Shopping for mathematical software electronically

Jack Dongarra
Eric Grosse

A quick, easy, and inexpensive way to get public-domain software via electronic mail

Engineers and scientists throughout the world are using collections of high-quality mathematical software at universities and in industry. Typically, this software is obtained from distribution agents—for example, IMSL, the National Energy Software Center (NESC), and the Numerical Algorithms Group (NAG). All these agencies do a fine job with the distribution of large packages of mathematical software. But how does one go about getting a single piece of software? The usual approach, unfortunately, entails sending a tape to an author, waiting until that author copies the program and mails back the tape, and then deciphering the often alien tape format. The whole process can waste an intolerable amount of time.

We've developed a system, called *netlib*, that provides quick, easy, and efficient distribution of public-domain software to the scientific computing community on an as-needed basis.

A user simply sends a request by electronic mail to *netlib@mcs.anl.gov* on a national network such as ARPAnet, and receives the requested software over the network.

Netlib in use

Imagine an engineer who needs to compute several integrals numerically. He or she consults the resident numeric expert, who advises trying the routine *dqag* for some preliminary estimates and then using *gaussq* for the production runs. The engineer types at the terminal

mail research!netlib
send dqag from quadpack
send gaussq from go

In a short time, the engineer receives back two pieces of mail from *netlib*. The first contains the double-precision Fortran subroutine *dqag* and all the routines from *quadpack* that *dqag* calls; the second contains *gaussq* and the routines it calls.

Should the engineer later decide that the routine *dqags* would be more effective, he or she could ask *send dqags but not dqag from quadpack* to get *dqags* and any subroutines not already sent with *dqag*.

This engineer happens to be connected to the uucp network. If, instead, his or her machine was on the ARPAnet, the person would use the address *netlib@mcs.anl.gov*. If the code was needed in upper case, the engineer would send a request in all caps; to get single precision, he or she need simply change the names of the routines or the libraries as appropriate.

As typical examples we give the following (with an explanation in parentheses):

send dgeco from linpack

(Retrieves routine DGECO and all routines it calls from the LINPACK library.)

send only dgeco from linpack

(Retrieves just DGECO and not subsidiary routines.)

send dgeco but not dgefa from linpack

(Retrieves DGECO and subsidiaries, but excludes DGEFA and subsidiaries.)

send list of dgeco from linpack

(Retrieves just the file names rather than the contents; this can be helpful when one already has an entire library and just wants to know what pieces are needed in a particular application.)

find eigenvalue

(Retrieves the names of routines in the collection related to the keyword *eigenvalue*.)

Netlib tolerates minor syntax deviations, since we do get requests like "Please send me rlmach from port. Thank you." Apparently, the system is so easy to use that people don't realize they're talking to a program. However, we make no attempt to accept arbitrary English.

Quick response

Just how quickly requests are answered depends on the speed of the network communications involved. Five or ten minutes is typical for ARPAnet, while uucp may require anywhere from minutes to days.

The actual processing time is insignificant. One user wrote back enthusiastically that the system was so fast he preferred using it to hunting around on his own machine for the library software.

The heart of the collection

Table 1 lists the public-domain software currently included in *netlib*. A particularly valuable aspect of the collection lies in the recent research codes and the "golden oldies" that somehow never made it into standard libraries. Almost all of these programs are in Fortran, though some are in C. There are also descriptions and benchmark data for various computers, a tabulation of polyhedra, test data for linear programming collected by Gay, and the "na-list" electronic address book maintained by G. Golub and Mark Kent.

In addition, we have included a few odds and ends for convenience, such as technical memoranda and reports produced by the Mathematics and Computer Science Division at Argonne.

We do *not* send out entire libraries. A computer center setting up a com-

0278-6648/89/0002-0037$01.00 © 1989 IEEE

prehensive numerical library should get magnetic tapes through the usual channels.

Security and other problems

Might not someone tamper with the program text as it is en route to the user? Yes, that is of course possible, but for the moment we feel that the threat is minimal. *Netlib* has been implemented so as not to open security holes in the host systems.

We have encountered problems occasionally with computers that are willing to send us mail but will not allow us to send mail back. Delays for multihop and internetwork mail are more common.

The most difficult problem we have encountered has been length limitation; a few of the programs are more than 100 kilobytes, and that is more than the mail systems at many ARPAnet sites will tolerate. We did consider Huffman coding to compress the files, but that would save only about a factor of two and require decoding programs. Instead, we split large items into several pieces of mail.

Comparison with other services

The *netlib* service provides its users with several features:

- There are no administrative channels to go through.
- Since no human processes the request, it is possible to get software at any time, even late at night.
- The most up-to-date version is always available.
- Individual routines or pieces of a package can be obtained instead of a whole collection. (One of the problems with receiving a large package of software is the volume of material. Often only a few routines are required from a package, yet the material is distributed as a whole collection and cannot easily be stripped off.)
- It's free!

On the other hand, netlib is simply a clearinghouse for contributed software and therefore subject to various disadvantages that have plagued such projects in the past. The only documents, example programs, and implementation tests are those supplied by the code author or other users. Also, there may be multiple codes for the same task and no help in choosing which is best.

We have made an effort not to stock duplicate copies of machine constants, but in general we have left submitted codes untouched. We have also tried to be more selective than many personal computer bulletin board systems: we do not allow users to put their own software automatically in the collection.

Package	Description
ALLIANT	set of programs collected from Alliant users
APOLLO	set of programs collected from Apollo users
BENCHMARK	various benchmark programs and a summary of timings
BIHAR	Bjorstad's biharmonic solver
BMP	Brent's multiple precision package
CHENEY-KINCAID	programs from the text Numerical Mathematics and Computing
CONFORMAL	Schwarz-Christoffel codes by Trefethen, Bjorstad, Grosse
CORE	machine constants, BLAS
DIERCKX	spline fitting
DOMINO	communication and scheduling of multiple tasks; Univ. Maryland
EISPACK	matrix eigenvalues and vectors
ELEFUNT	Cody and Waite's tests for elementary functions
ERRATA	corrections to numerical books
FISHPACK	separable elliptic PDEs; Swarztrauber and Sweet
FITPACK	Cline's splines under tension
FFTPACK	Swarztrauber's Fourier transforms
FMM	software from the book by Forsythe, Malcolm, and Moler
FN	Fullerton's special functions
GCV	Generalized Cross Validation
GO	"golden oldies" gaussq, zeroin, lowess, . . .
GRAPHICS	ray-tracing
HARWELL	MA28 sparse linear system
HOMPACK	nonlinear equations by homotopy method
ITPACK	iterative linear system solution by Young and Kincaid
LANCZOS	Cullum and Willoughby's Lanczos programs
LASO	Scott's Lanczos program for eigenvalues of sparse matrices
LINPACK	Gaussian elimination, QR, SVD by Dongarra, Bunch, Moler, Stewart
LP	linear programming
MACHINES	short descriptions of various computers
MICROSCOPE	Alfeld and Harris' system for discontinuity checking
MINPACK	nonlinear equations and least squares by More, Garbow, Hillstrom
MISC	everything else
NA-DIGEST	archive of mailings to NA distribution list
NAPACK	numerical algebra programs
ODE	ordinary differential equations
ODEPACK	ordinary differential equations from Hindmarsh
PARANOIA	Kahan's floating-point test
PCHIP	hermite cubics Fritsch+Carlson
PLTMG	Bank's multigrid code; too large for ordinary mail
POLYHEDRA	Hume's database of geometric solids
PORT	the public subset of PORT library
PPPACK	subroutines from de Boor's Practical Guide to Splines
QUADPACK	univariate quadrature by Piessens, de Donker, Kahaner
SEQUENT	set of programs collected from Sequent users
SIAM	typesetting macros for SIAM journal format
SLATEC	machine constants and error handling package from the Slatec library
SPARSPAK	George + Liu, sparse linear algebra core
SPECFUN	transportable special functions
TOEPLITZ	linear systems in Toeplitz or circulant form by Garbow
TOMS	Collected Algorithms of the ACM
VANHUFFEL	total least squares, partial SVD by Van Huffel
Y12M	sparse linear system (Aarhus)

Promoting modern numerical techniques

Netlib cannot replace commercial software firms. We provide no consulting, make no claims for the quality of the software distributed, and do not even guarantee the service will continue (to date, we have received funding for the project from the National Science Foundation).

About the authors

Jack J. Dongarra is Scientific Director of the Advanced Computing Research Facility at Argonne

Eric Grosse is a member of the Technical Staff in the Computing Science Research Center at AT&T Bell Laboratories in Murray Hill, New Jersey.