Matrix Market : a web resource for test matrix collections

Ronald F. Boisvert, Roldan Pozo, Karin Remington
National Institute of Standards and Technology
Gaithersburg, MD 20899 USA.
Email: {boisvert,pozo,karin}@cam.nist.gov

Richard F. Barrett
Los Alamos National Laboratory
CIC-8, MS B272, LANL, Los Alamos, NM 87545 USA.
Email: rbarrett@lanl.gov

Jack J. Dongarra
Univ. of Tennessee at Knoxville and Oak Ridge National Laboratory
Computer Science Department, University of Tennessee at Knoxville, Knoxville, TN 37996 USA. Email: dongarra@cs.utk.edu

Abstract

We describe a repository of data for the testing of numerical algorithms and mathematical software for matrix computations. The repository is designed to accommodate both dense and sparse matrices, as well as software to generate matrices. It has been seeded with the well-known Harwell-Boeing sparse matrix collection. The raw data files have been augmented with an integrated World Wide Web interface which describes the matrices in the collection quantitatively and visually. For example, each matrix has a Web page which details its attributes, graphically depicts its sparsity pattern, and provides access to the matrix itself in several formats. In addition, a search mechanism is included which allows retrieval of matrices based on a variety of attributes, such as type and size, as well as through free-text search in abstracts. The URL is http://math.nist.gov/MatrixMarket/.

Keywords

Numerical linear algebra, mathematical software, sparse matrices, testing and evaluation, reference data, test corpora, World Wide Web

Reprinted from *The Quality of Numerical Software: Assessment and Enhancement* (R.F. Boisvert, ed.), Chapman & Hall, London, 1997, pp. 125–137.

1 INTRODUCTION

A frequently applied method for the testing of numerical software is to exercise it on a battery of representative problems. Often such problems are generated randomly, insuring that a large number of test cases can be applied. Unfortunately, this is rarely sufficient for serious numerical software testing. Errors or, more likely, numerical difficulties typically occur for highly structured problems or for those near to the boundaries of applicability of the underlying algorithm. These parts of the domain are rarely sampled in random problem generation, and hence testing must also be done on problem sets which illustrate special behaviors. These are quite difficult to produce, and, thus, groups of researchers often exchange sample problem sets. Such data sets serve a variety of additional purposes:

- Defining the state-of-the-art.
- Characterizing industrial-grade applications.
- Catalyzing research by posing challenges.
- Providing a baseline of performance for software developers.
- Providing data for users who want to gain confidence in software.

Unfortunately, these collections are often lost when the underlying technology is picked up by the commercial sector, leaving software developers and users without an important tool to use to judge the capability of their products. One of the goals of this project is to identify, preserve, and make readily available such test corpora for use by researchers, developers, and users of mathematical and statistical software. In this paper we describe work that is underway in the area of test data for matrix algorithms.

2 THE MATRIX MARKET

The decomposition, solution and eigenanalysis of systems of linear equations remain important problems in scientific computation for which new algorithms and software packages are continually being developed. In order to make reliable, reproducible quantitative assessments of the value of new algorithmic developments it is useful to have a common collection of representative problems through which methods can be compared. For sparse matrices the Harwell-Boeing Sparse Matrix Collection has served this purpose exceptionally well for some time. Release I of that collection (Duff, Grimes and Lewis, 1989) contains nearly 300 matrices of order 9 to 35,558 from a variety of applications. One of the difficulties with such collections is that their size and diversity makes them unwieldy to manage and use effectively. As a result, the Harwell-Boeing collection has not been used as much as it should, and new matrices have not been regularly added to the collection.

Recent developments in network communications infrastructure, such as the World Wide Web (WWW), are opening up new possibilities for improving the access to and usability of test corpora of this type. In this paper we describe a WWW resource which provides a variety of services based initially upon the Harwell-Boeing collection. We call this resource the Matrix Market.

The Matrix Market is a repository of matrices. Matrices are gathered together into *sets*. Matrices in a set are related by application area or contributed from a single source. Sets can further be grouped into *collections* managed by a single group, such as the Harwell-Boeing collection. Individual matrices may be stored explicitly as dense or sparse matrices, or may be available implicitly via a code which generates them. The matrices can be of a wide variety of types, e.g., real, complex, symmetric, nonsymmetric, Hermitian. Some are representations of nonzero patterns only. Others include supplementary data such as right-hand sides, solution vectors, and initial vectors for iterative solvers. We store matrices and associated material one per file, in a fixed format..

For each matrix we provide a summary page in HyperText Markup Language (HTML) format outlining the properties of the matrix and displaying a graphical representation of its structure. Similarly, we have developed a WWW page for each set, which gives its background (e.g., source and application area), references, as well as a thumbnail sketch of each matrix's nonzero pattern. We maintain a separate database which contains all of the information on these pages in a highly structured form. This allows us to manipulate the data in various ways; for example, all of the matrix and set WWW pages are automatically generated from this database. The database also supports both structured and free-text retrieval.

In the remainder of this paper we describe each of the components of the repository in detail, with emphasis on the case of sparse matrix data. This reflects the organization and implementation of the Matrix Market prototype released in February 1996. The Matrix Market is still under active development, and some of the details of its interface and implementation may evolve over time. In the final section, we outline plans for extending this service to dense matrices and matrix generation software, as well as ideas for use of new WWW technologies such as Java (Flanagan, 1996) to enhance its the usefulness.

3 A MATRIX META-DATABASE

As a collection of test problems becomes large and diverse it becomes increasingly important to have effective tools for finding problems which have particular characteristics of interest. For sparse matrices, these might be things like symmetry, size, and percentage of nonzeros. The Harwell-Boeing exchange format includes a variety of such information; unfortunately, the distribution of this information inside large files of matrix elements makes it difficult to use for efficient searching. In addition, a large variety of additional useful data has only been accessible in printed reports. To allow efficient automation of matrix search and retrieval it is necessary to represent all of this information in a concise and well-defined format.

Our philosophy is to maintain a single central base of meta-data describing the matrix collection in a neutral format. This database should be rich enough to build a variety of applications, such as

1. search and retrieval facilities,
2. generators for matrix and set WWW pages (in HTML),
3. generators for printed catalogs (in LaTeX).

The Matrix Market meta-database consists of two files, one describing properties of individual matrices (the Matrix database) and a second describing properties of sets of

```
id       SHERMAN5
set      SHERMAN
for      Fully implicit black oil model (NX=16, NY=23, NZ=3, NC=3 )
problem  linear system
field    real
symm     unsymmetric
defin    indefinite
struct   sparse
storage  sparse assembled
shape    square
rows     3312
cols     3312
nnzero   20793
nrhs     1
```

Figure 1 Sample entry from the Matrix database.

matrices (the Set database). Each contains a sequence of multi-line records delimited by a blank line (i.e., two successive `newline` characters). Each record contains a list of attribute-value pairs. Each attribute name starts at the beginning of a line, followed by a `TAB` character, and its value. A value may be continued onto successive lines by placing a `TAB` as the first character on each additional line. Attributes may appear in any order.

Many of the attributes represented in the database correspond to information provided in the Harwell-Boeing report (Duff, Grimes and Lewis, 1989). Table 1 lists the attributes currently represented in the Matrix database; Figure 1 provides an example entry. For attributes and an example of the Set database, see Table 2 and Figure 2, respectively.

4 HTML INTERFACES FOR MATRIX TEST DATA

In this section we describe the user interface to the Matrix Market WWW service. The home page (see Figure 3), at `http://math.nist.gov/MatrixMarket/`, provides an overview and an entry to its various component services which we describe below.

4.1 Matrix Home Pages

Each matrix and set in the Matrix Market has a "home page" in HTML format. For matrix sets the home page provides general information such as the application domain, contributor, accession date, and references. A link to a compressed tar file containing all the matrices in the set is provided. In addition, the identifier, title and size of each matrix in the set is then listed, along with a thumbnail sketch (i.e., a small file in GIF format) that indicates the nonzero structure of the matrix. Clicking on the matrix identifier or thumbnail retrieves the home page for that particular matrix.

The home page for an individual matrix provides detailed information about the par-

Table 1 Attributes in the Matrix database

`id`	The unique identifier for the matrix.
`set`	The name of the set containing this matrix (see Table 2).
`for`	A brief phrase describing the application that this matrix comes from.
`problem`	The linear algebra problem that this matrix is intended to be used for. [One of: `linear system`, `eigenvalue`, `least squares`, `optimization`, `structure`. The latter indicates that only the nonzero structure itself is of interest.]
`field`	The arithmetic field that the matrix elements are drawn from. [One of: `real`, `complex`, `pattern`. The latter is used for matrices in which only the nonzero structure is provided.]
`symm`	Indicates the symmetry structure of the matrix (not the storage format). [One of: `unsymmetric`, `symmetric`, `skew symmetric`, `Hermitian`. Also, `unsymmetric with symmetric structure` can be used to indicate that the matrix is nonsymmetric, but has a symmetric nonzero structure.]
`defin`	Indicates definiteness properties. [One of: `positive definite`, `positive semidefinite`, `negative definite`, `negative semidefinite`, `indefinite`.]
`struct`	Indicates the nonzero structure of the matrix. (not the storage format). [One of: `banded`, `sparse`, `tridiagonal`, `block tridiagonal`, `diagonal`, `block diagonal`, `triangular`, `Hessenberg`]. More than one of these may naturally apply, and hence this (attribute,value) pair may be repeated.
`storage`	Indicates the type of Harwell-Boeing storage format used. [One of: `sparse assembled`, `sparse elemental`]
`shape`	Indicates the general shape of the matrix. [One of: `square`, `rectangular`] (This attribute can be inferred from others, it is included here to accommodate matrix generator codes at some later date.)
`rows`	Number of rows.
`cols`	Number of columns.
`nnzero`	Number of nonzero entries.
`nel`	Number of elements (for sparse elemental storage mode only).
`maxel`	Maximum number of elements (for sparse elemental storage mode only).
`nrhs`	Number of right-hand sides included. *Optional.*
`exact`	Number of exact solutions included. *Optional.*
`guess`	Number of starting vectors for iterative methods included. *Optional.*

Table 2 Attributes in the Set database

set	The name of the set.
title	A title for the set. A brief phrase which provides a succinct description of the set.
dis	The application domain (discipline) from which the set is drawn.
from	The name of the person(s) and institution(s) who contributed the set.
ref	Citation of a book or article in which the application or the matrices themselves described.
perf	Notes regarding (a) special properties of the matrices, (b) difficulties that matrix algorithms may experience with these matrices, or (c) data on the performance of specific software on these matrices. *Optional.*
acc	Date contributed to the Matrix Market. (Month and year.)
rem	Remarks. *Optional.*

```
set    SHERMAN
title  Oil reservoir simulation challenge matrices
dis    Oil reservoir modeling
from   Andy Sherman, Nolan and Associates, Houston, TX.
ref    Simon, H.D. (1985). Incomplete LU preconditioned
       conjugate-gradient-like methods in reservoir simulation.
       Proceedings of the Eighth SPE Symposium on Reservoir Simulation,
       Dallas, Feb.10-13, 387-396.
perf   See reference.
acc    Autumn 1984
rem    In the summer of 1984, Andy Sherman of Nolan and Associates,
       Houston, TX, USA, issued a challenge to the petroleum
       industry and the numerical analysis community for the
       fastest solution to a set of 5 systems of linear equations
       extracted from oil reservoir modeling programs. These are
       those five matrices. Each matrix arises from a three
       dimensional simulation model on an NX x NY x NZ grid using a
       seven-point finite-difference approximation with NC
       equations and unknowns per grid block. The corresponding
       right-hand side vector is also supplied.
```

Figure 2 Sample entry from the Set database.

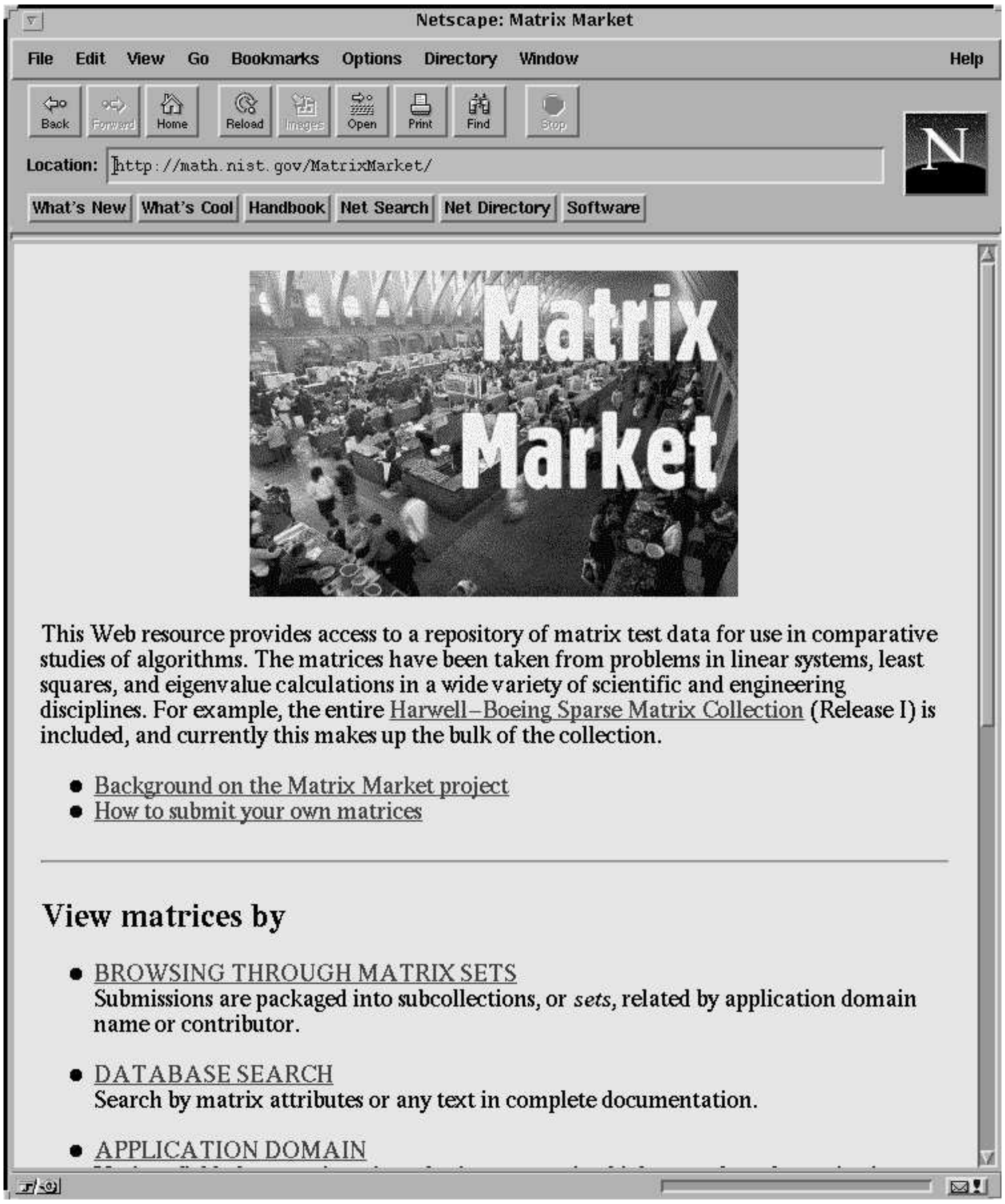
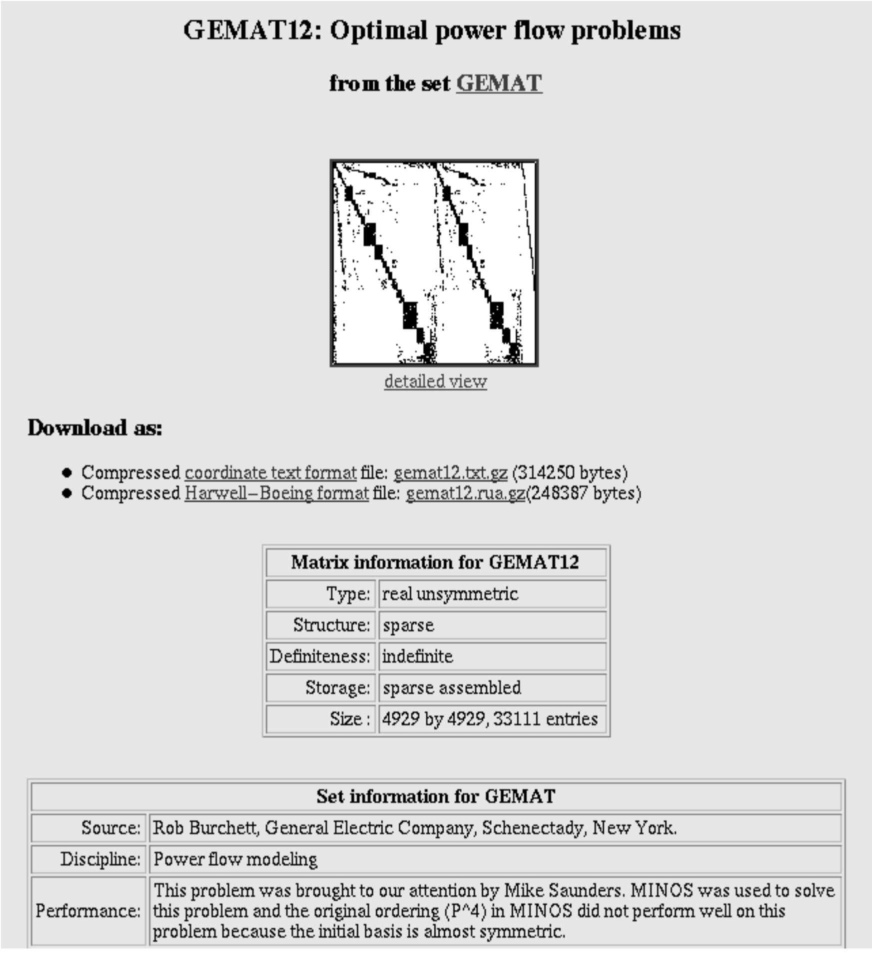


Figure 3 The Matrix Market home page.

ticular matrix, including its type (e.g., real unsymmetric, real symmetric positive definite, complex Hermitian, pattern symmetric), its overall structure (e.g., sparse, banded, block tridiagonal), storage mode (e.g. sparse assembled, sparse elemental) and number of rows, columns and nonzeros. In addition, a GIF image providing a more detailed view of the nonzero structure is provided; clicking on this image retrieves another image at a finer level of detail. The page allows readers to download the matrix as a compressed file in either Harwell-Boeing exchange format or in coordinate text format. An example is provided in Figure 4.

The matrix and set home pages are automatically generated by a Perl script which processes information from the Matrix and Set databases.



GEMAT12: Optimal power flow problems

from the set GEMAT

detailed view

Download as:

- Compressed coordinate text format file: gemat12.txt.gz (314250 bytes)
- Compressed Harwell–Boeing format file: gemat12.rua.gz(248387 bytes)

Matrix information for GEMAT12	
Type:	real unsymmetric
Structure:	sparse
Definiteness:	indefinite
Storage:	sparse assembled
Size :	4929 by 4929, 33111 entries

Set information for GEMAT	
Source:	Rob Burchett, General Electric Company, Schenectady, New York.
Discipline:	Power flow modeling
Performance:	This problem was brought to our attention by Mike Saunders. MINOS was used to solve this problem and the original ordering (P^4) in MINOS did not perform well on this problem because the initial basis is almost symmetric.

Figure 4 A matrix home page.

4.2 Finding Matrices

The main Matrix Market page gives readers several different ways to search for matrices.

1. *By Browsing Through Sets*
2. *By Database Query*
3. *By Application Domain*
4. *By Contributor*

Users who select *Browsing Through Sets* are presented with a list of the set identifiers along with a short sentence or phrase which describes the matrices in the set. Clicking on the set name retrieves the home page for the set. While this organization is convenient for browsing, it does not fulfill the needs of the typical user: to easily locate matrices which have certain properties of interest. To satisfy this need we have developed a WWW-based tool which allows users to query the Matrix and Set databases.

The *Database Query* tool is an HTML form which allows the user to specify a large variety of attributes describing selection criteria for matrices. Widgets are presented which allow selection based on predefined attributes in the database : arithmetic field (real, complex, pattern), symmetry (symmetric, unsymmetric, unsymmetric with symmetric structure, Hermitian, skew symmetric), definiteness (indefinite, positive definite, positive semidefinite, etc.), nonzero structure (banded, sparse, diagonal, block tridiagonal), storage mode (assembled, elemental), and shape (square, more rows than columns, more columns than rows). Text fields allow users to specify the minimum and maximum number of rows, columns and nonzeros. Additional buttons allow the user to require that right-hand sides, exact solutions, or initial vectors be available. Finally, a text field allows one to specify arbitrary patterns to match in the database entries themselves. This allows users to search for words or phrases that appear in narrative descriptions of the matrices, titles and authors of references, etc.

Only matrices which satisfy *all* of the requirements set down in this form are retrieved. The search engine is a C program which uses Unix `egrep` to perform searches in the database files. Although this implementation was initially considered a prototype, it has proven capable of handling the modest load and fairly small database required of a specialized tool such as the Matrix Market. Matches are returned on a WWW page organized by matrix set. Links to the home pages of individual matrices as well as the sets are provided.

A separate form is available for selecting matrices based upon *Application Domain*. Information describing the origins of the matrices in the collection is provided in the database, and hence the free-text search facility in the general search tool could be used for this purpose. However, since uniform terminology is not used in describing the application domains such a scheme is unreliable. To improve upon this, this form presents a set of links, identified by application domain, which result in the execution of predefined queries on the general search engine which are known to succeed. As a result, the format of the returned result is the same as for the general search tool.

Another form enumerates the *Contributors* (i.e., institutions) who provided matrices to the collection. These are organized by the institution type, i.e. industry, government, and academia. This is implemented in the same way as the *Application Domain* search tool.

5 DOWNLOADING MATRICES

The matrix WWW pages provide users with links that effectively download individual matrices to the local browser. Currently all matrix data provided by Matrix Market is compressed using `gzip`. (Typically, WWW browsers uncompress these files after downloading so that they can be displayed in the browser.)

While there are a multitude of matrix formats available (see for example, (Barrett, et al., 1994) only a subset are needed for exchanging sparse matrix data since, once read into memory, transformation between formats is straightforward. We supply only two: coordinate text file format and the Harwell-Boeing exchange format, the former for its simplicity, the latter as the current interchange standard for sparse matrix research.

5.1 Harwell-Boeing Exchange Format

The Harwell-Boeing format is the only well-defined mechanism for text-file exchange of sparse matrix data. Details of the format are provided in (Duff, Grimes and Lewis, 1989).

The format has several advantages. First, it includes a detailed header with a variety of descriptive information, such as matrix labels, type, order, number of nonzeros, etc. It also admits a very broad range of formatting for the numerical entries; this is done by providing Fortran format descriptors within the header that describes the numerical entries. Although this greatly eases the work for most contributors, it means that the text file is quite difficult to parse by those using languages other than Fortran.

To help remedy this situation, we have developed a set of C procedures for doing I/O with Harwell-Boeing files. The current implementation provides routines for gathering matrix information from the file header, reading into pre-allocated or dynamically allocated storage vectors, and writing to files according the Harwell-Boeing specification from storage vectors. A link to these routines is provided on the Matrix Market home page.

5.2 Coordinate Text File Format

The coordinate text format provides a simple and portable method to exchange sparse matrices. Any language or computer system that understands ASCII text can read this file format with a simple read loop. This makes this data accessible not only to users in the Fortran community, but also developers using C, C++, Pascal, or Basic environments.

In this format the first line lists three integers: the number of rows, columns, and nonzeros in the matrix. The nonzero matrix elements are then listed, one per line, by specifying row index i, column index j, and the value $a(i,j)$, in that order. White space is not significant, (i.e. a fixed column format is not used). Experiments show that such files are about 30% larger than corresponding Harwell-Boeing files. Compressed versions typically exhibit similar ratios. Note that there is no implied order for matrix elements. This allows print routines to traverse the sparse matrix in whatever natural order given by the particular storage scheme. A careful specification for this format, along with specializations for various matrix symmetries, is now under development.

6 EXTENSIONS

We are cooperating with Iain Duff, Roger Grimes and John Lewis to provide a WWW-based distribution mechanism for Release II of the Harwell-Boeing Sparse Matrix Collection. Release II is a significant generalization of Release I, and we expect that some of our database mechanisms will need to be expanded to accommodate it. Our plans are to make Matrix Market the primary archive for the collection, with mirrored copy on *netlib* (`http://www.netlib.org/`). We expect that Release II will be in place later this year.

There are a variety of additional facilities which could enhance a service such as Matrix Market. In the remainder of this section we discuss several of these in more detail.

6.1 Additional Numerical Indicators

There are a variety of additional numerical indicators which can be used to characterize sparse matrices. For example: extreme eigenvalues, Frobenius norm, maximum element, percentage of weakly diagonally dominant rows, columns, average number of nonzeros per row, column, number of elements in (shortest, longest) (row, column), upper and lower bandwidth, average bandwidth, bandwidth of closest 95%, 90%, 80% to diagonal, standard deviation of distance of elements to diagonal, number of nonzero diagonals, and percent of symmetric elements. Most of these, and others, are computed by tools such as the INFOFUN routine of SPARSKIT (Saad, 1994). Since such data might only be of limited value in searching we have not included it in our database. However, we do intend to run a standard code such as INFOFUN against all the matrices in the collection, and include such data in the matrix home pages.

6.2 Dense Matrices

Another direction for extending Matrix Market is to include dense matrices in its collection. Since dense matrices have uniform structure, dense matrix generators may, in fact, be most useful for inclusion. Examples are the LINPACK and LAPACK test suites which are already available via *netlib*. Particular static matrices which provide counter-examples to conjectures and folklore might also be of interest.

6.3 Matrix Generators

Many of the matrices found in the Matrix Market were obtained by running application codes which generated them. In many cases these codes are parameterized so that a wide range of matrices could actually be produced. A trivial example might be a code that generates a discrete 3D Laplacian for any set of grid sizes. Such a code could be used to generate matrices with the same general structure, but of any order. In other cases matrices with differing numerical properties could be obtained by varying various parameters, such the Reynolds number in a fluids computation. Such codes are available with the SPARSKIT distribution, for example. A service like Matrix Market could provide access to such codes in a variety of ways.

- **Downloading Source Code.** If freely available, the source code of a matrix generator itself could be simply made available for downloading. This would be useful for embedding the generator within a larger code for which tested sparse linear algebra software over a range of parameters presented by the generator.
- **Remote Execution Service.** If a user is only interested in a few particular instances of the parameter set, then having the ability run the generator remotely would be useful. In that case, an HTML form would be presented to specify the parameters of interest; submitting the form would cause the generator to run at the Matrix Market site, after which the matrix would be delivered to the user. Such a service could provide access to proprietary matrix generators (with author permission of the authors).
- **Executable WWW Content.** A third option is to provide the matrix generator as executable WWW content such as provided by Java (Flanagan, 1996). In this case, the generator is downloaded to the user's WWW browser, where it queries the user for values of its parameters, and then executes on the user's local machine. This is a more scalable approach than providing a remote execution service, but will present some challenges in that there has yet been little experience in developing portable, reliable and efficient floating-point applications in this environment.

6.4 Format Converters

At present Matrix Market only admits two file formats for the transmission of sparse matrices. Converting these formats for use in a particular application might be inconvenient. Java code could also be provided to download a matrix and convert it to a format more natural for digestion into a particular sparse matrix application.

6.5 Automated submission

Finally, we intend that Matrix Market provide a real marketplace for the exchange of matrix test data. For this to happen there must be a convenient way for users to submit their own matrices for possible inclusion into the collection. To do this, we envision a set of HTML forms which allow users to describe their matrices using various selection widgets and text fields, and to provide a URL where the matrix data can be picked up. Submitting the form would construct prototype database entries for the new matrix, and notify the maintainers that a new matrix should be examined for possible inclusion. We believe that it will still be necessary to review contributions to determine if they

- are in the public domain
- represent a significant application area,
- possess inherent interest or novelty of concern to algorithm developers, and
- do not substantially duplicate other matrices in the collection.

ACKNOWLEDGMENTS

This work is supported in part by Defense Advanced Research Projects Agency under contract DAAH04-95-1-0595, administered by the U.S. Army Research Office. We are indebted to Iain Duff and his colleagues at CERFACS and the Rutherford Appleton Laboratories, and Roger Grimes and John Lewis of Boeing Computer Services for making the Harwell-Boeing Sparse Matrix Collection and its User's Guide available to us. This paper is a contribution of the National Institute of Standards and Technology and is not subject to copyright.

REFERENCES

Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, Dongarra, J. J., Eijkhout, V., Pozo, R., Romine, R. and van der Vorst, H (1194) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia.

Duff, I. S., Grimes, R. G. and Lewis, J. G. (1989) Sparse matrix test problems. *ACM Transactions on Mathematical Software*, **15**, 1–14.

Flanagan, D. (1996) *Java in a Nutshell.* O'Reilly & Associates, Sebastopol, CA.

Saad, Y. (1994) SPARSKIT: A basic tool kit for sparse matrix computations (version 2). Available by anonymous ftp from ftp.cs.umn.edu.