Netlib and NA-Net: building a scientific computing community

Jack Dongarra, Gene Golub, Eric Grosse, Cleve Moler, Keith Moore

Abstract

The Netlib software repository was created in 1984 to facilitate quick distribution of public domain software routines for use in scientific computation. The Numerical Analysis Net (or "NA Net") had its roots in the same period, beginning as a simple file of contact information for numerical analysts and evolving into an email forwarding service for that community. It soon evolved to support a regular electronic mail newsletter, and eventually an online directory service. Both of these services are still in operation and enjoy wide use today. While they are and always have been distinct services, Netlib and NA-Net's histories are intertwined. This document gives the system authors' perspective on how and why Netlib and NA-Net came to exist and impact on their user communities.

Keywords: Mathematical Software, scientific computation, public domain,

1. Introduction

With today's effortless access to open source software and data via a high-speed Internet and good search engines, it is hard to remember how different the life of computational scientists was in the seventies and early eighties. Computing was mostly done in a mainframe world, typically supported by one of the few commercial numerical libraries installed by computer center staff. Since this left large gaps, scientists wrote or borrowed additional software. There were a few exemplary libraries such as EISPACK in circulation, and a larger body of Fortran programs of variable quality. Getting them was a bothersome process involving 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.

Netlib repositories were initially opened by Eric Grosse at Bell Labs and Jack Dongarra at Argonne National Laboratory. 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, and though the electronic mail interface is still supported, most access today is via the World Wide Web. There had been sharing of codes by interest groups such as SHARE since the mid-fifties, but 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, helped establish the trend of distribution of software via the network, and remains popular today.

By the time Netlib was established, an "na list" (na for "numerical analysis") of email addresses of numerical analysis specialists had 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. 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 continues to grow steadily in popularity.

This paper describes the early history of these two systems, the motivation of the architectures chosen, and assesses what parts worked well and what did not. We also comment briefly on other ways the world might have evolved.

2. Netlib

The goals of Netlib were 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. There were initially 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, much distribution of public domain software was accomplished by sending physical media such as magnetic tapes through the postal service. This was difficult not only because of the time and expense of 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. The ARPAnet had been in existence for several years and software was commonly shared over the ARPAnet's FTP protocol, but the ARPAnet served a limited community and the Internet protocols which were 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. For a picture of the technical environment of the early Internet, uucp, BITNET, and separate systems like AOL with gateway size limits due to dialup and restart-from-beginning error recovery, see [8] and [9].

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. This all 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 [13]. Another early information distribution service by email was at CSNET [10]. Clearly, by the beginning of the eighties a critical mass of email users in the scientific community had arrived and email-based servers were inevitable.

2.1 Collection Criteria and Contents

Netlib limited its scope to mathematical software and related information of interest to scientific computing. There was also 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 in 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, novelty of the software, as well as resolving 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 judgement call based on publication of the underlying algorithm in a top journal, reputation of the software in the community, and an instinct based on personal computing experience in the area for what was novel and worthwhile.

The initial collection contained LINPACK, EISPACK, MINPACK, FNLIB, routines from the book *Computer Methods for Mathematical Computations* by Forsythe, Malcolm, and Moler, 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, etc. There was substantial effort in collecting these, especially TOMS, in a unified form suitable for network distribution. But there was never any doubt that the real intellectual effort and credit belonged to the software component authors and not to netlib. For the period considered in this paper, essentially all scientific computing was done by writing Fortran or C (as opposed to Macsyma or Excel or Matlab, say) and that remains the focus of Netlib.

The dominant commercial packages at the time were Harwell, IMSL, and NAG, with important but smaller roles played by PORT, Boeing, and Department of Energy libraries. Although these were of high quality and had a financial foundation for user support, they did not adapt well to the switch that was underway from mainframes in computer centers to departmental servers and PCs. At least as importantly, there was user dissatisfaction with the delay in getting new algorithms into the commercial releases.

The ability of the user to change the software directly when needed is a common theme in justifying open source today, but was even more important at a time when commercial software companies made a different tradeoff compared to today regarding quality assurance versus rapid release.

Another important source of software came from small snippets of code that users would paste directly into their own programs. The best and most prominent of these was Numerical Recipes [11], which was published after netlib was well along but 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 also has had a role in some areas of numerical analysis by providing a home for the numerical research community to share a historical collection of many competing algorithms for 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.

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

2.2 Server locations and mirroring

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

The Argonne server was physically moved to Oak Ridge National Laboratory (ORNL) in October of 1989, when Jack Dongarra moved to 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 in December of 1989, and an Australian mirror also existed by 1989. In May of 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. These servers were closely synchronized to one another, and both of the servers were configured to accept email requests sent to 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 (moving to netlib.sandia.gov); Netlib.org, at the University of Tennessee; the UK Mirror Service in Kent; and Netlib.no in Bergen, Norway. These servers are synchronized to one another via a process described in [1]. Since the introduction of anonymous FTP access to the primary Netlib servers many more unofficial mirrors have been created; and 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. Recall from the previous section the critical role played by the chapter editors. Since this is a volunteer effort by busy people top in their field, Netlib has to make their life as easy as possible, and the principal contribution is 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 was, as far as we know, also a Netlib innovation.

2.3 Access protocols

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 (due to the need to explicitly route a message through the gateways) and unreliable. As text-based electronic mail was the only service common to all of those networks, this was the means initially utilized by Netlib to provide access to items in its collection, but as Internet protocols began to be more widely deployed other methods which were easier to use became feasible. A server for the Internet's File Transfer Protocol (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 [2]. 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 NEOS were much more influential in that direction for the numerical community.

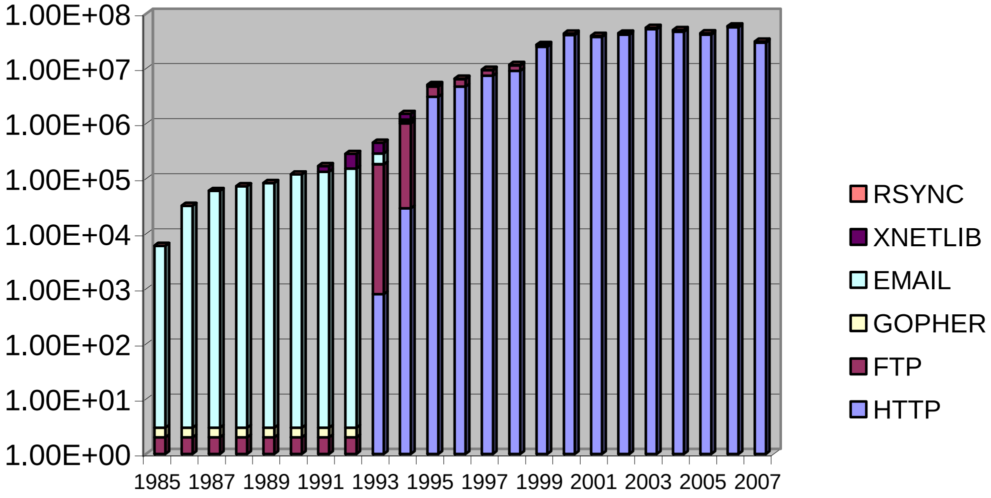
One of the unsuccessful features of Netlib was the use of digital signatures. Since large software is downloaded and compiled without inspection and may be distributed via mirror sites of unknown security, we thought people would appreciate having PGP-signed MD5 checksums. We also built mechanisms that 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 home pages, it could be inconvenient to locate contact information. The NA-Net database went far to solve this for the numerical community, but was never as complete as journal editors would like when looking for referees. Netlib introduced a way to search the SIAM membership list. Since this was an era when network connectivity was unreliable, the method adopted was to distribute an encrypted version of the database along with a program to allowed restricted searches to protect privacy and deter spam [7]. Eventually this was phased out in favor of the AMS Combined Membership List online.

Two innovative features worked well but were eventually casualties of being tied to a research operating system that is no longer is hosting 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 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 [12] operating system's generalized vision of a file.

2.4 Usage Patterns

At least at the UTK and ORNL servers, Netlib usage rose steadily in the period from 1985-2000. Traffic levels have varied in recent years. Some of the variation in recent years may be due to an increased use of crawlers and mirrors. Approximately 80% of the current traffic appears to be from individual users, 7% from mirrors, and 12% from web crawlers, which are presumably building indices for use by search engines.



Netlib Usage at UTK/ORNL
Figure 1

2.5 Community Impact

A large amount 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 maintain a moderated collection of high quality software which is organized and catalogued with appropriate metadata to enable easy searching. Netlib was not the first such collection; a few of the best-funded and well-run laboratories had local libraries tended by numerical consultants who grew to knew the local needs especially well. As one relevant example, the Stanford Linear Accelerator Center collaborated with the 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 such libraries by making them available world-wide, 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 has more than been 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 High Performance 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 showed 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 killed a smaller collection.

3. 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 whitepages database.

3.1. 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 Gene 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 [3].

At an early stage, Gene Golub collected addresses of numerical analysts. Jim Wilkinson once asked, “Whom can I contact on the net?” and Gene Golub passed along his file. It then occurred to Gene Golub that we ought to have universal addresses. So at Stanford was set up a system where a user could address a person as na.<lastname>@score. We felt that the na-net would allow our community to communicate more easily with one another, especially when net addresses were 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 took out 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 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, 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 February 13, 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 - in contrast to previous versions which 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. [4]

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 of 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.

3.2 NA-Digest History and Content

In the early days the NA-Digest came about because Gene Golub's secretary was sending the digest to the whole NA-Net address list. So we quickly put an end to that and the Digest came into being.

Gene Golub was the original editor of the NA-Digest. In July of 1987 Gene Golub's 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 of 2005. The digest is currently edited by Tamara Kolda of Sandia National Labs.

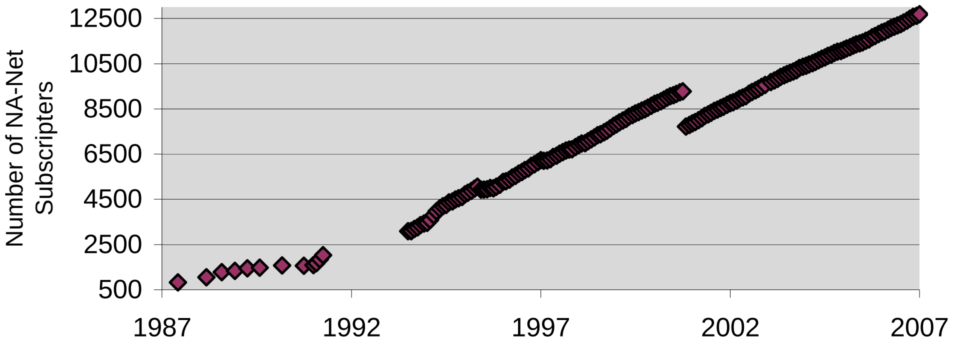
As the name suggests, the intent of the NA-Digest has been to have short announcements summarizing more extensive material available elsewhere. Today, almost all of the Digest contributions have URLs pointing to more complete announcements available on the Web.

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 members of the community. Examples of material commonly appearing in the digest include: conference announcements, workshop announcements, advertisements for jobs, software release announcements, change of address announcements, new book announcements, journal contents, and notices about community members: awards, significant achievements, and deaths.

Reasonably complete archives of the digest exist and are 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.

3.3 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 11295 subscribers today (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, i.e. `BITNET` addresses.)



NA-Net Subscriber Growth
Figure 2

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 by sending mail to na.sendlist@na-net.ornl.gov. Even after the list was removed from Netlib and the sendlist address was disabled, subscriber addresses were occasionally found to have been "leaked" by various means.

3.4 Community Impact

The international numerical analysis community is small enough that is still has a cohesive, “family” feeling. We believe that the impact of the NA digest is that it has helped maintain that feeling. The fact that it is still low tech, simple text email means that it is accessible to anyone who has 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. Moler reports that he has several times met people who recognize his name primarily as the “guy who sends me email every week.”

We are indebted to an anonymous referee 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 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."

4. Related Projects

Netlib had a number of spin off efforts which we will summarize here.

- XNetlib began in 1990 and predates 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 From 1994 - 2004, the NHSE (National HPCC 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 created, and are still being created, with the Repository in a Box (RIB) toolkit developed in 1996. However, continued operation of the site without funding has become impractical.
- Repository in a Box (RIB) provides a toolkit for building and maintaining metadata repositories. RIB was developed by the National HPCC Software Exchange (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. [5]
- NetBuild is a tool which 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.

5. Lessons for funding models

One reason to look at the history of successful projects 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 in its 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 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 its place on the center stage.

At the application level the science has to be captured in mathematical models, which in turn are expressed algorithmically and ultimately encoded as software. Accordingly, on typical projects the majority of the funding goes to support this translation process that starts with scientific ideas and ends 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 that has taken years to build up and that must be maintained, ported, and enhanced for many years to come if the value of the application codes that depend on it are 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.

Thus the life of computational science 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, multi-dimensional 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 or under funded or both.

The efforts of Netlib and the NA-Digest, for the most part, have been through volunteer effort. These services were viewed by the community as important and successful, which was by itself enough motivation 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 come up. 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 was true in past decades, we believe progress will require a greater level of sustained funding from governmental sources.

6. Summary

Software distributed by Netlib comes with the disclaimer that "anything free comes with no guarantee". In contrast to commercial vendors like NAG and IMSL, 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 even easier for the scientific computing community.

NA-Digest has provided a singular resource for the advancement of science and education by helping to create a community of those interested in scientific computation. It has allowed for cross-pollination of ideas and techniques amongst scientists, engineers, and numerical analysts, both from academia and industry.

6. Acknowledgements

The authors would like to 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 NSF 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.

7. References

[1] Eric Grosse. Repository Mirroring. *Transactions on Mathematical Software* 21(1), March 1995.

[2] Jack Dongarra, Thomas Rowan, Reed Wade. Software Distribution Using XNETLIB. *Transactions on Mathematical Software* 21(1), March 1995.

[3] Mark Kent. The Numerical Analysis Net (NA-NET). Technical Report 85, Institut für Informatik, Eidgenössische Technische Hochschule Zürich, January 1988.

[4] Jack Dongarra, Bill Rosener. NA-Net / Numerical Analysis Net. Technical Report ORNL/TM-11986, Oak Ridge National Laboratories, December 1991.

[5] S. Browne, P. McMahan, and S. Wells, "Repository In a Box Toolkit for Software and Resource Sharing," University of Tennessee, Knoxville, TN, Computer Science Technical Report, UT-CS-99-424, 1999. http://www.nhse.org/RIB/pubs/ut-cs-99-424.ps.gz

[6] K. Moore and J. Dongarra, "NetBuild (version 0.02)," Department of Computer Science, University of Tennessee, Knoxville, TN, CS Technical Report, UT-CS-01-461, March 2001. http://www.cs.utk.edu/~library/TechReports/2001/ut-cs-01-461.ps.Z

[7] Joan Feigenbaum, Eric Grosse and James A. Reeds, "Cryptographic Protection of Membership Lists", Newsletter of the International Association for Cryptologic Research 9:1 (1992) 16--20. ftp://cm.bell-labs.com/cm/cs/doc/91/4-12.ps.gz

[8] Donnalyn Frey and Rick Adams, "!%@:: A Directory of Electronic Mail Addressing and Networks: Third Edition", O'Reilly & Associates, 1993. (first edition was 1989)

[9] Ed Krol, "The Whole Internet User's Guide & Catalog: Second Edition", O'Reilly & Associates, Sebastopol CA, 1994. (first edition was 1992)

[10] C. Partridge and C. Mooers and M. Laubach, "The CSNET information server: automatic document distribution using electronic mail", SIGCOMM Comput. Commun. Rev., 17:4 (1987) 3--10. ACM Press, New York NY USA.

[11] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press, New York NY USA, 1992. (first edition was 1986)

[12] Pike, R., Presotto, D., Thompson, K., and Trickey, H. Plan 9 from Bell Labs. In Proceedings of the Summer 1990 UKUUG Conference (London, July 1990), pp. i-9.

[13] Tomovich, C., MOSIS-A Gateway to Silicon, IEEE Circuits and Devices Magazine, Mar 1988, Volume 4, Number 2, pp 22-23 ISSN: 8755-3996

Definition of Terms

AMS - The **American Mathematical Society (AMS)** is an association of professional mathematicians dedicated to the interests of mathematical research and scholarship, which it does with 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 - **EISPACK** is 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, etc.).

IMSL - The **IMSL 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 - **LINPACK** is a software library for performing numerical linear algebra on digital computers.

Macsyma - **Macsyma** is 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 - **MATLAB** is 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 E-mail or FTP.

MINPACK- **MINPACK** is 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 - QUADPACK is a FORTRAN subroutine package for the numerical computation of definite one-dimensional integrals.

SHARE - is 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.

Author Affiliations:
Jack Dongarra, University of Tennessee and ORNL (early work done at Argonne)
Gene Golub, Stanford University
Eric Grosse, Google (work done at Bell Labs)
Cleve Moler, MathWorks
Keith Moore, University of Tennessee

Jack Dongarra holds an appointment as University Distinguished Professor of Computer Science in the Computer Science Department 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 (ORNL), an Adjunct Professor in the Computer Science Department at Rice University, and Turing Fellow at the University of Manchester. He specializes in numerical algorithms in linear algebra, parallel computing, use of advanced-computer architectures, programming methodology, and tools for parallel computers. He has contributed to the design and implementation of the following open source software packages and systems: EISPACK, LINPACK, the BLAS, LAPACK, ScaLAPACK, Netlib, PVM, MPI, NetSolve, Top500, ATLAS, and PAPI. He is a Fellow of the AAAS, ACM, and the IEEE and a member of the National Academy of Engineering.

Gene Golub is the Fletcher Jones Professor of Computer Science at Stanford University. He was educated at the University of Illinois at Urbana-Champaign, receiving his B.S. (1953), M.A. (1954), and Ph.D. (1959) all in the field of Mathematics. He has been at Stanford since 1962, and has advised over thirty doctoral students, many of whom have themselves achieved distinction. He was awarded the B. Bolzano Gold Medal for Merits in the Field of Mathematical Sciences in addition he is a membership in the National Academy of Engineering, the National Academy of Sciences, and the American Academy of Arts and Sciences. He is a fellow of the AAAS and has been awarded eleven honorary degrees. One of his best-known books is Matrix Computations, co-authored with Charles F. Van Loan, and he is recognized as a major contributor to algorithms for matrix decompositions, such as the singular value decomposition (SVD).

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. PhD, Stanford, Computer Science. IEEE, SIAM, ACM.

Cleve Moler is chairman and chief scientist at The MathWorks, Three Apple Hill Drive, Natick, MA, 01760. He was educated at Caltech and Stanford.
He has been a professor at Michigan, Stanford, New Mexico, and UCSB. He is the original author of MATLAB, and of three textbooks on numerical methods.

Keith Moore received a B.S. in Electrical Engineering at Tennessee Technological University in 1984 and a M.S. in Computer Science at the University of Tennessee in 1996. He was a Research Associate in the Computer Science Department at the University of Tennessee from 1991 until 2007. He has participated in several Internet Engineering Task Force (IETF) standardization working groups beginning in 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 (URNs) and transition mechanisms for Internet Protocol version 6.