Netlib and NA-Net: Building A Scientific Computing Community

Jack Dongarra
Gene H. Golub
Eric Grosse
Cleve Moler
More

IEEE Annals of the History of Computing, Volume 30, Number 2, April-June 2008, pp. 30-41 (Article)

Published by IEEE Computer Society

Netlib and NA-Net: Building A Scientific Computing Community

Jack Dongarra
University of Tennessee, ORNL, and University of Manchester

Gene H. Golub
(Deceased 16 November 2007)

Eric Grosse
Google

Cleve Moler
MathWorks

Keith Moore
University of Tennessee

Two resources evolved in the early 1980s to serve the needs of the scientific computing community. These resources were Netlib, a software repository that facilitated distribution of public-domain software, and NA-Net, a file of analysts' contact information that eventually supported an online directory and newsletter digest. The authors who created Netlib and NA-Net explain the history of these resources and their continuing impact.

With today's effortless access to open source software and data via a high-speed Internet and responsive search engines, it is hard to remember just how different the life of computational scientists was in the 1970s and early 1980s. At the time, computing largely took place in a mainframe world, typically supported by one of the few commercial numerical libraries installed by computer center staff. Since this situation was imperfect (the existing software did not solve all of scientific computing's needs), scientists were forced to write or borrow additional software. There were a few exemplary libraries such as EISPACK in circulation, and a larger body of Fortran programs of variable quality. Obtaining them was a bothersome process, however, that involved leveraging personal contacts, government bureaucracies, negotiated legal agreements, and the expensive and unreliable shipping of 9-track magnetic tapes or punch card decks. There had to be a better way. That better way manifested itself as two different, separately developed resources filling different functions but which were both aimed at serving the scientific computing community. These resources, still active today, were Netlib and NA-Net.

Netlib repositories—collections of mathematical software, papers, and databases—were initially established in 1984 by Eric Grosse at Bell Labs and Jack Dongarra at Argonne National Laboratory. Based on a suggestion from Gene Golub to build a repository of research software, Dongarra and Grosse designed the first Netlib repository. Each repository made a collection of high-quality mathematical software available via electronic mail—the Bell Labs server provided access via *uucp* (Unix to Unix CoPy) protocols, while the Argonne server made the collection accessible via IP (Internet Protocol). The Netlib collection quickly spread to mirror servers around the world. As the Internet became ubiquitous, other access methods were added. Although the electronic mail interface is still supported, most access today is via the World Wide Web.

 Published by the IEEE Computer Society 1058-6180/08/$25.00

Prior to Netlib (short for ''network library''), software codes had been shared by interest groups such as SHARE since the mid-1950s, but such groups tended to be limited by membership or personal relationships. By widening the distribution of public-domain numerical software, Netlib gave greater leverage to early open source efforts and helped establish the trend of software distribution via the network. Netlib remains popular today.

In a separate development that was also aimed at the scientific computing community, by the time Netlib was established, an ''na list'' (*na* for ''numerical analysis'') of email addresses had already been maintained for several years by Gene Golub, then chair of the Computer Science Department at Stanford University. By 1983, this list was being used to provide an electronic mail forwarding service to numerical analysis specialists. Mail to *na.lastname@su-score* would be forwarded to the list member with that last name. An email broadcast facility was also provided: mail sent to *na@su-score* would be forwarded to everyone on the list. By February 1987, this broadcast facility had evolved into a moderated email digest, which soon became a weekly electronic newsletter. A ''white pages'' database and a World Wide Web interface were eventually added, resulting in the set of services provided by NA-Net today. The NA-Net remains a widely used and valuable resource for the numerical analysis community. The NA-Digest is one of the oldest electronic periodicals, and its popularity continues to grow.

In this article, which describes the early history of these two resources and the rationale behind the system architectures chosen, we assess what parts worked well and which did not.

Netlib

The concept behind Netlib was to facilitate quick, easy, and efficient on-demand access to useful public-domain computational software of interest to the scientific computing community. The mechanism chosen for distribution of this software was electronic mail. Initially, there were two repositories: one at Bell Labs and the other at Argonne National Laboratory. The Bell Labs server provided access via *uucp* protocols, while the Argonne server made the collection accessible via Internet mail. To request files from either server, a user would send one or more commands as message text to that server's email address. For instance, a message consisting of the text ''send dqag from quadpack'' sent to either *research!netlib* (*uucp*) or *netlib@anl-mcs* (Internet) would result in a reply consisting of the ''dqag'' subroutine from the ''quadpack'' package, along with any additional routines needed to use that subroutine.

At the time Netlib was introduced, public-domain software was chiefly distributed by physical media such as magnetic tapes being sent through the postal service. This was difficult not only because of the time and expense involved with handling physical media, but also because of the lack of any widely used standards for writing magnetic tapes. Each computing platform wrote tapes in its native format; and because of differences in word size, record size, and organization of the data on tape, reading foreign tapes could be quite difficult and labor intensive. Fortran programs were formatted with sequence numbers in columns 73–80 and blank fill, which was painfully inefficient in days of dial-up lines before compression was prevalent.

Of course, the Arpanet had been in existence for several years by that time, and software was commonly shared over the Arpanet's File Transfer Protocol (FTP), but the Arpanet served a limited community, and the Internet protocols adopted by the limited-commercial-use Arpanet in 1981 were only beginning to see wider use. Software could also be uploaded to or downloaded from ''bulletin board systems'' or BBSs, but this could entail significant long-distance telephone charges to peruse distant servers. A picture of the technical environment of the early Internet, *uucp,* Bitnet, and separate systems like AOL with gateway size limits due to dial-up and restart-from-beginning error recovery has been documented elsewhere.[1,2]

By contrast, Arpanet electronic mail systems standardized very early on a common message format to be used across all platforms, and *uucp* mail used a similar format. Bitnet had some different conventions, but adequate gateways existed. Costs of email were absorbed into budget overheads, so experimental use could flourish without formalities. Collectively, these conditions made electronic mail a superior medium for the exchange of source code, at least for small files, and Netlib made effective use of it. Unknown to us at the time, silicon chip designs were being sent off for fabrication by email in a service known as MOSIS (Metal Oxide Semiconductor Implementation Service).[3] Another early informa-

Definition of Terms

AMS—The American Mathematical Society (AMS) is an association of professional mathematicians dedicated to the interests of mathematical research and scholarship through various publications and conferences.

BLAS—Basic Linear Algebra Subprograms (BLAS) are standardized application programming interfaces for subroutines to perform basic linear algebra operations such as vector and matrix multiplication.

EISPACK—A software library for numerical computation of eigenvalues and eigenvectors of matrices, written in Fortran.

FNLIB—Portable special function routines (e.g., Bessel functions, the error function, and so on).

IMSL—The IMSL (International Mathematics and Statistics Library) Numerical Libraries are software libraries of numerical analysis functionality that are implemented in widely used computer programming languages of C, Java, C#.NET, and Fortran. Software developers will typically embed algorithms from these libraries into their software applications, using their preferred programming language. The IMSL Libraries are provided by Visual Numerics Inc.

LINPACK—A software library for performing numerical linear algebra on digital computers.

Macsyma—A computer algebra system that was originally developed from 1968 to 1982 at MIT as part of Project MAC and later marketed commercially. It was the first comprehensive symbolic mathematics system and one of the earliest knowledge-based systems; many of its ideas were later adopted by Mathematica, Maple, and other systems.

Matlab—A numerical computing environment and programming language. Created by The MathWorks, Matlab allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages.

MOSIS—MOSIS (Metal Oxide Semiconductor Implementation Service) is probably the oldest (1981) integrated circuit (IC) foundry service and one of the first Internet services other than supercomputing services and basic infrastructure such as email or FTP.

MINPACK—A library of Fortran subroutines for the solving of systems of nonlinear equations, or the least-squares minimization of the residual of a set of linear or nonlinear equations.

NAG—NAG Numerical Libraries is a software product sold by The Numerical Algorithms Group Ltd (originally the Nottingham Algorithms Group). The product is a software library of numerical analysis routines.

PORT—The PORT Mathematical Subroutine Library from Bell Labs is a collection of Fortran 77 routines that address many traditional areas of mathematical software, including approximation, ordinary and partial differential equations, linear algebra and eigensystems, optimization, quadrature, root finding, special functions, and Fourier transforms, but excluding statistical calculations. PORT stands for Portable, Outstanding, Reliable, and Tested.

QUADPACK—A Fortran subroutine package for the numerical computation of definite one-dimensional integrals.

SHARE—A volunteer-run user group for IBM mainframe computers that was founded in 1955 by Los Angeles-area IBM 701 users. It evolved into a forum for exchanging technical information about programming languages, operating systems, database systems, and user experiences for enterprise users of small, medium, and large-scale IBM computers such as IBM S/360, IBM S/370, zSeries, pSeries, and xSeries.

tion distribution service by email was at CSnet.[4] Clearly, then, by the beginning of the 1980s, a critical mass of email users in the scientific community had arrived, and email-based servers were inevitable.

Criteria collection and contents

Netlib limited its scope to mathematical software and related information of interest to scientific computing. Netlib's developers made an effort to limit the collection to software of demonstrated high quality. The fact that high-quality public-domain software packages such as LINPACK and EISPACK were available for Netlib's initial collection helped to set high standards for future additions to the collection. More generally, the numerical analysis community's tradition of producing robust software to address clearly defined problems made it somewhat easier to establish a high-quality library for scientific computing than for some other areas.

Another way in which Netlib maintained its quality was by having all contributions reviewed by a ''chapter editor'' who provided some assurance of the quality, stability, and novelty of the software, and who resolved authorship disputes. However, Netlib never attempted to do serious testing, as asked of the reviewers for ACM's *Transactions on Mathematical Software* (*TOMS*). Chapter editors typically made a quick judgment call based on publication of the underlying algorithm in a top journal, reputation of the software in the community, and an instinct developed from personal computing experience in the area for what was novel and worthwhile.

The initial Netlib collection contained LINPACK; EISPACK; MINPACK; FNLIB; routines from the book *Computer Methods for Mathematical Computations* by Forsythe, Malcolm, and Moler; and QUADPACK, as well as a collection of ''golden oldies.'' Soon, additional codes were added from *TOMS* as well as some benchmarking codes, a biharmonic solver, multiprecision arithmetic package, BLAS (Basic Linear Algebra Subprograms), and so on. Substantial effort was expended in collecting these, especially by *TOMS,* in a unified form suitable for network distribution. There was never any doubt, however, that the real intellectual effort and credit belonged to the software component authors and not to Netlib. For the period considered in this article (1984 to approximately 1994), essentially all scientific software was written in Fortran or C (as opposed to Macsyma or Excel or Matlab, say). Scientific software remains the focus of Netlib.

The dominant commercial packages at the time were from Harwell Software Libraries, IMSL Numerical Libraries, and NAG Numerical Libraries, with important but smaller roles played by the PORT Mathematical Subroutine Library, and the libraries of Boeing and the Department of Energy. Although these packages were of high quality and had a financial foundation for user support, they were difficult to adapt in the transition that was then under way, from mainframes in computer centers to departmental servers and personal computers. At least as important as the difficulty of adapting commercial packages was user dissatisfaction with the delay in getting new algorithms into the commercial releases. With Netlib, conversely, since the software was downloaded over the network it was available in ''real'' time—there were no delays in ordering and delivering the software. The user's ability to change the software directly when needed is a common theme in justifying open source today.

Another important source of software, outside of Netlib, came from small snippets of code that users would take from Netlib and paste directly into their own programs. The best and most prominent of these was *Numerical Recipes,*[5] which was published after Netlib was well along but which had at least as wide an impact. An enduring role of the software in Netlib, by comparison, is to handle those more difficult numerical problems that are out of reach of small algorithms.

Unlike the commercial libraries and efforts such as *Numerical Recipes,* Netlib has also provided a home for the numerical research community to share a historical collection of many competing algorithms designed to solve the same problem. This can confuse the engineer who wants digested advice on the single best method to use but enhances the research community by making it feasible to run fairer comparisons of new algorithms against old. Extending that role further, Netlib hosts collections of standard data sets or functions for such algorithm comparisons.

With Netlib, since software was downloaded over the network, it was available in ''real'' time—there were no delays in ordering and delivering the software.

Finally, Netlib has material that does not fit the broader pattern, such as the polyhedra database. These materials were added at a time when file distribution was such a burden that authors were desperate for help and seized the Netlib opportunity.

Server locations and mirroring

The Bell Labs Netlib server went online 3 January 1984, and was used for software distribution within Bell Labs during that spring. Both the Bell Labs and Argonne servers were in public use by July of that year.

The Argonne server was physically moved to Oak Ridge National Laboratory (ORNL) in October 1989, when Jack Dongarra moved to the University of Tennessee. As the Netlib collection became more popular, it became advantageous to mirror the collection outside of North America; however, the number of official mirrors was deliberately kept small while we learned how to minimize the differences between the servers. A European mirror was established in Oslo, Norway, in December 1989, and an Australian mirror also existed by 1989. In May 1993, the original Sequent server that had been in service at Argonne was replaced by two SparcStation 2 machines—one at ORNL and the other at the University of Tennessee in Knoxville (UTK).

These servers were closely synchronized to one another, and both of the servers were configured to accept email requests sent to the *Netlib@ornl.gov* domain.

The ORNL server was retired in 1997. The official servers today consist of *netlib.bell-labs.com* in Murray Hill, New Jersey (which has moved to *netlib.sandia.gov*); *Netlib.org*, at the University of Tennessee; the UK Mirror Service in Kent, England; and *Netlib.no* in Bergen, Norway. These servers are synchronized to one another via a process described elsewhere.[6] Since the introduction of anonymous FTP access to the primary Netlib servers, many more unofficial mirrors have been created; there is a great deal of variation among these mirrors with regard to the organization and currency of their collections.

What is at least as important as the network performance benefit from mirroring is the cultural aspect, namely in the critical role played by the chapter editors. Since this is a volunteer effort by busy people working at the top of their field, it's imperative that Netlib make as few demands on their life as possible. Its principal contribution, therefore, is in allowing them to make their changes directly in their own file trees, and mirror the changes to the other Netlib sites. Setting up a large, reasonably coherent repository that is genuinely shared across multiple organizations in a peer fashion has been, as far as we know, another Netlib innovation.

Netlib features

At the time Netlib was created there was no ubiquitous Internet, but rather an admixture of incompatible networks, each with different hosts, services, and means of addressing. In many cases the networks were interconnected with mail gateways, but these could be difficult to use (because of the need to explicitly route a message through the gateways) and unreliable besides. Because text-based electronic mail was the only service common to all of those networks, this was the means Netlib initially utilized to provide access to items in its collection, but as Internet protocols began to be more widely deployed, other, easier-to-use methods became feasible. A server for the Internet's FTP was provided on the Bell Labs Netlib server in August 1991 and soon afterward at the ORNL server. The xNetlib browser was first deployed in December 1991, allowing users to peruse and download items from the ORNL server via a graphical user interface for the X Window System.[7]

In response to the growing popularity of the World Wide Web, HTTP Netlib servers were established in October 1993 at UTK and ORNL; Gopher was enabled in November 1993. A Web server interface was implemented at Bell Labs one week after Mosaic arrived there; from our perspective it was the cross-platform convenience of that browser, much more than HTML or HTTP, that was revolutionary. Although the email interface is still supported, HTTP—and to a lesser extent, FTP—quickly eclipsed all other sources of Netlib traffic. Netlib dabbled in software-as-a-service by providing email-delivered translation of Fortran into C, but sister systems such as Network-Enabled Optimization System (NEOS) were much more influential in that direction for the numerical community.

One of Netlib's unsuccessful features was its use of digital signatures. Since large software is downloaded and compiled without inspection and may be distributed via mirror sites of unknown security, we thought people would appreciate having PGP-signed MD5 checksums. We also built mechanisms that would allow authors to update their own files, subject to proper signatures. Neither feature got much use, presumably because we did not make them sufficiently automatic and because there have been no known instances of security abuse involving Netlib.

Another security-related feature was popular briefly but faded. Before the days of personal homepages, it could be inconvenient to locate someone's contact information. The NA-Net database went far to solve this for the numerical analysis community, but was never as complete as journal editors would like when, for example, looking for referees. For its part, Netlib introduced a way to search the SIAM (Society of Industrial and Applied Mathematics) membership list. Since this was during an era when network connectivity was unreliable, the method Netlib adopted was to distribute an encrypted version of the database along with a program to allowed restricted searches to protect privacy and deter spam.[8] Eventually this was phased out in favor of the AMS (American Mathematical Society) Combined Membership List online.

Two innovative features designed for Netlib worked well but were eventually casualties of being tied to a research operating system that no longer hosts Netlib. The first of these was a way of asking for a bundle, specifically in the Unix *tar* format, for a minimal set of files with no unresolved dependencies. The second was a ''historic Netlib'' server that was a kind of time

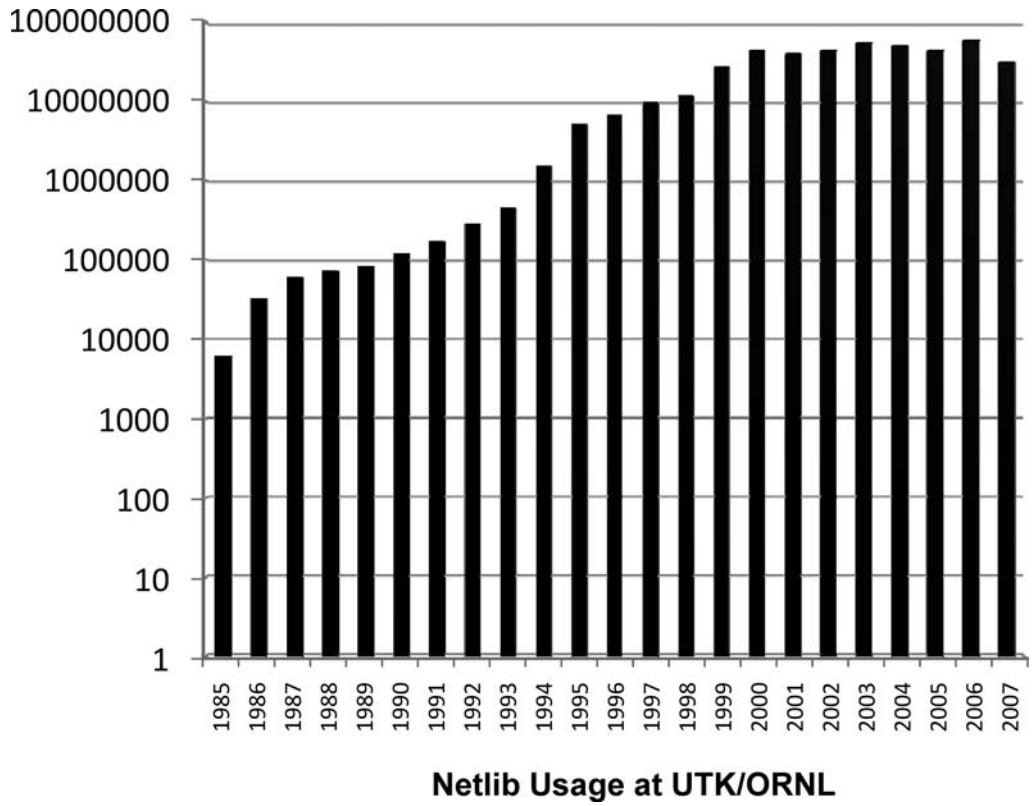

Figure 1. Netlib usage at the University of Tennessee–Knoxville and Oak Ridge National Laboratory.

machine allowing one to see all the changes in a program, much as version control systems do. It was pleasant to be able to provide such features without introducing any new access protocols, using the power of the Plan 9 operating system's generalized vision of a file.[9]

Usage patterns

At least at the UTK and ORNL servers, Netlib usage rose steadily for 1985–2000 (see Figure 1). Traffic levels have varied in recent years, possibly in part because of an increased use of crawlers and mirrors. Approximately 80 percent of the current traffic appears to be from individual users, 7 percent from mirrors, and 12 percent from Web crawlers, which are presumably building indices for use by search engines.

Community impact

A lot of mathematical software is available, either commercial or free. However, not all that software is of high quality. It can also be difficult to locate the appropriate software by using web search engines, since the descriptions available for searching may be lacking or may not match the vocabulary used by the searcher. A good solution to these problems is to have experts in the field of numerical analysis—as is the case with Netlib—maintain a moderated collection of high-quality software that is organized and catalogued with appropriate metadata to enable easy searching.

Netlib was not the first collection of such software, however; a few of the best-funded and well-run laboratories had local libraries tended by numerical consultants who grew to know the local needs especially well. As one relevant example, the Stanford Linear Accelerator Center collaborated with Stanford University's Computer Science Department to train graduate students in numerical analysis while contributing to state-of-the-art numerical processing for physics calculations. Students from that program went on to be active contributors to Netlib.

The main impact of Netlib was to democratize and reduce the duplication of libraries such as Stanford's by making them available worldwide, promoting the best practice of reuse to a broader audience. A secondary unintended impact may have been to weaken the case for employing local consultants. We have no data on the extent of this effect, but

would argue an experienced consultant is vastly better than any catalog. Whatever harm may have been done on this score has been more than balanced by the increased professional credit earned by authors of widely used codes. It is not uncommon in grant proposals for authors to cite Netlib download statistics as evidence of their reputation and practical impact in the field.

The experience gained from Netlib has been transferred to other projects designed to promote software collections management and software reuse. The National HPCC Software Exchange (NHSE) strove to apply the techniques and technologies of Netlib to more loosely coupled software repositories. The NHSE also made software repository creation and maintenance easier through the Repository In a Box (RIB) toolkit.

Netlib pioneered network delivery of software during a period of explosive growth in networking, requiring the tracking of new delivery technologies as mentioned above. Additionally, Netlib has shown that it is possible to manage such collections in a distributed manner, both in terms of infrastructure and administration. The Netlib collection itself has stopped growing as rapidly as in its first decade, as algorithm writers found it increasingly easy to publish their software directly. However, there is a definite risk that individual sites will eventually disappear, with the contents lost to mankind. There is no guarantee that Netlib will exist forever, but the distributed administration has already survived several changes in employers that would have spelled disaster for a smaller collection.

Related projects

Netlib has had a number of spin-off efforts, summarized here.

- *XNetlib* began in 1990, predating the Web and tools like Netscape. XNetlib was a tool for ''Web based'' software distribution. Whereas Netlib originally used e-mail as the user interface to the collection of public-domain mathematical software, XNetlib used its own specialized X-Window interface and Unix socket-based communication.
- *NHSE.* For 1994–2004, the NHSE (National HPCC [High-Performance Computing Council] Software Exchange) existed as a distributed collection of software, documents, data, and information of interest to the high-performance and parallel-computing community. The significance of the collaborative effort is evident through the many useful reports and tools generated as well as the many repositories that have been (and still are being) created, with the Repository-In-a-Box (RIB) toolkit developed in 1996. However, continued operation of the site without funding has become impractical.
- *RIB* provides a toolkit for building and maintaining metadata repositories. RIB was developed by the NHSE Technical Team at the University of Tennessee, Knoxville. Initially, RIB only provided tools for the creation of software repositories. The recent release of RIB v2.0 allows RIB to create general metadata repositories. The creation of software metadata repositories remains the primary application of RIB.[10]
- *NetBuild* is a tool that automates the process of selecting, locating, downloading, configuring, and installing computational software libraries from over the Internet. Additional tools aid in the construction and cataloging of libraries in the format used by NetBuild.[11]

One of the primary benefits of NA-Net's forwarding facility was that it provided all of its subscribers with a uniform address. Before the Internet, email traveled a hodgepodge of dissimilar networks.

NA-Net

The Numerical Analysis Net (NA-Net) is a collection of several services designed to foster information and a sense of community among numerical analysts. It currently consists of an email forwarding service, the NA-Digest, and a white-pages database.

Software and hosting

NA-Net traces its origins to a list of email addresses maintained by Gene Golub at Stanford University and distributed to others within the numerical analysis community. At some point an electronic mail system was specially configured so that mail to *na.lastname@su-score* (later, *score.stanford.edu*) would be forwarded to the person on the list with that last name. This was originally implemented by manually copying entries from Golub's list into a system alias file (requiring interaction by the system administrator each time the list was updated). Eventually software was written by Mark Kent and Ray Tuminaro to allow the forwarding aliases and address list (for human perusal) to be generated from a common database.[12]

At an early stage, Golub collected addresses of numerical analysts. Jim Wilkinson (a leader in the field of numerical linear algebra) once asked, "Whom can I contact on the Net?" and Golub passed along his file of analyst contacts. It then occurred to Golub that we ought to have universal addresses. So at Stanford a system was set up where a user could address a person as *na.<lastname>@score*. We felt that the NA-Net would enable our community to communicate more easily with one another, especially when Net addresses were—at the time—so complicated. (Some users commented that they could only communicate via NA-Net!)

One of the primary benefits of NA-Net's forwarding facility in those days was that it provided all of its subscribers with a uniform address. Before Internet access became ubiquitous, email traveled over a hodgepodge of dissimilar networks, each with its own addressing scheme. When sending a message between dissimilar networks, it was often necessary to "source route" the message through a gateway by embedding the recipient's address inside the address of the gateway. Because each network had its own addressing convention, an address that worked to reach a recipient from one location of the network would not necessarily work from another. But when sending mail to any NA-Net subscriber it was only needed to understand how to send mail to one *na.lastname* address; the same pattern would work for any other NA-Net subscriber. This eliminated some of the guesswork of mailing between networks, at least when mailing to other NA-Net subscribers.

A broadcast facility was also set up so that a message sent to *na@score.stanford.edu* would be forwarded to everyone on the list. Eventually traffic volume and accidental misuse became significant enough that some sort of moderation was required for the broadcast facility, and so it was converted to an email "digest." A moderator would review messages sent to *na@score.stanford.edu*, and the selected messages would then be sent out to everyone. The first issue of the digest was 13 February 1987. At first the digest was sent out at irregular intervals but soon settled into a weekly publication.

In December 1990, the NA-Net was moved to Oak Ridge National Laboratory, using new software written by Bill Rosener. The new software preserved the *na.lastname* forwarding and digest functions, but also allowed individuals to add themselves to the NA-Net, remove themselves, or change addresses, unlike previous versions that required manual maintenance of the subscriber list.

To subscribe, a user sent an email message to *na.join@na-net.ornl.gov* with the following fields in the message body:

Firstname: user's first name
Lastname: user's last name
E-mail: user's email address

The NA-Net server would then reply with a message indicating whether the user was successfully added. To unsubscribe, a user would send a similar message to the address *na.remove@na-net.ornl.gov*; to change an address the message was sent to *na.change@na-net.ornl.gov*. Each function had its own email address and its own requirements for the format of the data to be supplied.[13]

In May 1991, a "white pages" facility was added to NA-Net. Members could store information about themselves in the white pages database, such as their interests and home and work addresses. This information would then be made available in response to queries sent by email to *na.whois@na-net.ornl.gov*.

By June 1993, the service had become so popular that the server was having difficulty handling the email traffic. At this time the entire NA-Net software package was entirely rewritten by Keith Moore to improve scalability (especially of email distribution) and robustness, but the user interface remained the same as before. This software remains in use today, with only minor changes. In November 1994, a web interface was added to rid users of the burden of having to submit requests in text-based email with rigidly defined syntax.

NA-Digest history and content

In the early days the NA-Digest came about because Gene Golub's secretary was sending

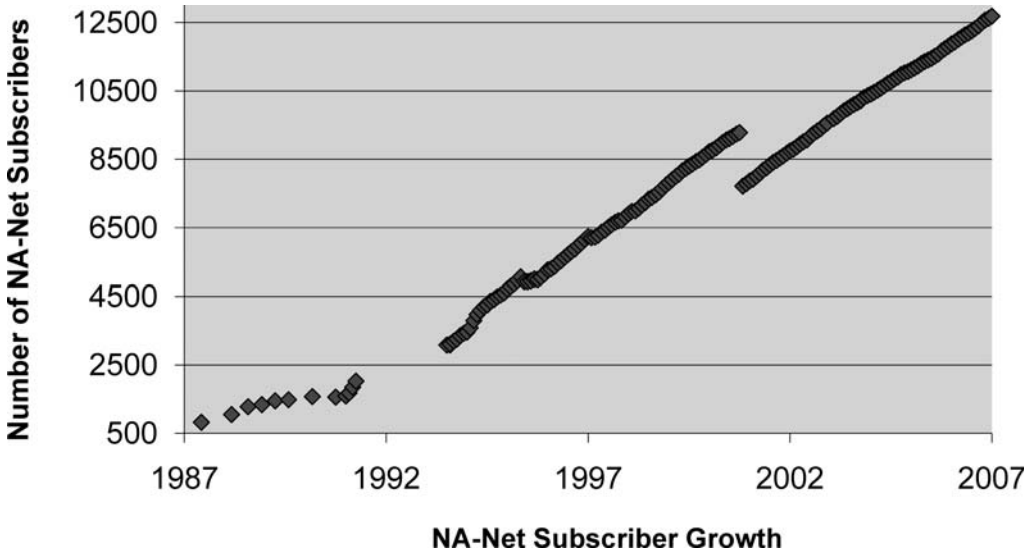

Figure 2. NA-Net subscriber growth.

the digest to the whole NA-Net address list. Gene was the original editor of the NA-Digest until July 1987, when he began a sabbatical leave from Stanford, and at that time Cleve Moler of MathWorks began editing the digest. With only occasional absences, Cleve continued to edit the digest until September 2005. Currently, the digest is edited by Tamara Kolda of Sandia National Labs.

As the name suggests, the intent of the NA-Digest has been to have short announcements summarizing more extensive material available elsewhere. Today, almost all of the digest contributions have URLs pointing to more complete announcements available on the Web.

The digest has generally contained anything of interest to the numerical analysis and mathematical software community. This has included both technical discussions and information about community members. Examples of material commonly appearing in the digest include announcements of conferences, workshops, software releases, change of addresses, and new books; advertisements of jobs; and notices about community members: awards, significant achievements, and deaths.

Reasonably complete archives of the digest exist, hosted at http://www.Netlib.org/na-digest-html/. The archives contain a great deal of material of interest to anyone studying the history of numerical analysis software.

Usage patterns

Based on early reports in the NA-Digest and on log files maintained since 1993, the number of NA-Net subscribers has increased steadily from 821 subscribers in May 1997 to 11,295 subscribers today (see Figure 2). (We are missing some data, as the result of lost log files, in the interval 1991–1993. The discontinuity in 2000 was caused by removal of many addresses that were no longer reachable, such as Bitnet addresses.)

Several thousand messages per day are forwarded through NA-Net's email forwarding service. From a perusal of log file entries, many of these unfortunately appear to be spam. For many years the NA-Digest subscriber list was made available for download over Netlib, a holdover from the days when the network was small and most people with network access could be assumed to act reasonably. The list was also available to anyone who sent mail to *na.sendlist@na-net.ornl.gov*. Even after the list was removed from Netlib and the sendlist address was disabled, however, subscriber addresses were occasionally found to have been ''leaked'' by various means.

Community impact

The international numerical analysis community is small enough that it still has a cohesive, ''family'' feeling. We believe the NA digest has helped maintain that feeling; moreover, the fact that the digest is still distributed via low-tech, simple text email means that it is accessible to anyone with access to a computer and a network connection. This is particularly important to subscribers around the world who are unable to travel to many meetings. Cleve Moler reports that he has several times met people who recognize

Many of us who want to hasten growth in computational science believe that to do so requires much more community focus and funding on its vital core: software and the mathematical models and algorithms it encodes.

his name primarily as the ''guy who sends me email every week.''

We are indebted for the feedback we received of a referee's comment (made anonymously for a paper's review in July 2007) for observing that

> NA-Net's existing context was probably SIGNUM. This ACM Special Interest Group for Numerical Computation built a community around the quarterly SIGNUM Newsletter, which was mailed to members. (Within the ACM, some SIGs are held together by strong conferences that define them, while others are held together by the exchange of information in a newsletter. SIGNUM never had a strong recurring conference—SIAM served that role better—so it was primarily a ''newsletter SIG.'') NA Digest provided a much better forum for this type of information exchange, delivering it more quickly, eventually reaching a much wider community, and providing other valuable services. SIGNUM folded around the year 2000 after being in existence for more than 25 years. The success of NA-Net was probably a strong reason for its demise.

Lessons for funding models

One reason to consider the history of successful projects such as Netlib and NA-Net is to try to generalize and see what decision makers today might learn for encouraging other successes. In this section, we speculate on the critical issue of motivation and support.

The idea that computational modeling and simulation represents a new branch of scientific methodology, alongside theory and experimentation, was introduced about three decades ago. It has since come to symbolize the enthusiasm and sense of importance that people in our community feel for the work they are doing. But when we try to assess how much progress we have made and where things stand along the developmental path for this new ''third pillar of science,'' recalling some history about the development of the other pillars can help keep things in perspective. It seems clear that while computational science has had many remarkable youthful successes, it is still at a very early stage of growth.

Many of us today who want to hasten that growth believe that the most progressive steps in that direction require much more community focus and funding on the vital core of computational science: software and the mathematical models and algorithms it encodes. But when it comes to advancing the cause of computational modeling and simulation as a new part of the scientific method, there is no doubt that the complex software ''ecosystem'' it requires must take center stage.

At the application level, the science must be captured in mathematical models, which in turn are expressed algorithmically and ultimately encoded as software. Accordingly, on typical scientific projects, the majority of funding supports this translation process, beginning with scientific ideas and ending with executable software, and which over its course requires intimate collaboration among domain scientists, computer scientists, and applied mathematicians. This process also relies on a large infrastructure of mathematical libraries, protocols, and system software built up over many years and which must be maintained, ported, and enhanced for many more years to come if the value of the application codes that depend on it is to be preserved and extended. The software that encapsulates all this time, energy, and thought, routinely outlasts (usually by years, sometimes by decades) the hardware it was originally designed to run on, as well as the individuals who designed and developed it.

The life of computational science, therefore, revolves around a multifaceted software ecosystem. In the early days, Netlib and NA-Net supplemented commercial libraries, conferences, and journals in building such an ecosystem. But today there is (and should be) a real concern that this ecosystem of computational science, with all its complexities, is not ready for the major challenges that will soon confront the field. Domain scientists now want to create much larger, multidimen-

sional applications in which a variety of previously independent models are coupled together, or even fully integrated. They hope to be able to run these applications on petascale systems with tens of thousands of processors, to extract a good fraction of the performance these platforms can deliver, to recover automatically from the processor failures that regularly occur at this scale, and to do all this without sacrificing good programmability. This vision of what computational science wants to become contains numerous unsolved and exciting problems for the software research community. Unfortunately, it also highlights aspects of the current software environment that are either immature, underfunded, or both.

The efforts of Netlib and the NA-Net's NA-Digest, for the most part, have been through volunteers. These services have been viewed by the community as important and successful, which by itself has been motivation enough for us to devote substantial time to the effort, and enough goodwill for our employers to allow that effort. Advancing to the next stage of growth for computational simulation and modeling will require the solution of hard, basic research problems in computer science and applied mathematics as well as creating and promulgating a new paradigm for the development of scientific software. So the obvious questions arise: How should this kind of activity be supported? Where should it be done? universities? companies like MathWorks? We've been told many times by the funding agencies that ''it isn't research.'' These are difficult, but important, questions going forward. With less industrial research support for broad ecosystems efforts than has been available in past decades, we believe progress will require a greater level of sustained funding from governmental sources.

Summary

Software distributed by Netlib comes with the disclaimer that ''anything free comes with no guarantee.'' In contrast to commercial vendors like the Numerical Algorithm Group (NAG) and IMSL Numerical Libraries, Netlib offers no support beyond whatever documentation contributing authors choose to provide with their code. On the other hand, Netlib provides free, easy access to a large body of high-quality code, and the phenomenal growth of Netlib attests to the value of this service. We hope that Netlib, by making high-quality code even more accessible, has encouraged software developers to make their source codes freely available and will continue to make good programming still easier for the scientific computing community.

Both the NA-Digest and NA-Net have provided a singular resource for the advancement of science and education by helping to create a community—and, at the same time, a *sense* of community—of those interested in scientific computation. It has allowed for cross-pollination of ideas and techniques among scientists, engineers, and numerical analysts, both from academia and industry.

Acknowledgments

The authors thank Don Fike for assistance with Netlib statistics, and Mark Crispin and Mark Kent for information about early NA-Net implementation. Mel Ciment, who was at the National Science Foundation, provided seed funding for Netlib and NA-Net when they first started. Sandy Fraser gave crucial executive support at Bell Labs, not only in budget but by accepting the legal uncertainties of network software distribution in the early days.

References and notes

1. D. Frey and R. Adams, *!%@ A Directory of Electronic Mail Addressing and Networks,* 3rd ed., O'Reilly & Associates, 1993.
2. E. Krol, *The Whole Internet User's Guide & Catalog,* 2nd ed., O'Reilly & Associates, 1994.
3. C. Tomovich, ''MOSIS-A Gateway to Silicon,'' *IEEE Circuits and Devices Magazine*, vol. 4, no. 2, 1988, pp. 22-23.
4. C. Partridge, C. Mooers, and M. Laubach, ''The CSNET Information Server: Automatic Document Distribution Using Electronic Mail,'' *SIGCOMM Comput. Commun. Rev.*, vol. 17, no. 4, 1987, pp. 3-10.
5. W.H. Press et al., *Numerical Recipes in C: The Art of Scientific Computing,* 2nd ed., Cambridge Univ. Press, 1992.
6. E. Grosse, ''Repository Mirroring,'' *Transactions on Mathematical Software*, vol. 21, no. 1, 1995, pp. 89-97.
7. J. Dongarra, T. Rowan, and R. Wade, ''Software Distribution Using XNETLIB,'' *Transactions on Mathematical Software*, vol. 21, no. 1, 1995, pp. 79-88.
8. J. Feigenbaum, E. Grosse, and J.A. Reeds, ''Cryptographic Protection of Membership Lists,'' *Newsletter of the International Association for Cryptologic Research*, vol. 9, no. 1, 1992, pp. 16-20; ftp://cm.bell-labs.com/cm/cs/doc/91/4-12.ps.gz.
9. R. Pike et al., ''Plan 9 from Bell Labs,'' *Proc. Summer 1990 UKUUG Conf.*, London, July 1990, pp. i-9.
10. S. Browne, P. McMahan, and S. Wells, *Repository In a Box Toolkit for Software and Resource Sharing,*

tech. report UT-CS-99-424, Dept. of Computer Science, Univ. of Tennessee, Knoxville, 1999; http://icl.cs.utk.edu/publications/tech_reports/1999/ut-cs-99-424.ps.
11. K. Moore and J. Dongarra, *NetBuild (version 0.02)*, tech. report UT-CS-01-461, Dept. of Computer Science, Univ. of Tennessee, Knoxville, 2001; http://www.cs.utk.edu/~library/TechReports/2001/ut-cs-01-461.ps.Z.
12. M. Kent, *The Numerical Analysis Net (NA-NET)*, Tech. Report 85, Institut für Informatik, Eidgenössische Technische Hochschule Zürich, Jan. 1988.
13. J. Dongarra and B. Rosener, *NA-Net / Numerical Analysis Net*, Tech. Report ORNL/TM-11986, Oak Ridge Nat'l Laboratories, Dec. 1991.

Jack Dongarra is a University Distinguished Professor of Computer Science at the University of Tennessee and holds the title of Distinguished Research Staff in the Computer Science and Mathematics Division at Oak Ridge National Laboratory. He is also an adjunct professor in the Computer Science Department at Rice University, and Turing Fellow at the University of Manchester. He is a Fellow of the AAAS, the ACM, and the IEEE and a member of the National Academy of Engineering. Dongarra has an MS in computer science from the Illinois Institute of Technology and a PhD in applied mathematics from the University of New Mexico.

Gene H. Golub was born in Chicago in 1932 and died 16 November 2007 in Stanford, California. The Fletcher Jones Professor of Computer Science at Stanford University, Golub was a member of the National Academy of Engineering, the National Academy of Sciences, and the American Academy of Arts and Sciences, and an AAAS fellow. His honors include the B. Bolzano Gold Medal for Merits in the Field of Mathematical Sciences. One of his best-known books is *Matrix Computations* (Johns Hopkins Univ. Press, 3rd ed., 1996) coauthored with Charles F. Van Loan. He received a BS, an MA, and a PhD (all in mathematics) from the University of Illinois at Urbana-Champaign.

Eric Grosse is an engineering director for security at Google. Earlier, he was at Alcatel-Lucent's Bell Laboratories working on products and research in network security, systems, algorithms for numerical approximation, and visualization and scientific computing software. Grosse received a PhD in computer science from Stanford University. He is a member of the IEEE, the SIAM, and the ACM.

Cleve Moler is chairman and chief scientist at the MathWorks. He has been a professor at the University of Michigan, Stanford University, University of New Mexico, and the University of California, Santa Barbara. He is the original author of Matlab, and of three textbooks on numerical methods. Moler received a BS from the California Institute of Technology, and an MS and a PhD from Stanford, all in mathematics.

Keith Moore was a research associate in the Computer Science Department, University of Tennessee, from 1991 until 2007. He has participated in several Internet Engineering Task Force standardization working groups since 1990, which produced the MIME format for electronic mail messages; extensions to the SMTP protocol for negotiation of the message size and for delivery reporting options; standard formats for reporting electronic mail delivery successes, failures, delays, and receipt notifications; definition and resolution mechanisms for Uniform Resource Names; and transition mechanisms for Internet Protocol version 6. Moore received a BS in electrical engineering at Tennessee Technological University and an MS in computer science at the University of Tennessee.

Readers may contact Jack Dongarra about this article at dongarra@cs.utk.edu.