MATRICES AND THEIR APPLICATIONS

Donald J. Rose, Department of Mathematics, University of Denver

Ralph A. Willoughby, Mathematical Sciences Department, IBM Research

INTRODUCTION

The main body of this Proceedings consists of 15 papers presented at a Symposium on Sparse Matrices and Their Applications which was held at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York on September 9-10, 1971. The conference was sponsored by the National Science Foundation, Office of Naval Research, IBM World Trade Corporation, and the Mathematical Sciences Department of IBM Research.

Sparse matrix technology is an important computational tool in a broad spectrum of application areas, and a number of these areas are represented in this Proceedings. Of course, the mathematical and computational techniques, presented in the context of a given application, impact many other applications. It is this cross-fertilization that has been a primary motivation for this and two previous sparse matrix conferences† [Willoughby(1968A); Reid(1970A)]. Some fields such as Linear Programming, Power Systems, and Structural Mechanics were systematically surveyed in the first two conferences and are not surveyed here. In addition to the applications themselves, sparse matrix technology involves Combinatorics, Numerical Analysis, Programming, and Data Management.††

†Brackets are used in the introduction to cite references in the unified bibliography at the end of this Proceedings.

††See [Smith (1968A); McKellar and Coffman (1969A); Buchet (1970A); Denning (1970A); Mattson et al (1970A); Moler (1972A)] for a discussion of various aspects of memory hierarchies.

The major ideas in each paper will be summarized in this introduction. These ideas will be interspersed with a brief survey of sparse matrix technology. The papers are ordered alphabetically within groups. The groups are determined partly by application area and partly by mathematical character. Details concerning each paper and related sparse matrix techniques will be given after the listing of the groups of papers in the order in which they occur.

The first group consists of the papers by Calahan, Erisman, Gustavson, and Hachtel. These papers concern problem classes in the field of Computational Circuit Design. Linear Programming is a second application area which involves sparse matrix technology of a very general character.† The papers by†† Hellerman-Rarick and Tomlin comprise the second group.

The sparse matrix technology associated with the field of Partial Differential Equations is the subject of the papers by Evans, George, Guymon-King, and Widlund. Finite element methods are a very active field of research in this area and the papers by George and Guymon-King concern the finite element approach.

The papers by Glaser-Saliba and Hoernes form a Special Topics group. The former paper represents the application of sparse matrices in the field of Analytical Photogrammetry, which is concerned with the determination of reliable metric information from photographic images. The second paper concerns Data Base Systems.

The final group of papers are by Cuthill, Rheinboldt-Basili-Mesztenyi, and Rose-Bunch. These concern the fields of Combinatorics and Graph Theory.

Computational Circuit Design

In the next few paragraphs some aspects of sparse matrix technology, which have been motivated by problems from the field of Computational Circuit Design, will be sketched along with a discussion of the first group of papers. This field is in some sense a problem class representative of many applications. Also it is the most well developed with respect to sophisticated sparse matrix techniques. It is for these reasons that this application area is considered first.

Computational Circuit Design is a very broad and highly developed area, and it is beyond the scope of this introduction to systematically sketch all the various problem types in this field. The interested reader should consult the two special issues of the IEEE Proceedings [IEEE (1967A), (1972A)] and of the Transactions on Circuit Theory [IEEE (1971A)] for pertinent articles and extensive bibliography.

---

† In particular, one can have highly irregular sparseness structures in these first two fields. The matrices are, in general, neither positive definite symmetric nor diagonally dominant.

†† A hyphen is used to connect co-authors.

The algebraic derivation of the sparse linear systems in classical Electrical Network Theory can be found in the survev article [Bryant (1967A)]. A novel tableau approach to this derivation has been motivated by recent advances in sparse matrix technology [Hachtel, Brayton, and Gustavson (1971A)].

One class of problems in computational design [Hachtel and Rohrer (1967A)] concerns the numerical integration of the initial value problem

$$\dot{w} = f(t,w,p), \tag{1}$$

where the vector $w(t_o)$ is specified. The vector, p, of design parameters is to be systematically altered so as to find a specific design vector, p_o, which yields "Optimal" time behavior for system (1).

The unavailability, until recently, of efficient integration techniques for stiff systems of ordinary differential equations† has been a bottleneck in the modeling and computer analysis of problems in many application areas. This is especially true for the class of problems described in the previous paragraph. In that case the efficiency of the integration is a critical factor in the feasibility of the calculation.

The "stiffness" in system (1) manifests itself in the abnormal size (>>1) of the quantity $K = \mathcal{L}\Delta t$, where $\mathcal{L}$ is the Lipschitz constant associated with the w-variation of f, and Δt is the desired average sampling interval for the output of system (1). Efficiency is achieved by using an "essentially" unconditionally stable implicit integration formula for (1) of the form

$$w_{m+1} - \alpha h\dot{w}_{m+1} = R \tag{2}$$

where $t_{m+1} = t_m + h$ and R involves w and $\dot{w}$ for $t \le t_m$.

System (1) is, in general, nonlinear in w, and hence (2) is nonlinear in w_{m+1}. For stiff systems, the usual predictor-corrector methods converge only when h is intolerably small. In this situation, system (2) must be solved by a strongly convergent technique such as Newton's method. Thus, one solves

$$(I - \alpha hJ^{(k)})\Delta w = R + \alpha h\dot{w}^{(k)}_{m+1} - w^{(k)}_{m+1} \tag{3}$$

for Δw and sets $w^{(k+1)}_{m+1} = w^{(k)}_{m+1} + \Delta w$, where $J = \partial f/\partial w$ = Jacobian matrix. A simple starting guess in (3) for w_{m+1} is $w^{(0)}_{m+1} = w_m$ = final iteration vector at $t = t_m$. One can also use an extrapolation procedure for determining $w^{(0)}_{m+1}$.

If the Jacobian matrix is full and $O(n^2)$ elements depend on

†See chapter 6 in [Lapidus and Seinfeld (1971A)].

the current guess for w, then methods like [Broyden (1969A)] are necessary to insure that the amount of work at each step is of order n^2 rather than n^3, where n is the number of components of w.

Fortunately, when n is large in Computational Design problems, the Jacobian is typically sparse.† The sparseness structure can be highly irregular, but computational efficiency is achieved, in spite of this generality, by exploiting the fixed sparseness structure for the Jacobian.

System (3) is of the form

$$Ax = b \qquad (4)$$

where $A = (a_{ij})$, $1 \leq i,j \leq n$. Associated with (4) and a given Jacobian matrix for system (1) is a set of index pairs $S = \{(\mu,\nu) | a_{\mu\nu} \equiv 0\}$. The set S specifies the sparseness structure of the matrix A. There are a number of other ways to specify the sparseness structure for A. For example, one can define for a given sparse matrix class a Boolean adjacency matrix A_s, where $(A_s)_{ij} = 1$ means $a_{ij} \neq 0$. This representation requires n^2 bits regardless of the sparsity of A, and also is not necessarily the most efficient representation from a programming point of view. Threaded index lists with pointers are an important computational tool for dealing with sparseness structure [Ogbuobiri (1970B); Zollenkopf (1970A)]. Gustavson systematically discusses this latter approach, via examples, in his paper in this Proceedings.

A sparse matrix can also be represented by a graph in various ways [Harary (1970A)]. This topic will be considered in more detail when the last group of Proceedings papers are being discussed.

In a given design calculation associated with the initial value problem (1), the system (4) is generated and solved a large number of times. If one fixes, a priori, the order in which the equations and unknowns are processed in Gaussian elimination or in triangular factorization, then the entire sequence of machine operations needed to solve (4) is also determined, a priori, simply from the sparseness structure of A.

At this point, it is convenient to introduce some standard notation which is associated with the Crout form of triangular factorization [Gustavson, Liniger, and Willoughby (1970A)]. Let $A = LU$ where $L = (\ell_{ij})$, $\ell_{ij} = 0$ for $j>i$ (lower triangular), $U = (u_{ij})$, $u_{ii} = 1$, $u_{ij} = 0$ for $j<i$ (unit upper triangular). It is also convenient to introduce a composite L\U matrix as $C = (c_{ij})$ where $c_{ij} = \ell_{ij}$ for $j \leq i$ and $c_{ij} = u_{ij}$ for $j>i$. Each element

---

†That is, only a few unknowns occur in each equation.

of C is generated by a single formula[†]

$$c_{ij} = (a_{ij} - \sum_{k=1}^{m-1} c_{ik}c_{kj})d \qquad (5)$$

where $m = \min(i,j)$, $d = 1$ for $i \geq j$, and $d = c_{ii}^{-1}$ if $i<j$. If $a_{ij} = 0$ and, for $1 \leq k \leq m-1$, $c_{ik}c_{kj} = 0$, then c_{ij} is "logically zero." Otherwise, a reduced formula defines c_{ij}. In this formula only nonzero numbers occur.

In 1966, a highly efficient symbolic factorization program, GNSO (GeNerate SOlve) was created [Gustavson et al (1970A)]. GNSO generates a linear (loop-free) code SOLVE, which is specifically tailored to the zero-nonzero structure of A. The SOLVE program represents a machine language code for computing the reduced formula for each $c_{ij} \neq 0$. The program SOLVE can be very long and as an alternative, two programs SFACT and NFACT were created [Chang (1968A)]. SFACT generates the sparseness structure of C in the context of Tinney's row Gaussian elimination.[††] The program NFACT uses the sparseness information for C to enhance the speed of execution of the numerical elimination.

Gustavson's paper in this Proceedings is a fundamental exposition of the main programming concepts involved in extensions[†††] to his own GNSO program, Chang's programs SFACT and NFACT, and those of Tinney et al. Row Gaussian elimination, with diagonal pivoting in the natural order, is treated both for the unsymmetric and symmetric cases. An ordering program OPTORD is also described.

In the design problem associated with system (1) there is a nested set of computation loops. Because of this, the elements a_{ij} of A have a hierarchy of "variability types" (i.e., $\equiv 0$; $\equiv$ constant; or dependent on p, t, or w respectively). The last three variability types imply increasing frequency of change of the numerical value of the element a_{ij} with that variability type. If the inner loop calculations are not memory-limited, then it is desirable to segment the calculations. This is done in such a way as to avoid repeated calculation of quantities which are constant within those inner loops.

---

† $\sum_{k=\alpha}^{\beta} s_k = 0$ by definition if $\beta<\alpha$.

††Tinney and his colleagues have developed an extensive sparse matrix technology for problems in the field of Power Generation and Distribution.[Sato and Tinney(1963A); Tinney and Walker(1967B); Tinney(1968A); Ogbuobiri, Tinney, and Walker(1970A); Ogbuobiri(1970B)].

††† FORTRAN subroutines, based on these concepts, are in the IBM program product SL-MATH, which was announced recently by IBM World Trade Corporation.

The frequency of change of elements of A can be used in connection with an ordering algorithm. These matters are discussed in [Hachtel, Brayton, and Gustavson (1971A)] and also in the papers by Gustavson and Hachtel in this Proceedings.

In particular, Hachtel considers the case where the vector b in system (4) is sparse and where only a few components of x are required as output. The quantities, generated during the Gaussian elimination process as well as the forward and backward substitutions, will have an inherited variability type. The forward dependency chain will determine the "data type" of a quantity (e.g., a number may have a t-variable type but a w-data type since this number is needed to generate a w-variable type quantity in some subsequent calculation).

The determination of variable type and data type designation for a quantity is a one-time a priori process which can be exploited in an extension of the GNSO and SFACT calculations. The cost of these symbolic preconditioning calculations is amortized over the number of times system (4) is to be generated and solved for a given specification of A_s and of the variability type of each nonzero a_{ij}.

In the paper by Erisman, a sparse matrix program, TRAFFIC, is described. This program has been designed for a particular application: frequency domain analysis of linear passive electrical networks. System (4) is represented in this class of problems as $Y(\omega)v = c$, where $Y(\omega)$ is an n x n complex symmetric admittance matrix, v is the vector of unknown node to datum voltages, c is the vector of input currents, ω is the frequency and n is the number of nodes.

This class of matrices has the additional property that diagonal pivoting in any order is numerically acceptable. Thus, for any permutation matrix P, $M = PY(\omega)P^T$ has the stable triangular factorization†

$$M = U^T DU \tag{6}$$

where U is unit upper triangular and D is the (complex) diagonal matrix of pivots. One also has $L = U^T D$, and this latter equation is used in a factorization algorithm, which was developed at the Bonneville Power Laboratory. Programming details for this Bonneville algorithm are described in Gustavson's paper.

Again there is a fixed sparseness structure for M in (6) for all values of ω. The matrix P, which orders the equations in $Y(\omega)v = c$, can be determined, a priori, so as to minimize some complexity criterion [Rose (1971A)].

Erisman describes and motivates the techniques built into the program TRAFFIC, and, at the end, an illustrative example for a large problem (3300 order with 60,000 nonzero elements) is given.

† Each of the papers in the Proceedings has its own set of notations and these are, in general, different from the notation here in the introduction. The superscript T refers to the transpose operation.