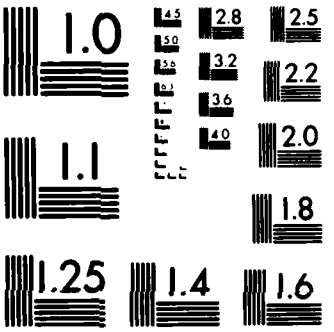
AD-A151 270 VERY LARGE LEAST SQUARES PROBLEMS AND SUPERCOMPUTERS 1/1
(U) PURDUE UNIV LAFAYETTE IN DEPT OF COMPUTER SCIENCES
J R RICE 31 DEC 84 ARO-20219.4-MA DAAG29-84-M-0194
UNCLASSIFIED F/G 9/2 NL



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(2)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ARO 20219.4-MA
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): Very Large Least Squares Problems and Supercomputers
5. TYPE OF REPORT & PERIOD COVERED: Book chapter, 1984
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): John R. Rice

AD-A151 270

8. CONTRACT OR GRANT NUMBER(s): DAAG29-84-M-0194

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Purdue University of Parallel and Vector Computing, Computer Sciences, Purdue University, W.Lafayette, IN 47907
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Post Office Box 12211, Research Triangle Park, NC 27709
12. REPORT DATE: December 31, 1984
13. NUMBER OF PAGES: 16

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Supercomputers, workshop, least squares problems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is a summary of an intensive workshop held at Purdue University on November 3-4, 1983. Eleven people of different backgrounds (numerical analysis, supercomputers, photogrammetry, geodetics and molecular structures) met to exchange ideas and to evaluate present and future developments.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

VERY LARGE LEAST SQUARES PROBLEMS AND SUPERCOMPUTERS

John R. Rice
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

1. SUMMARY AND PRINCIPAL CONCLUSIONS

This paper comes from a workshop held at Purdue University and reflects the participants views as filtered by the author. The principal participants are listed at the end.

The goals of the workshop were: (a) to provide an interchange between different groups working on very large least squares problems, (b) to provide an interchange between computer scientists involved with supercomputer systems and scientists using supercomputers and (c) to assess the state of the art in solving very large least squares problems. This workshop is one of a series held by the Purdue Center for Parallel and Vector Computing and supported by the Army Research Office, the National Science Foundation and the Office of Naval Research. The number of participants was kept small so as to allow for discussions in depth and complete expressions of views.

Twelve sources of very large least squares problems were identified ~~(see Section 2)~~, the ones principally involved in the discussions were:

Geodetic surveys,
Photogrammetry,
Molecular structures,
Gravity field of the earth, and
Partial differential equations.

A brief review of the current methods used in these problems is given in Section 3.

There were extensive discussions of issues of both a general nature and specific to the very large least squares problem. These issues involved the least squares problems, methods for their solution, the use of supercomputers and future developments. These are detailed in Section 4, the principal conclusions are summarized below. This paper summarizes lengthy discussions and reflects their general tenor; no participant is likely to agree in detail with all statements made here.

A. *Problems.* There are several important least squares problems that require supercomputer power. There is substantial similarity in the structure of the problems from different areas; the matrices possess a block structure (sometimes at two levels) which reflects a "local connection" nature in the underlying physical problem.

B. *Methods.* Most of the standard least squares methods are being used somewhere. There is a definite need for a comprehensive software package for least squares that includes sparse matrix facilities.

C. *Computations*. Programming effort is more often a bottleneck than computer time, but neither are likely to be dominant, (the most common dominant effort is to get the data). The preprocessing, postprocessing and general inefficiency in using masses of data on several devices is an important bottleneck. Current pipeline machines and attached processors sacrifice portability and clarity to high efficiency.

D. *Future*. The most promising area for algorithm improvement is in the handling of sparsity. There are important least squares problems that require much more resources (including computer power) than are currently available. Supercomputer architectures are not going to stabilize, so it is important that high level, somewhat architecture independent languages (even a clean Fortran extension) can be used by scientists. Reasonable portability is essential for many reasons, including the success of "national resource" supercomputer centers. *A very critical need is to make supercomputers easier to use.*

2. VERY LARGE LEAST SQUARES PROBLEMS

We first describe the very large least squares problems that were considered. References are given for more information about most of the problems; there were a few of them about which very little was known first hand.

A. The Geodetic Survey Problem. The existing geographical survey points are not completely consistent because of errors in the measurements. A classical procedure has been to adjust the measurements to obtain a best least squares fit to the nonlinear relationships that must hold. The National Geodetic Survey (NGS) currently has a program under way to adjust the measurements for the entire North American continent. This computation will involve about 540 thousand variables and 6.5 million relationships. The adjustment of the geodetic measurements for the entire earth is planned for the future.

See [Golub and Plemmons, 1980, Plemmons, 1979] for details of this problem. Its main features are:

(i) A natural multilevel block structure. Data and computations are usually organized by, say, counties, then by states, then by countries.

(ii) Highly variable accuracy in data. Some survey data is over 50 years old and much less accurate than recent data.

(iii) Very good approximate solutions available for iteration on the nonlinearities.

(iv) Data is expensive to obtain; the least squares computation costs are a moderate part of the whole process.

B. The Photogrammetry Problem. When one takes a series of aerial photographs, one neither knows the locations on the photographs nor the locations of the cameras. One identifies some points on the photographs whose ground locations are known precisely (these are often marked on the ground so they show up clearly in the photos). Overlapping photos are taken showing these points several times; this information is combined with knowledge of the camera properties to create a model of the camera locations. The parameters are then determined as a least squares solution of this nonlinear model. The camera parameters and the ground point parameters are obtained in a simultaneous solution. In large systems a block elimination scheme is often

employed so that a reduced system of only camera parameters is solved first, followed by a "back solution" for the others. The total number of unknown parameters (6 per camera, 3 per ground point) can number in the hundreds for a modest system, and in the thousands for a large system.

Similar computations occur in the precise measurement of the position and shape of large structures such as radio telescopes.

The main features of this problem are:

(i) There is a natural block structure within the least squares normal equations' coefficient matrix. Each photograph and each ground point contribute such a block. Non-zero off-diagonal terms are limited to a band which arises from the "local connection" nature (similar to the geodetic problem) of the photographs and ground point.

(ii) Raw data is collected in two places: (a) obtaining the photographs, and (b) measuring the point locations on the photographs. Obtaining the data is an expensive process, however individual point measurements may be repeated or added relatively inexpensively. The least squares computation is the other major step in producing the final results.

C. The Molecular Structures Problem. The linear least squares problem is in the inner loop of a complex process depicted by the following (simplified) steps.

1. Collect Data
   Grow single crystals of a pure macromolecular substance of sufficient size
   Obtain x-ray diffraction patterns
   Preprocess pictures and use symmetry to enhance data quality

2. Determine approximate molecular structure
   Uses various chemical and physical procedures plus considerable analysis of data

3. Create nonlinear least squares problem
   Include basic covalent process of atomic interactions
   Include 10 to 15 types of "restraints" which incorporate various chemical and molecular facts.

4. Iterate
   Linearize problem by Newton's method
   Discard "most" terms in the Jacobian matrix J
   Solve $J*\Delta x=b$ for the Newton step as a linear least squares problem

5. Reconstruct Computed Molecule
   The numbers are postprocessed to produce a visual model

6. Evaluate Computed Molecule
   The computed models and observed electron densities are compared visually. If they agree to within the uncertainty of the nonlinear least squares, the model is accepted
   Otherwise:

identify water and other solvent structures
reorient certain submolecules
add or delete atoms or submolecules
add restraints on the structure
go back to step 3

There are three positional variables for each atom in the molecule; a simple molecule has a few hundred atoms, a complex one (e.g. a small virus) has a few million. Current work involves molecules with several tens of thousands of atoms. This least squares problem is thus embedded in a large, complex scientific project. The linear least squares problem in the inner loop may take 10-30 minutes on a Class VI computer and the nonlinear iteration requires solving many of these. Even so, this is not necessarily the dominant part of the computation. There may be a stack several feet high of X-ray films with thousands of information spots on each; runs to preprocess this data can require over many hours or a day even on a Class VI computer. See [Hendrickson and Konnert, 1980] for more details on the overall problem and [Blumdell and Johnson, 1976] for more details on the mathematical model and least squares problem.

This problem may be interpreted as 2 1/2 dimensional, the molecule is like a long sausage that winds around itself. Most of the terms in the model refer to local relationships (positions or angles) along the molecule. With an appropriate numbering of the atoms, these relationships produces a "local connection" nature in the least squares problem. All of the important local connection terms in J are near the main diagonal and the others are negligible. However, where the sausage folds over itself, there are non-local (in the numbering system) effects. The folding is not random, one of the major unsolved problems in molecular structures is how and why these giant molecules fold. These non-local terms are in the "restraints" and are a crucial part of determining the structure; they produce "randomly" scattered small blocks away from the main diagonal of J.

The main features of this problem are:

(i) There are several large scale computational steps involved.

(ii) The least squares problem has a local-connection structure modified by a relatively small number of other terms.

(iii) The blocks in the matrix J are small (3 by 3 to 20 by 20 or so).

(iv) There is only a very rough initial approximation for the nonlinear problem, it probably has terms missing (at least in the beginning) so the least squares residuals are not "small".

(v) The outer loop involves someone visually comparing electron density maps with the current model, usually using a computer graphics system. This is the most time consuming aspect of the work.

D. Gravity Field of the Earth. There is a standard model of gravity using spherical harmonics which is derived from viewing the earth as a homogeneous ellipsoidal planet. As more accuracy is desired, one adds more terms to compensate for the nonhomogeneous mass distribution and the actual shape of the earth. NASA has a mission GRM (Geopotential Research Mission) to collect a massive amount of near earth data to be used to determine many thousands of terms in the expansion in spherical harmonics. This will be a standard least squares fitting problem, it is not a sparse matrix problem because the spherical harmonics do not have any "local support" behavior.

An alternative (not part of NASA's plan) is to use a piecewise polynomial representation of the gravity field. The idea behind this is that the detailed effects of irregular shapes and masses are not well modeled by spherical harmonics and one is going to obtain the usual slow convergence properties of polynomial and trigonometric approximations will lead to a least squares problem with quite regular sparsity structure. For more details on this problem see [NASA,1982], [Moore et al,1982].

The main features of this problem are:

(i) A massive amount of data, very expensive to collect.

(ii) Very regular and uniform structure in the data, the problem and the underlying models.

(iii) The classical model leads to a full matrix least squares problem.

(iv) The possible piecewise polynomial model leads to a sparse least squares problem with a very regular structure. The blocks in the matrix are relatively small, about 10 by 10.

E. **The Least Squares Method for Partial Differential Equations** (PDEs). There is a classical least squares (finite element) method for solving PDEs that is rarely used in practice. It is closely related to other widely used methods (e.g. collocation and Galerkin) and the probable reason for its "neglect" is that people feel that it offers no apparent advantage over the more standard methods. See [Rice, 1983] for a discussion of this method.

This method would be used primarily with piecewise polynomial basis functions which would lead to least squares problem with a regular block sparsity structure, similar to that which appears in the more standard PDE methods. The number of unknowns can easily reach 1 million for three dimensional PDEs, there would be a small number (1 to 10) of equations per unknown.

The principal features of this problem are:

(i) There is very little data, the equations are generated mathematically.

(ii) There is a very regular block sparsity structure to the problem. The blocks are small to moderate in size (4 by 4 to 50 by 50).

(iii) The number of equations can be very large, there are applications where one solves a large sequence of very similar problems.

F. **Tomography.** This is a specialized application where one reconstructs an object by taking X-ray cross sections. It is similar to data fitting in that one has a fixed number of data from a continuum; it differs from data fitting in that one observes various linear functionals (e.g. integrals) from the continuum rather than actual values. See [Herman, 1976, 1978 and 1980] for more information.

The principal characteristics of these problems are:

(1) The systems of equalities and inequalities are huge, order about a million.

(2) The sparsity is somewhat haphazard, less than 1 per cent of the matrix elements are non-zero.

(3) The principal computational tool is the row-action method, see [Censor, 1981].

G. **Force Method in Structural Analysis.**

There are two principal methods of matrix structural analysis, the displacement (or stiffness) method and the force (or flexibility) method. The force method has certain advantages for multiple redesign problems or nonlinear elastic analysis because it allows the solution of modified problems, by least squares computations, without restarting the total computation from the beginning. This can result in significant savings for large scale problems. See [Kaneko, Lawo and Thierauf, 1982] and [Kaneko and Plemmons, 1984] for details.

The main features of this problem are:

i. The force method consists of two stages. Stage 1 involves the computation of a basis matrix B for the null space of the equilibrium matrix E for the structure and stage 2 involves the solution of a certain least squares problem with B serving as the observation matrix. B can be dense even though E is sparse, depending upon the method for computing B.

ii. There is very little data. The elements of E are generated mathematically and B is computed from E.

iii. Engineering substructuring methods can lead to a block angular form for the least squares matrix B, similar in form to those of the observation matrices in the Geodetic and Photogrammetry problems.

H. **Very Long Base Line Problem.** The object is to measure astronomical distances by using interferometer methods with base lines that are thousands of miles (using geographically separated radio telescopes) or millions of miles (using observations taken at different points on the earth's orbit around the sun). There are enormous quantities of data that are relatively inaccurate.

I. **Digital Terrain Modeling.** The digital terrain modeling problem is a combination of the photogrammetry problem discussed earlier and the surface fitting problem discussed next. The terrain information is obtained by photogrammetry and then a mathematical model is obtained by another least squares fit. In some application the modeling can be done locally which decouples the latter least squares problem and makes it simply a large sequence of independent, small least squares problems.

J. **Surface Fitting.** One has a physical surface where many positions are known. The surface is modeled by piecewise polynomials of modest degree (1 to 3) joined with some smoothness (continuity, perhaps less, perhaps one or two continuous derivatives). The model has parameters which are determined by a least squares fit to the observed data.

The size of these problems commonly varies from rather small, say a few dozen parameters, up to fairly large, perhaps a thousand parameters. One can, of course, visualize almost arbitrarily large problems, especially if one goes to three dimension problems. The matrices involved have the block structure expected from a "local basis" model of the surface. See [Schumaker, 1978].

The principal characteristics of these problems are:

(i) Usually modest to moderate in size, that is 50 to 1000 unknowns and 2 to 5 observations per unknown.

(ii) A fairly regular block structure in the matrices with modest sized blocks, say 4 by 4 to 16 by 16.

K. **Cluster Analysis and Pattern Matching.** Some pattern recognition algorithms are essentially least squares problems (usually nonlinear). One usually has a modest sample of values and a very flexible model with a relatively small number of parameters; a few hundred values and 5 to 50 parameters are common. As we become more adept at these problems, we can expect the size of the problem to grow very substantially.

The principal characteristics of these problems are:

(i) Modest to moderate in size, but potentially quite large.

(ii) Considerable variation in structure as widely different models may be used. Many models probably give full matrices.

3. METHODS AND MATRIX STRUCTURES

The linear least squares problem is formulated mathematically with an n by m matrix $A=(a_{ij})$, unknowns x_i, $i=1$ to m and data b_j, $j=1$ to n. One wishes to solve $A\mathbf{x}=\mathbf{b}$, but $n>m$ so this system is generally inconsistent. Thus one determines the least squares solution $\mathbf{x}$ so that

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_{ij}x_i - b_j\right)^2 = \|A\mathbf{x}-\mathbf{b}\|_2^2 = \text{minimum}$$

In the discussion that follows, we assume that $n>m$ and m is large. See [Lawson and Hanson, 1974], [Rice, 1981] for more information.

A. **The Normal Equations.** A simple analysis shows that the least squares solution x satisfies the linear system

$$A^TA\mathbf{x} = A^T\mathbf{b}$$

which is an m by m system with a symmetric and (normally) positive definite coefficient matrix A^TA. The total work for this solution method is, including forming A^TA, $m^2n/2 + m^3/6$ multiplications. The main advantage of this approach is simplicity, the disadvantages are (i) the computation might be less stable numerically and (ii) any sparsity structure in A is usually destroyed.

B. **The Residual Equations.** Let $r_j = \sum_{i=1}^{m} a_{ij}x_i - b_j$ be the residual of the j-th equation. Then a simple analysis shows that $\mathbf{x}$ and $\mathbf{r}$ solve the system

$$\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix}\begin{pmatrix} \mathbf{r} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$$

This system is larger than the normal equations, $(n+m)$ by $(n+m)$, but retains the sparsity of A. This system is indefinite.

C. **Orthogonalization.** One may apply an orthogonal matrix Q to $A\mathbf{x}=\mathbf{b}$ to obtain

$$Q\,A\mathbf{x} = Q\mathbf{b}$$

and determine Q so that $QA = R$ is "upper triangular". That is

$$R = \begin{pmatrix} T \\ 0 \end{pmatrix}$$

where T is square and upper triangular. One then solves $T\mathbf{x} = \mathbf{b}\cdot$ where $\mathbf{b}\cdot$ is the first n elements of $Q\mathbf{b}$. The elementary reflections or elementary rotation matrices are usually recommended to construct Q. See [Lawson and Hanson, 1978] or [Rice, 1981] for more details. The total work for this solution method is $m^2n - m^3/6$. The main advantages of this method are numerical stability and the potential of using any sparity that A might have, the disadvantage is that it is twice as much work as the normal equations (assuming that m^2n dominates m^3, as it usually does).

D. **Iteration, Splitting and Conjugate Gradient Methods.** Since the normal equations are symmetric and positive definite, most standard iteration methods are applicable. The convergence of such methods can often be accelerated by splitting the problem. Consider a linear system $C\mathbf{x} = \mathbf{d}$ written in the form

$$M\mathbf{x} = N\mathbf{x} + \mathbf{d}$$

Thus C is split into $M - N$ and the idea is to choose M so that $M\mathbf{z} = \mathbf{f}$ is easy to solve and M^{-1} is a good (reasonable?) approximation to C^{-1}. Various iterations can then be defined to use M in a useful way, the simplest is the iteration

$$M\,\mathbf{x}^{(k+1)} = N\,\mathbf{x}^{(k)} + \mathbf{d}$$

Choosing M as diag(C) gives the Jacobi method, choosing M as the lower triangular part of C gives Gauss-Seidel.

A particularly effective iteration is the conjugate gradient method where one takes

$$M\,\mathbf{z}^{(k)} = (\mathbf{d} - C\,\mathbf{x}^{(k)}) = \text{residual at } k\text{-th iteration}$$
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k-1)} + w_{k+1}(\alpha_k\,\mathbf{z}^{(k)} + \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$$

The parameters w_{k+1} and α_k are determined by separate computations, see [Concus, Golub and O'Leary, 1976] for further details.

Iteration with splitting (and the conjugate gradient method in particular) are attractive for the residual equations form of the problem, because the sparsity of A is completely preserved. Even though the residual form involves a much larger matrix than the normal form, it might require much less storage in a computation if the sparsity of A is exploited.

4. DISCUSSION: ISSUES AND RESPONSES

A set of issues was prepared before the workshop and they received extensive discussion. Issues as originally presented are listed along with a summary of the discussions.

PROBLEMS

A. *Do the problems from different areas have similarities?*

There is a surprising amount of similarity. The matrix A can almost always be put in the following form (sometimes called the dual block angular form):

$$\begin{pmatrix} A_1 & & & B_1 \\ & A_2 & & B_2 \\ & & A_3 & B_3 \\ & & & \vdots \\ & & & A_k B_k \\ & & & D \end{pmatrix}$$

This reflects a "local connection" structure in the underlying physical problem (the spherical harmonics expansion of gravity is one exception). There is a wide variation in the number and size of the blocks. Some problems have large block with k modest in size (10-100) while others have much smaller blocks but many more of them. A number of the problems have two levels of sparsity structure. That is, the blocks A_i and/or B_i are themselves large sparse matrices, usually with this same general pattern of sparsity. There might be some difference in the sparsity patterns between the two levels. The molecular structures problem has this structure with a relatively small number of other blocks scattered through the matrix.

B. *What is the scientific significance of these problems?*

Some of these problems are integral parts of large national scientific programs (e.g. geodetic survey, very long base line and gravity model). Others are ubiquitous in some important areas (e.g. PDE computations, digital terrain modeling, structural analysis, photogrammetry). Still others are integral parts of the developing frontiers of significant scientific research programs (e.g. molecular structures, tomography, pattern analysis).

C. *Are there very large least squares problems of potential interest that have not yet been seriously attempted?*

Three problems were mentioned: PDE computations, geological structure (the analog of the gravity problem, but below the surface of the earth) and cell biology (the natural long range extension of the molecular structures problem).

METHODS

D. *What methods are thought to be the most suitable for these problems?*

There is no clear "winner" yet. The exploitation of sparsity is not yet thoroughly explored; different patterns of sparsity give the advantage to different methods. The normal equations and conjugate gradient (applied to the residual equations) are the most widely used. A drawback of the conjugate gradient method is the difficulty in simultaneously obtaining variance and covariance information.

The impact of vector computers will be substantial but, again, no clear pattern has yet emerged. These calculations deal primarily with very long, very sparse vectors. Substructuring is naturally applicable to these problems for the multiprocessor computers. Again, the algorithmic questions are mostly open.

E. *Is it practical to use the same methods - or same algorithms - or same software - in different applications areas?*

There are definite similarities in the problems from different application areas; this implies that similar methods are applicable. There is not enough generally used software to give as real experience in applying the same software in different applications areas. However, limited experience plus informed conjecture suggests that some software can be used widely. Well designed software could be modified or parameterized to give good efficiency in a variety of applications.

F. *How much exchange of know-how is there between scientists in different application areas? between numerical analysts or computer scientists and scientists?*

There is some exchange of know-how, but it is not systematic nor uniform. The amount of isolation among groups interested in essentially the same problem seems to be typical of science in general.

COMPUTATIONS

G. *Is the vectorization of the linear algebra the major step in adapting methods to current supercomputers?*

There is definitely much more to be done than to vectorize the linear algebra (although this must be done also). The principal task is to reorganize the algorithms so as to exploit the natural sparsity in the problems and yet also exploit the vector processing power of the supercomputers. Experiences were reported where it was as difficult to overcome "non-numerical" bottlenecks (like I/O or page thrashing) as to make the arithmetic run fast. The opinion was expressed that obtaining efficient, well organized software is a bigger hurdle than devising vector algorithms or reorganizing algorithms to be vectorizable.

Current supercomputers were strongly criticized for inadequate Fortran support. To obtain good performance on Cyber 205 or Cray 1 requires a lot of detailed idiosyncratic changes in the codes which renders them totally useless for any other computer. The view was expressed that many people do not want to invest years in codes that cannot be used by their colleagues and which become useless once a newer machine is acquired.

H. *Is least squares computation the major part of the total computations?*

The least squares computation is almost always in the "inner loop" of the computations and thus a significant computational expense. However, it is rarely the dominant part of the computation. Input/output, data processing, preprocessing and postprocessing are also significant computations and some applications also involve significant numerical computations of other types (e.g. nonlinear systems of equations).

I. *What is the nature of the difficulty in getting the data for very large least squares computations?*

There are a couple of areas (PDEs and quantum mechanics) where obtaining the data is a minor part of the problem. For most applications, this is a major part of the problem and for some (e.g. geodetic survey, molecular structures determination, and gravity field analysis) the cost of obtaining the data completely dominates the computational (and programming) costs.

J. *How does programming supercomputers for very large least squares computations compare with programming ordinary machines?*

A high level of general dissatisfaction was expressed for programming the current Class VI machines. They were described as "a pain"; the resulting software is totally non-transportable and generally obscure. The attached array processors are no better. This is not inherently the nature of supercomputers; one participant had considerable experience with the TI-ASC machine and felt it was much more "usable" than his current experience with the Cyber 205.

K. *Is computer time a major bottleneck in getting results for these problems?*

Yes, but it is not dominant in most cases. The preprocessing and postprocessing of results tends to require a lot of human attention and involve delays of various kinds (e.g. getting files from one machine to another, getting output plotted, making tapes, etc.). These activities slow down the whole process much more than the few hours that one is waiting for the "scientific computations" to be done. One sometimes has to wait many hours (or even days) to obtain adequate amounts of computer time.

L. *Is programming effort a major bottleneck in getting results for these problems?*

Yes. It is sometimes more of a bottleneck than computer time, but still usually not the dominant factor. There is often considerable difficulty in finding people who have the desired knowledge of supercomputers, programming and the application area.

THE FUTURE

M. *What are the prospects for being able to solve the very large least squares problems at the frontiers of science? Do we need much faster computers - or much faster algorithms - or both?*

The prospects are good. Both faster computers and faster algorithms are needed; neither one obviously dominates the other. It is just as important to have better user interfaces, better languages and supporting tools as it is to have faster computers and algorithms.

N. *Is it more important to make the computer faster or easier to use?*

The question is misleading; the critical task is to make the very fast computers easy to use.

O. *What would be the scientific impact of much greater computational powers in these areas.*

It would do a lot of good (no specific list of impact areas was generated). Perhaps the greatest impact would come from the ability to do conceptually straight forward things better. A great deal of effort is now required to solve a lot of problems that have little technical difficulty or novelty; this is taking away from the time available for problems that require a lot of thought.

P. *What are the prospects of discovering significantly better supercomputer algorithms for least squares?*

They seem good for two reasons. First, one can see that it is possible to devise better ways of handling sparsity, data and memory space. Second, history tells us that it is unwise to believe that better methods will not appear.

WRAP-UP OBSERVATIONS

Q. There is a strong need for a flexible package (or several packages) of sparse least squares routines

R. Supercomputer hardware is not going to stabilize. People cannot rewrite and tailor massive codes for each new architecture (never mind variances on a theme) that appears. Thus scientists must keep programs expressed at high levels and processors for these languages must be developed for each new architecture.

S. The concept of a set of "national resource" supercomputer access sites is not viable without reasonable transportability of working programs among the supercomputers.

5. WORKSHOP PARTICIPANTS

The principal participants in the workshop are listed below with their relevant field of expertise and affiliation.

James Bethel (photogrammetry)	Purdue University
Iquacio Fita (molecular structures)	Purdue University
Dennis Gannon (supercomputers)	Purdue University
Gene Golub (numerical linear algebra)	Stanford University
Wayne Hendrickson (molecular structures)	Naval Research Laboratory
Greg Kramer (applications programmer)	Purdue University
Charles Lawson (numerical linear algebra)	Jet Propulsion Laboratory
Robert Plemmons (matrix computation, geodesy)	North Carolina State

John Rice (supercomputers) — Purdue University
Michael Rossmann (molecular structures) — Purdue University
Ahmed Sameh (supercomputers) — University of Illinois

6. REFERENCES

Abad-Zapatero, C., Abdel-Meguid, S.S., Johnson, J.E., Leslie, W.A.G., Rayment, I., Rossmann, M.G., Suck, D. and Tsukihara, T., *Structure of southern bean mosaic virus at 2.8: A resolution*, Nature (London), 286, (1980), pp. 33-39.

Blundell, T.L. and Johnson, L.N., *Protein Crystallography*, Academic Press, New York and London, (1976).

Brandt, A., McCormick, S. and Ruge, J., *Algebraic multigrid for automatic multigrid solution with applications to geodetic computations*, to appear.

Bjorck, A., *Iterative refinement of linear least squares solutions I*, BIT 7, (1967), pp. 257-278.

Bjorck, A., *Methods for sparse linear least squares problems*, In: Sparse Matrix Computations, (J.R. Bunch and D.J. Rose, eds.), Academic Press, New York, pp. 177-194.

Buehner, M., Ford, G.C., Moras, D., Olsen, K.W., and Rossmann, M.G., *Three-dimensional structure of D-glyderaldehyde-3-phosphate dehydrogenase*, J. Mol. Biol. 90, (1974), pp. 25-49.

Censor, Y., *Row-action methods for huge sparse systems and their applications*, SIAM Review 23, (1981), pp. 444-466.

Concus, P. Golub, G.H. and O'Leary, D.P., *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, In: Sparse Matrix Computation, (J.R. Bunch and D.J. Rose, eds.), Academic Press, N.Y., (1976).

deBoor, C. and Rice, J., *Extremal polynomials with applications to Richardson iteration for indefinite system*, SIAM J. Sci. Stat. Comput. 3, (1982), pp. 47-57.

George, J. and Heath M., *The solution of sparse least squares problems using Givens rotations*, Linear Algebra App. 34, (1980), pp. 69-83.

George, J., Heath, M. and Plemmons, R., *Solution of large-scale sparse least squares problems using auxiliary storage*, SIAM J. Sci. Stat. Comput. 2, (1981), pp. 416-429.

George, J., Golub, G.H., Heath, M. and Plemmons, R., *Least squares adjustment of large-scale geodetic networks by orthogonal decomposition*, to appear.

Golub, G., *Numerical methods for solving linear least-squares problems*, Numer.

Math. 7, (1965), pp. 206-216.

Golub, G. and Plemmons, R., *Large scale geodetic least squares adjustments by dissection and orthogonal decomposition*, Linear Algebra and Appl. 34, (1980), pp. 3-28.

Golub, G. and Reinsch, C., *Singular value decomposition and least squares solutions*, Numer. Math. 14, (1970), pp. 403-420.

Golub, G. and Wilkinson, J., *Note on the iterative refinement of least squares solutions*, Numer. Math. 9, (1966), pp. 139-148.

Heath, M., Plemmons, R. and Ward, R., *Sparse orthogonalization schemes for structural optimization using the force method*, to appear.

Hendrickson, W.A., *A procedure for representing arbitrary phase probability distributions in a simplified form*, Acta Cryst. B27, (1971), pp. 1472-1473.

Hendrickson, W.A., *Radiation damage in protein crystallography*, J. Mol. Biol. 106, (1976), pp. 889-893.

Hendrickson, W.A., *Transformations to optimize the supercompposition of similar structures*, Acta Cryst. A35, (1979), pp. 158-163.

Hendrickson, W.A., Klippenstein, G.L. and Ward, K.B. *Tertiary structure of myohemerythrin at low resolution*, Proc. Natl. Acad. Sci. USA 72, (1975), pp. 2160-2164.

Hendrickson, W.A. and Konnert, J.H., *Stereochemically restrained crystallographic least-squares refinement of macromolecular structures*, In: Biomolecular Structure Conformation, Function and Evolution (R. Srinivasan, ed.) Pergamon, Oxford, 1, (1981), pp. 43-57.

Hendrickson, W.A. and Konnert, J.H., *A restrained parameter thermal factor refinement procedure*, Acta Cryst. A36, (1980), pp. 344-350.

Hendrickson, W.A. and Konnert, J.H., *Incorporation of stereochemical information into crystallographic refinement*, In: Computing in Crystallography (R. Diamond, S. Rameseshan and K. Venkatesan, eds.) Indian Academy of Sciences, Bangalore, (1980), pp. 1301-1323.

Hendrickson, W.A. and Ward, K.B., *Pseudosymmetry in the structure of myohemerythrin*, J. Biol. Chem. 252, (1977), pp. 3012-3018.

Herman, G.T., *An introduction to some basic mathematical concepts on computed tomography*, In: Roentgen-Video-Techniques for Dynamic Studies of Structure and Function of the Heart and Circulation (P. Heintzen, ed.), Georg Thieme Publishers, Stuttgart, Germany, (1978).

Herman, G.T., *Image reconstruction from projections: The fundamentals of computerized tomography*, Academic Press, New York, (1980).

Herman, G.T. and Lent, A., *A computer implementation of a Bayesian analysis of image reconstruction*, Info. and Control 31, (1976), pp. 363-384.

Hume, D., Litsey, J. and Plemmons, R., *Software for ordering large sparse least squares prior to Givens reduction*, Proc. 1981 Army Numer. Anal. Computers Conf., ARO Rpt. 81-3, (1981), pp. 267-282.

Kaneko, I. and Plemmons, R., *Minimum norms solution to linear elastic problems*, Inter. J. Numer. Meth. Engin., to appear.

Lawson, C.L., *Sparse matrix methods based on orthogonality and conjugacy*, JPL TM 33-616, (1973).

Lawson, C.L., *Solving large dense least squares problems using two levels of storage*, JPL internal report CM 366-466, (1979).

Lawson, C.L. and Hanson, R.J., *Solving least squares problems*, Prentice-Hall, (1974).

Lawson, C.L. and Ekelund, J.E., *An out-of-core least squares program*, JPL internal report CM 366-435, (1977).

Lawson, C.L., Hanson, R.J., Kincard, D.R. and Krogh, F.T., *Basic linear algebra subprograms for fortran usage*, ACM Trans. Math. Software, 5, (1979), pp. 308-323.

Moore, J., Smith, D., Keating, T. and Langel, R., *Geopotential research mission (GRM)*, Goddard Space Flight Center, Greenbelt, MD, (1982).

Murthy, M.R.N., Reid, T.J., III, Sicignanco, A., Tanaka, N. and Rossmann, M.G., *Structure of beef liver catalase*, J. Mol. Biol. 152, (1981), pp. 465-499.

NASA, NASA Geodynamics Program, Annual Report 1981, NASA Tech. Memo. 85126, (1982).

Pagano, M., Golub, G. and Luk, F., *A large sparse least squares problem in photogrammetry*, In: Proc. Computer Science and Statistics; Twelfth Annual Symposium on the Interface, Waterloo, Canada, (1979), pp. 26-28.

Paige, C. and Saunders, M., *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software 8, (1982), pp. 43-71.

Rice, J.R., *Matrix computations and mathematical software*, McGraw-Hill, (1981).

Rice, J.R., *Numerical methods, software and analysis*, McGraw-Hill, (1983).

Rossmann, M.G., Abad-Zapatero, C., Murthy, M.R.N., Liljas, L., Jones, T.A. and Strandberg, B., *Structural comparisons of some small spherical plant viruses*, J. Mol. Biol., 165, (1983), pp. 711-736.

Rossman, M.G., Abad-Zapatero, C., Hermodson, M.A. and Erickson, J.W., *Subunit interactions in southern bean mosaic virus*, J. Mol. Biol. 166, (1983), pp. 37-83.

Saad, Y., *Iterative solution of indefinite symmetric linear systems by methods using orthogonal polynomials over two disjoint intervals*, SIAM J. Num. Analys. 20, (1983), pp. 784-811.

Sameh, A., *Solving the linear least squares problem on a linear array of processors*, In: Purdue Workshop on Algorithmically Specialized Computer Organizations, (Gannon, Seigel and Snyder, eds.) Academic Press, (1984).

Schumaker, L.L., *Fitting surfaces to scattered data*, In: Approximation Theory III (Lorentz et al., eds.), Academic Press, (1970), pp. 203-268.

END

FILMED

4-85

DTIC