Presented at the Workshop on Operational Aspects of Numerical Software Libraries, Argonne, IL, May 7 - 9, 1975

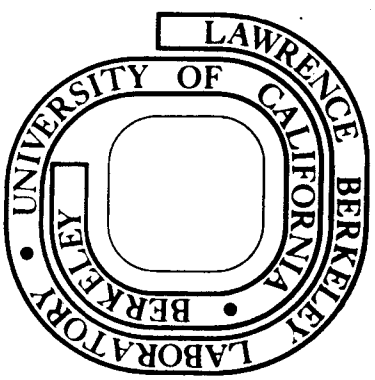
LBL-3841

THE LAWRENCE BERKELEY LABORATORY NUMERICAL SOFTWARE LIBRARY

Elon Close

May 1, 1975

Prepared for the U. S. Energy Research and Development Administration under Contract W-7405-ENG-48



LBL-3841

DISCLAIMER

LBL-3841

THE LAWRENCE BERKELEY LABORATORY NUMERICAL SOFTWARE LIBRARY

BY

Elon Close
Mathematics and Computing Group
Lawrence Berkeley Laboratory
Berkeley, California

May 1, 1975

THE LAWRENCE BERKELEY LABORATORY NUMERICAL SOFTWARE LIBRARY

General Description

The discussion that follows is limited to software that deals primarily with the numerical aspects of computers used at the Lawrence Berkeley Laboratory. That is, I shall not discuss programs or packages that pertain to such fields as graphics, data bases, report generators, payrolls, bit or string manipulators, system routines, executive programs, housekeeping routines; in short, the whole myriad of things that are in any computer library that handles a diverse load of computing activities. I mention this mainly to bring out what we all know; that is, that a lot of what computers do and do best is not in the realm of "numerical". We immediately are restrictive when we limit ourselves to a numerical software library and, in the final analysis, I believe that whatever we do in this restricted area must be compatible with the larger library needs of the general computing facility. Also, the boundaries that separate us from what I might call "the rest of the world" are not all that clear and firm and are, in my opinion, one of our problem areas. So, with the understanding that I am addressing myself to a restricted area of our computer library, I will proceed to try and sketch for you an overview of our current numerical software library.

Formerly we were a special purpose computing center that supplied services primarily for high-energy physics. Most of our computing was done by these physics groups and was in the area of analyzing physics data. It is only recently that we have become a general-purpose computing facility. And not until very recently have we begun a systematic library effort. Consequently, our library is what might be characterized as a mixed bag of subroutines and programs. Some of them are probably reasonably

good, a few might be even the best. Some, maybe even most, are no more than average in any sense of the word. Some are horrible, probably to an extent that we have yet to find out about. This library was developed over the past number of years partly out of a response to specific user needs and partly because of inclusion of routines that people had which they thought should be available as library routines through the computer center library rather than individually from them. To be more specific, the areas that are covered are the standard SHARE arithmetic routines, categories such as evaluation of polynomials, roots of polynomials, etc.

Though this covers a fair spectrum of what might be called numerical software, it does not in itself give any indication of how much or how little we cover any subject; nor does it indicate that we, in any sense, have good routines for the problem at hand. For those of you who might wish to see in more detail the content of our library, I have with me some copies of our library index. This index is probably our most direct way of familiarizing users with what we have in the source library.

Since this is supposed to be a rather general description, I will not spend time on any particular routines. What I should mention is that the library that I have been talking about is what we call our source library and resides on our IBM data cell. This, then, is a directly accessible library in that users simply fetch for themselves the routines they want. When I say "source" I mean principally FORTRAN, there are a few COMPASS routines, but no object decks.

This, however, is not all of our library. And by this I am not referring to system routines, etc. We have groups in our Laboratory other than the Mathematics and Computing Group that develop programs, subroutines, and systems that use the computer. Eventually, these find

their way to our Computer Center Library, or at least we hope that they do. This can happen in different ways. One way is that they can actually ask us, or we them, to include their routines in our data cell library. Another way is that they can leave with our librarian writeups for their routines and use the library newsletter to advertise their wares, so to speak. The acual location of the software may again be on the data cell which is easily accessible, or a tape or IBM chip store, or the left upper shelf of someone's office (the deck with the wide rubber band and blue face card). We, of course, don't recommend the latter case, but in reality it does happen and I would have to admit that I myself am guilty of the practice.

Our Library also consists of what you might call applications programs. For example, we have magnetostatic programs, stress analysis programs, etc. These programs have not traditionally been in our data cell source library, but usually have been in the library in the sense that writeups and people pointers are available through the librarian. I will say more about this class of numerical software below, but simply comment now that the inclusion of these formally in our Library brings with it problems. It is my hope that some of these topics will be discussed in depth later in this conference.

In addition to our standard, or if you will, traditional library, we have begun to acquire and use libraries distributed by others. In particular, we have two packages from The National Activity to Test Software (NATS) Group located at Argonne; they are EISPAC and FUNPACK. The first is, as you well know, a set of routines dealing with eigenvalues and eigenvectors and operations associated with matrices. The second pertains to the evaluation of various special functions and integrals. We also subscribe to The International Mathematical and Statistical

Laboratories (IMSL) Library services that provides codes covering branches of numerical mathematics and statistics. The use of packages of this type is relatively new for us. We are hopeful that this type of service can help our library provide useful routines of known quality and performance to our users. We, ourselves, with our limited staff, cannot hope to provide the extensive software coverage that is necessary in a diverse installation such as ours. Our experience in this is somewhat limited and we hope to learn and to grow in this area.

And last, but certainly not least, we consider that other computer center libraries are an extension of our own. By this, I mean that if we do not have within our own facilities the software that is needed we will do our best to obtain it from some other source. Whether such routines really end up as part of our library has to be decided separately for each such case. Usually they don't and we might consider this to simply be one of our library user services.

Documentation

The area of documentation has been a perplexing one for our Installation. The present manner in which we document our programs is somewhat haphazard and depends to a certain extent on what type of software package we have and where it comes from. I will describe what we have in the way of documentation with the understanding that this is not necessarily what we want to have in the future. It just happens to be the way things have evolved up to the present date.

For our traditional library, that is the source library that resides on our data cell and is mostly subroutines, we have relied on typed sheets that describe the package. These are submitted to the librarian who maintains a file of the program writeups that are then given to users who

request them. Actually the writeup file is accessible to anyone during normal working hours so that people who work at our Installation can for themselves pick up any writeups that they are interested in. There is also an index to available programs and writeups. This is on the data cell and remote users can copy this index to their local printers. However, they would then have to call the librarian and ask that the appropriate writeups be mailed to them.

These writeups usually follow an outline consisting of: Identification, Abstract, Usage, Restrictions, Problem Description, References. They are quite often short -- one page. They are usualy adequate for using the program or subroutine, but quite often are far from adequate if there is a problem or question. For example, a statement in the writeup that a series is used to complete accuracy really leaves one hanging in the air. Subroutines that have this short writeup usally do not have any other type of documentaion except that which exists in the source listing or in the author's notes or in the cited (if any) references. In spite of the obvious limitations and inadequacies, these short writeups have done and continue to do a job, they let the user use the routine, but sort of at his own risk.

For larger, more complex programs and particularly for application type packages, we usually have a user's manual. These can be in the form of our technical internal documents (TID) or often in the form of an LBL report. These reports are usually fairly complete furnishing not only the ncessary information to use the program, but also lots of detail on the method, a good reference list, test cases with sample output, and possibly listings. We furnish no guide lines for such manuals and usually are simply the pointer to the location of the manual and the program. The responsibility lies solely with the authors of the manual and the program.

It is unfortunate that we have so little to do with these, since sometimes we are left out completely and don't even know of their existence until someone asks whether we have such and such a program.

Recently, we have been trying to encourage the use of our data cell for the storage of writeups. A basic editing program, BARB, is available and can be used to produce quite readable text. The advantages of such a move are obvious for the remote user in that they can easily copy out to their installation's line printer exactly what they want and we don't have to be bothered with their requests. In effect, it makes for a directly accessible documentation system. We can easily update the writeups and the user can easily obtain them. For the documentation of our system, this has worked rather well. A lot of effort has been expended in these writeups and I believe that presently we have better, more complete, and up-to-date system documentation than we have ever had. This, however, is not true with regard to our numerical software library. It is, of course, at an early stage and I am sure it will get better. We would like to do more in this area.

Another form of documentation, if you wish to call it that, is the writing of short descriptive guides. Clifford Risk, a Consultant with the Mathematics and Computing Group, has written a short guide to differential equation solvers for us and is now looking at quadrature routines. We feel that guides of this sort fill a need, and we expect to do more in other areas. They not only show the user what is available locally but furnish some background on the numerical schemes and some information on the correct use of such routines and some of the differences and similarities. These also help us focus on what we have in our library, what we would like to have or should have, and which routines

are duplications and possibly should be weeded out. They benefit thus not only the user but the library as well.

We also use our Computer Center Newsletter to help let people know what we have and what we no longer have. As routines are added to or deleted from our Library, notices are run in the Newsletter. Also, we have at various times used the Newsletter to report on testing or correcting of routines. It is rather obvious from the above discussion that our documentation leaves much to be desired. We would expect that as we concentrate more effort in this area a more uniform satisfactory form of documentation would evolve.

Problems and Unique Features

I have the feeling that if I am not careful this could turn out to be the longest part of this talk. I think that one of the fastest ways to summarize our problems and unique features is to refer to Table 1 which shows the system that we have. Basically our problem is that we have to furnish reliable software packages to our computing environment. The uniqueness of any of our problems would stem only from the fact that our computing environment has areas that are different than some of the other installations. I am sure that many of our problems are common and in no way are unique. In the discussion that follows I shall briefly touch on various areas that we might tend to look at as problems.

We have available a fair amount of easily accessible storage space. For example, our data cells, disks, and high access tapes, common files. It would thus seem natural to place our numerical software library and our documentation in this storage space. As I have already noted, we do have a source library on the data cell and are encouraging the placing of writeups on that facility. I have already mentioned the advantage to

our remote users from this approach, and, in fact, this accessibility is an advantage to all our users and our staff. The problem is that the space available is finite and the different media have different degrees of accessibility. It is not obvious how much we should put in any one place. We might feel that all documentation should be on the data cell, this, however, might force some programs to be on the tape. Should subroutines that are used frequently be on a common or permanent file, or on the data cell, or tape? This is further complicated by the fact that our permanent files are not necessarily permanent, nor do our tape files always reside on tape. For example, on the 7600 all tapes are staged to disk and can remain there until automatically purged. In any case, it is obvious that we need some kind of an overall plan to help us allocate our resources effectively.

Another problem that we have is what to do with, or how to handle, large packages from the outside users. In this area we must deal with the problems of documentation, maintenance, and where and how the package will be stored and retrieved. While this problem tends to be somewhat more pronounced in the field of application type programs, it exists even in the library services that we obtain from others such as IMSL, FUNPACK, and EISPACK. Packages such as these are not presently integrated into our Library, but are in themselves separate and the routines that they have must be made available to all our machines and all our users. How they can be made to mesh smoothly with our other library routines or whether they even should be so integrated is an open question.

We are, as are other installations, continually presented with the problem of source versus object or assembly language code and how best to get the routines to interface with not only two machines, but with

different compilers. As you will note from the table we have not one but three FORTRAN compilers and a couple of CDC COMPASS assemblers. As if that were not enough, we have an ALGOL and a PASCAL compiler and also a PL 1. As you might suspect, these latter receive very little support from our library. However, people do use these languages and sometimes take the trouble to get routines for them. It would be nice if we had a way of taking advantage of their work and making it available to their colleagues who use these compilers after they have left. Presently, we do nothing for these users.

One of the areas in which we have an ability we have not used is in monitoring the usage of various routines in our library. If we had histograms for the frequency of use of the various library routines it would help us in deciding what to keep, or how accessible the material should be. I might note that we presently are in the process of going through our data cell source library and we have found routines that won't even compile under our present compilers, so it is pretty obvious that those routines have not been used for some time.

An area that I have yet to mention that causes problems is that of accuracy of the routines. We have machines that in single precision can furnish ansers to 14 or so significant figures. Some of our routines are good to only 7 or 8 figures, and occasionally drop to zero significant figures. The users do not always know of this. In a sense, we have routines that lie, the worst kind. We thus need routines that do what they say they do and say what they do, two very separate functions. Again, with regard to accuracy, not all our users want to pay the price (time) of full precision. In certain number crunching situations where a function is called over and over again the ability to trade time for accuracy may be needed. Whether and to what extent the library services should deal

with this is one of our problems, although not a central one.

To what extent the following areas constitutes problems I am not sure. It is, however, clear that we have yet to deal effectively with the whole question of maintaining, documenting, and furnishing to our computer users a consistently dependable set of routines that meet their numberical software needs. The structure, content, location of the library; the dissemination of information, writeups, corrections to new routines; the communication with other laboratories so that we may learn from them and eliminate unnecessary duplication; the classification of routines in some meaningful manner that at the same time is standard; all these are problems to us in the sense that we have yet to effectively resolve them.

I would like to conclude by saying that our Laboratory is pleased to participate in workshops such as this one being held here at Argonne. It is through this mutual sharing of our experiences and resources that positive progress will be made toward obtaining standard dependable numerical software libraries. There is no question that the need exists.

TABLE 1

Machines	CDC 7600	65,536 SC	513,000 LC
	6600	131,072	
	6400	131,072	
Storage	7600	2 CDC 7638 Disks	Total 160,000,000 words
	6600	2 CDC 6603 Disks	Total 15,000,000 words
		CDC 841	25,000,000
		CDC 844	96,000,000
	Magtape	1000 high access, 40,000 tape library	
	3-IBM	2321 Data Cells	Total 150,000,000 words
	IBM	1360 Photo-Digital Storage	5,000,000,000 words
	8 CDC	659 M T 9 ch.	
	20 CDC	607 M T 7 ch.	
Job Input	4 CDC	405 Card Readers	
	Remote teletypes, Vista Consoles, terminals		
	Remote Batch - University Computing Co. COPE Controller		
Languages	FORTRAN IV	CDC FUN (RUN76)	
		Fortran Extended (FTN)	
		University of Minnesota's MNF	
	COBOL, SNOBOL, ALGOL, PASCAL, LIPS, PL1		
	CDC COMPASS		

F1 BKY LINVER
1 of 1

I. IDENTIFICATION

A. Title: LINVER

B. Author: Loren P. Meissner, x6361

C. Date: 10/15/65 - Revised 1/5/74

D. Machine: 6600/7600

E. Language: Chippewa FORTRAN

II. PURPOSE

Matrix inversion, including ill-conditioned matrices. 1974 version does not destroy A. Uses adjustable dimension.

III. USAGE

To invert the matrix A,

CALL LINVER (A,SCR6,X,N,DET,IEX,CNR,SINGUL,L,SCR1,SCR2,SCR3,SCR4,SCR5)

where

A	is the given N x N matrix.
X	is the required inverse.
N	is a positive integer.
DET and IEX	return the determinant in the form DET*(10**EX).
CNR	is an estimate of the spectral norm of A.
SINGUL	is a logical variable which is set .TRUE. if A is singular.
L	is the column length of A and X in storage -- i.e., it is the value of N1 in the main program statement DIMENSION A(N1,N2),X(N1,N3). Thus L must equal or exceed N.
SCR2,...,SCR6	are scratch areas each containing at least N cells.

IV. RESTRICTIONS

The arrays A, X, and SCR1 occupy a total of 1 * (3 * N) cells, and SCR2,..., SCR6 require N * 5 additional cells. Time: about 1.5 second for N=70 on 7600.

V. METHOD

See LINIT writeup. The identity matrix is generated in SCR6 one column at a time. LINVER leaves LU factored form of A in SCR1.

WARNING: LINVER will not work with FTN or MNF. Use RUN76 only.

TECHNICAL INFORMATION DIVISION
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720