# John R. Rice: Mathematical Software Pioneer

Thomas Haigh
*University of Wisconsin, Milwaukee*

**Editor: Thomas Haigh**

John Rischard Rice is one of the founders of mathematical software as a distinct scholarly community. Trained in mathematics he spent four decades in Purdue University's Department of Computer Sciences. During the 1970s, he convened a series of seminal conferences on the topic and established *ACM Transactions on Mathematical Software*, which he edited for many years. Rice led development of the Ellpack system for solving partial differential equations and helped to establish techniques for the performance evaluation of mathematical software. He has published 21 books.

### Early Years

Born on 6 June 1934 in Tulsa, John Rice spent his early years in small town Oklahoma. At the age of 14, he left for Ethiopia, thanks to a temporary assignment for his father. Three years later, his family returned to Oklahoma and he enrolled in Oklahoma State University. During his five years in Stillwater, he earned both a BS and MS in mathematics.[1]

Engineers and other technical workers were in short supply during the early- and mid-1950s as the US ramped up its high-technology defense industry to maintain its lead over the Soviet Union. Rice spent two summers working in the Los Angeles area, then the world center for scientific computer use with its base of military contractors such as Douglas Aircraft, the Rand Corporation, and Lockheed. In 1954 a campus recruiter lured him half a continent away to North American Aviation, where an IBM 701 had recently been installed. Rice spent his first summer operating a desk calculator, but when he returned the next year, the data-smoothing process had been automated and he was able to shift his focus toward computer programming. He worked with a team designing a simple digital guidance computer for a ramjet missile. Rice recalled that actual coding work at North American was undertaken primarily by women with high school educations, so as a mathematician, he worked collaboratively with a woman named Melba Nead to create programs.

In 1956, Rice moved to California full time, enrolling in the PhD program in mathematics at the California Institute of Technology. Caltech was a shock for Rice—when teaching calculus to undergraduates he realized that "at least half of the students there were smarter than I was." Faculty expectations for doctoral students were even higher. Rice remembered most students in the 1950s failing their qualifying examinations, thanks to a misguided "idea that we only give PhDs to von Neumann and up." At that time, Rice's interest in automatic computation was unusual in the department, although numerical analysis pioneer John Todd joined its faculty while he worked on his dissertation. Rice continued to work at North American, ultimately deriving the topic of his PhD thesis on nonlinear approximation from the curve-fitting procedures required to squeeze large tables of coefficients into the tiny memories of early computers.[2] His experiences at North American also provided the topic of another early paper on the application of Runge-Kutta methods for solving differential equations.[3]

After graduating in 1959, Rice moved to Washington D.C. for a one-year postdoctoral position at the National Bureau of Standards in a small group of numerical analysts headed by Phil Davis. This was also his first exposure to programming in a high-level language, as

## Background of John R. Rice

**Born:** 6 June 1934 in Tulsa, Oklahoma

**Education:** Oklahoma State University, BS (mathematics), 1954; Oklahoma State University, MS (mathematics), 1956; California Institute of Technology, PhD (mathematics), 1959.

**Employment:** National Bureau of Standards, postdoctoral researcher, 1959–1960; General Motors, senior research mathematician, 1960–1964; Purdue University, professor, 1964–1989, W. Brooks Fortune Distinguished Professor, 1989–2004; professor emeritus, 2004 to present; Arxan Technologies, founder and scientist, 2001 to present.

**Selected Honors and Awards:** fellow of the American Association for the Advancement of Science, 1972; member of the National Academy of Engineering, 1994; fellow of the ACM, 1996.

 Published by the IEEE Computer Society 1058-6180/10/$26.00 

Fortran had arrived at the bureau. In 1960, he began a four-year spell working for General Motors, which maintained a small team of mathematicians in Warren, Michigan. This was a research group rather than a support operation, although Rice remembered that some of its staff took the opportunity to work with problems from the firm's operations. Among his colleagues was Carl deBoor, who would later earn a PhD and go on to a distinguished academic career. Rice also got involved with Share, the user group for large scientific IBM computers. He contributed routines for elementary functions to its library and worked in its Numerical Analysis Project, led by Hirondo Kuki of the University of Chicago Computation Center, to improve the quality of the mathematical routines in the library.[4]

### *Purdue University*

At General Motors, Rice was free to follow his research interests without reference to the immediate needs of the firm's engineers. But he was aware that "it was in some sense a dead end position. I would never get anywhere at General Motors if I didn't actually start working for General Motors..."[5] The prospect did not appeal to him, so Rice moved to a faculty position in the rapidly growing computer sciences department at Purdue University in 1964. Founded in 1962, this department was one of a handful with a PhD program in computing. He found himself with less time to do research than at General Motors, but he enjoyed the exposure that teaching gave him to broad areas of computer science. This ultimately led him to write a popular series of introductory textbooks, beginning with the 1969 *Introduction to Computer Science*.[6] Rice recalled that

> when I got interested in this introduction to computer science, we had a big fight in the department over how to redo the department's introductory course and I won the fight by volunteering to teach the course...[7]

The course had previously used one of Dan McCracken's books on Fortran, but Rice favored a focus on programming principles such as algorithms and data structures over the mechanics of particular languages. According to Rice, his book "was a very good seller for a few years, until other people found out what I was finding out: that students don't want to learn principles, they

**Rice found himself with less time to do research at Purdue, but he enjoyed the exposure that teaching gave him to broad areas of computer science.**

want to learn how to program."[8] A later textbook, *Computers: Their Impact and Use*, followed in the mid-1970s and was intended to give a broad grounding in computing for nontechnical students. Variants were produced teaching Fortran, Basic, and PL/1.[9] Rice also appreciated the chance to work with graduate students. Among them were Tom Aird, later a technical leader of the mathematical software company IMSL; Ronald F. Boisvert, who led work in applied mathematics at the National Institute of Science and Technology; and a number of others with successful university careers.

Rice mentioned that the strength of Purdue's applied mathematics staff was also a major appeal. His initial appointment was as a full professor jointly between the two departments and his research continued to focus on mathematical methods. According to Aird,

> John was famous for developing polyalgorithms, so that meant instead of just one algorithm for a specific problem you could write a program that would have some logic or some elements of knowledge of the problem area, and put together one of several algorithms.[10]

For his first decade at Purdue, approximation theory remained Rice's main research area. It was the topic of his first book, a survey for graduate students.[11] But he recalled that eventually "people were counting the number of angels that dance on the head of a pin as opposed to solving real problems." As a result, he "gradually got less and less enthused... when I entered the field it was very dynamic." His focus shifted to the solution of partial differential equations (PDEs).[12] (One result was the Ellpack project, which I discuss later.)

In 1983, Rice became chair of the computer sciences department and shifted his appointment there full time. He has written at length on the history of this department.[13] He saw it as being consistently marginalized by the university administration, writing in *Annals* that ''Purdue's Department of Computer Sciences was consistently rated in the top 10 (higher than any other Purdue science department), yet it had to continually reaffirm its performance and value within the university.''[14] In the 1980s it was plagued, like other departments of its kind, with a spike in student numbers without any corresponding surge in faculty numbers and resources. Rice believed this led to an exodus of talented faculty, including department head Peter Denning, producing a situation where ''graduate courses have 110 students and then that kind of thing, which our dean thought was fine.''

Other departments faced the similar pressures during this era, but Rice believed some factors were specific to Purdue. One of those was the aforementioned dean who refused to allow vacant faculty positions to be filled so that ''Computer Science was getting what it deserved, which was very little.''[15] As department chair from 1983 to 1996, Rice had some success in rebuilding its fortunes, although a second boom in enrollment in the mid-1990s led to another resource crisis. From 1992 to 1994, Rice organized a new interdisciplinary graduate program in computer science and engineering, which attracted large numbers of students despite limited support from the university administration. During our interview, he complained repeatedly of disparities in funding and resources between computer science and physics. His term as department head ended following a confrontation with the dean of the College of Science on this topic.[16]

According to Rice, none of this turmoil ever made him seriously consider leaving Purdue.

> I was very comfortable . . . my family liked it here, I liked living in a small town. I've lived in places like Detroit and Washington and Los Angeles, and I didn't care for those places. So . . . I never actively looked for a job.

However, he did regret having ''wasted a lot of energy in being a department head,'' having finally realized that such positions ''are really losers for faculty unless you want to become the president of the university . . . I should have let somebody else do it.''[17]

### *Building a Community for Mathematical Software*

Rice did more than anyone to nurture the creation of mathematical software as a distinct field of research and a well-defined international community of researchers. In the late 1960s, the main gathering point for those interested in mathematical software was ACM SIGNUM, its special interest group on numerical mathematics. Joe Traub had initiated the *SIGNUM Newsletter* in 1966 (then known as the *SICNUM Newsletter*, the C standing for committee). Rice was chosen to deliver its first official lecture at the main ACM annual meeting.[18] Its membership grew rapidly, quickly exceeding 1,000 people (although no dues were charged at that point). Over the next few years, SIGNUM organized sessions at larger ACM meetings, sponsored an award, and organized a series of themed workshops on particular topics related to numerical software. Although there were many other venues suitable for discussion of numerical analysis, Rice believes that SIGNUM stood out as a place where software issues and work of an experimental character could be presented.[19]

Rice organized the Mathematical Software Symposium held at Purdue in April 1970, sponsored by SIGNUM and paid for with a grant from the Office of Naval Research. Within the field, this is regarded as the single most important event in the emergence of a community of skilled researchers who defined themselves primarily as experts in mathematical software. Its proceedings appeared as a book in the ACM Monograph series the next year.[20] As well as the proceedings of the papers presented at the conference, and a selection of four exemplary pieces of software, the book included a preface and four introductory chapters from Rice summarizing the history, current state, and challenges of mathematical software at that time. In the preface he noted that the ''scientific public'' still viewed the creation of software as ''mechanical, even if tedious and time consuming'' and lacked the discernment needed to separate software written with the skill of ''college freshmen'' from those ''few people who are much, much better.'' He diagnosed a second challenge, this one ''sociological, in that mathematical software is not yet a viable scientific community or entity.''[21]

The conference played an important role in overcoming both challenges. Rice recalls that ''a lot of common interests surfaced there. It brought together people who didn't usually speak to each other."[22] Social connections began to develop between people working on packages for different kinds of software. Rice organized a second conference on the subject at Purdue in 1974, and a third followed in Madison in 1977. Looking back on Rice's career, three of his students wrote that the conference and its sequel were "key events which served to galvanize a community of researchers which continues to be productive to this day."[23]

In 1974, a core community of mathematical software experts was institutionalized as the International Federation for Information Processing (IFIP) Working Group 2.5 on mathematical software. Rice was a founding member and remains an active participant. The system of IFIP working groups had been around since the early 1960s, with Working Group 2.1 on Algol the most prominent. The new group was an invitation-only elite international club. It held annual meetings, ran a series of technical projects, and organized a series of thematic working conferences every three years or so. These meetings and conferences followed a similar pattern to the earlier mathematical software conferences.

### *ACM Transactions on Mathematical Software*

Rice suggested that planning for a journal on mathematical software began around 1971, probably at an National Science Foundation (NSF) Workshop on Software and Algorithm Evaluation held in Colorado. To this point the main publication venue for mathematical software had been the Algorithms department of *Communications of the ACM*. According to Rice, this suffered from two main deficiencies: the trivial nature of many of the programs published there and the use of Algol as a publication language at a time when real work was done almost exclusively in Fortran. Lloyd Fosdick, then the department's editor, was a mathematical software enthusiast who shared Rice's interest in a new venue where large, useful peer-reviewed programs could be published in practical languages accompanied by machine readable versions of the source code. Rice felt that "mathematical software somehow has a niche between theory and practice where algorithms can thrive" and that practicing scientists and engineers could benefit from, and even contribute to, a journal of this kind. It would also benefit the emerging mathematical software community. Rice recalled that many of them were academics who were looking for academic respectability, and so were in grave need of "a place I can publish papers and this will help me."[24]

**The first issue of the ACM Transactions on Mathematical Software (TOMS) appeared in 1975 with Rice as editor in chief.**

Work proceeded slowly. One difficult question was the appropriate sponsoring society. Mathematical software lay at the intersection of several disciplines. Numerical analysis had been a core area of early computer science departments in the 1960s, but by the 1970s, it was becoming more marginal within computer science as the field sought a unique body of theory. Work on numerical analysis gradually migrated from the ACM to the Society for Industrial and Applied Mathematics (SIAM), which was already well established as a technical publisher in the area. Yet Rice found that mathematical software held less legitimacy as a research area for mathematicians than for computer scientists. According to Rice,

> SIAM has never been very graceful about admitting that computer science is a real discipline, and they're still having a hard time seeing that things like programming and systems have any intellectual content. I talked to SIAM at length on the *Journal of Mathematical Software* and it was that "software" word that was the problem they had. There were plenty of people in the power structure of SIAM who did not want software.[25]

Working with the ACM posed its own challenges in the early 1970s. Rice cites the influence of several ACM leaders, most notably Herb Grosch, who were "absolutely totally opposed to having academics [be] influential in ACM, and that means not having any scholarly research oriented journals." In the absence of reliable accounting information the associations' scholarly journals were being blamed for its financial crisis.

**Ellpack helped bring several then-novel software engineering ideas into the world of mathematical software.**

However, Rice was a member of the ACM Publications Board and was able to make a strong case for the proposed journal's financial viability, boosted by a subsidy from the NSF for its first issues to be used as conference proceedings for the Second Mathematical Software Conference.[26]

The ACM Council agreed, and the first issue of the *ACM Transactions on Mathematical Software* (TOMS) appeared in 1975 with Rice as editor in chief. Large programs were included on a microfiche card provided with the journal and could be ordered on magnetic tape or punched cards if needed from a collection known as Calgo. Initially, this library was administered by IMSL, but it eventually shifted to the online Netlib service. Netlib's founder, Jack Dongarra, credits Rice with securing inclusion of TOMS material in the online Netlib collection, saying also that ''John was the person who originated the journal and basically was the founder of this idea of mathematical software.''[27] Rice remained editor in chief until 1993, during which time the journal prospered and published many important pieces of software.

The concept of peer reviewing software in pursuit of higher quality seems to be unique to the mathematical software community and is quite startling to outsiders. An earlier attempt to introduce this for routines in the IBM Share library had lost steam in the 1960s. But TOMS made the concept work, and within this community, software has been peer reviewed for decades. W.M. Cody, an early editor of TOMS, explained the process:

> It was the usual responsibilities, you received manuscripts from Rice and found referees for them. In the case of this journal, many of the manuscripts involved computer programs of one sort or another, mathematical software. So we tried to find somebody who would test it and banter back and forth about the quality of the software as well as the importance, as well as the text material.[28]

### Ellpack

His growing interest in mathematical software also led Rice to lead the development of what became the Ellpack project. In mid-1976, a meeting to explore a new software framework for solving partial differential equations (PDEs) was held at NASA's Institute for Computer Applications in Science and Engineering with research groups from centers such as the University of Texas, Purdue, and Yale. The original idea came from Harvard University's Garrett Birkhoff, with institutional support for this initial meeting from NASA's Jim Ortega. Development of Ellpack was supported by a series of NSF grants.

Ellpack was conceived as a modular framework for elliptical PDE solvers, letting researchers state test problems in a standard format and apply a specified set of problem solving modules to test the results. Researchers in this area focused on different steps in the solution process, for example, in ''discretizing'' the PDEs into systems of linear equations or into solving those systems of linear equations. Rice believes that progress was being slowed by the need to write complete programs and provide suitable test output just to experiment with changes to one part of the solution process. Over time, the system would accumulate a standard library of test problems and automatically log the results of each run to record the amount of time and memory consumed to solve them with a particular combination of solvers on a particular machine. This reflected an evolution of mathematical research, at least within this community, toward the experimental investigation of the real-world performance of particular mathematical methods when executed on actual computer architectures. This was the topic of the IFIP WG 2.5 Conference on Performance Evaluation of Numerical Software held in 1978.[29]

In a paper presented at the third Mathematical Software Symposium in 1977, Rice wrote that ''one of its primary objectives is to test the concept of a modular approach using interchangeable software parts.''[30] Ellpack's technical structure supported a loosely coupled collaborative community. In language strikingly reminiscent of that applied to modern open source software projects, Rice wrote in 1977 that his role was that of ''a coordinator and benevolent dictator.'' Code came from collaborators including David Young at the University of Texas, Martin Schultz at Yale, and Allan George at Waterloo as well as from Rice and his colleagues at Purdue. Others, such as Jim Ortega and Gene Golub, provided advice and facilitation to the project.

After several interim releases, a final version of Ellpack was accompanied by a book in 1985.[31] Ellpack itself was written, like other mathematical software packages of its day, in Fortran with special attention to portability so that it could easily be compiled on a range of machines. Its first implementation was as a Fortran preprocessor. It came with an impressive collection of solver modules and a library of more than 50 test problems.

Collaborative mathematical software projects to produce ''Packs'' were increasingly common in this era, inspired by the success of Eispack in the early 1970s and the successor Linpack project, which was then ongoing. Ellpack was designed to support collaboration between a handful of groups researching software for the solution of elliptical PDEs. This was already a reasonably mature and active area, with applications in many areas of science and engineering. But according to Rice, the connections between different methods were not well understood and their development remained crude by today's standards.[32]

Ellpack helped bring several then-novel software engineering ideas into the world of mathematical software. Its construction as a set of what Rice came to call ''software parts'' reflected a major research trend of the era toward techniques for modularization and code reuse. Later Ellpack versions included a special-purpose programming language to describe the equations to be solved were described and the series of steps to be applied to them—another very modern idea at this time.[33] During the 1970s, special-purpose end-user languages were becoming increasingly popular for other kinds of mathematical software such as statistical software and matrix manipulation.[34]

By this point, Rice believed that Ellpack had evolved from serving primarily as a test bed for researchers into a tool used by scientists and engineers with technical problems to solve.[35] Approximately 500 copies were distributed. Numerical analysts are forever bemoaning the tendency of engineers to stick with the mathematical methods they learned in college textbooks, however inefficient and inaccurate those might be. Ellpack packaged the fruits of recent mathematical research to disseminate a range of efficient methods in accessible form. Nothing in the software itself helped users to pick the right method, so this was the main concern of the accompanying book.[36] Less complete selections of PDE solvers were also included in the two commercial mathematical libraries then available, IMSL and NAG, but unlike these more general products, Ellpack did not require its users to write programs in Fortran.

### Other Projects

The final main phase of development work on Ellpack, from around 1982, was supported by a major NSF grant funding a broader Purdue project on ''computing about physical objects.'' This broader context led Rice to a more general interest in the creation of what he came to call ''problem-solving environments,'' in which scientists and engineers could work interactively with a range of computerized capabilities to solve their problems. According to Rice, this idea took shape at a 1985 workshop attended by various experts in mathematical software and numerical methods.[37]

To support this vision, Rice immersed himself in the literature on artificial intelligence, particularly the then-fashionable world of expert systems. He even taught an introductory course in AI. Rice organized another series of conferences at Purdue to explore the application of AI techniques and had some success in shifting the attention of scientific software researchers toward the user-oriented issues inherent in the concept of a problem-solving environment.[38] According to Rice, the flow of ideas was largely one way, as AI experts were not particularly interested in mathematical software.

Rice's interest in user experiences led him to another important project in the early 1980s. Founded in 1970, IMSL was one of two major firms worldwide in the niche business of commercial mathematical software libraries. For many years, Rice was an active member of its technical advisory board, reflecting the firm's deep ties with the community of academic mathematical software specialists. As IMSL sought to diversify beyond its core market of scientific computer centers, Rice persuaded it to implement a new high-level language for end users called Protran. The language was the topic of one of Rice's own books.[39] Like Ellpack, it was initially created as a Fortran preprocessor.

Tom Aird acknowledged Rice's contribution to this project:

> [T]here's a quote in the manual that I think was very insightful where John Rice writes, "The PROTRAN system is an extension of FORTRAN which brings many of the IMSL library programs into the language in a simple natural way. It provides much shorter programs for various examples, and two, it is an example of the higher level languages that will spread throughout numerical and scientific computation in the coming years." So it seemed compelling to people who thought about these things that that was a natural progression of how programming would be done...[40]

Much to Aird's regret, IMSL was unable to market Protran successfully as a commercial product, perhaps because it required a much larger volume of low margin sales and was not well suited to the firm's existing sales force. In retrospect, however, Rice was clearly correct in recognizing the need for rise of specialized high-level languages such as those used by Maple, Matlab, and Mathematica.

By the late 1980s, the dominant architectures for high-performance computing were shifting away from single-processor machines. Rice and his colleagues launched a project to create a parallel successor to Ellpack, known as //Ellpack. As well as parallel versions of the discretizers, solvers, and other tools this also included GUI and visualization tools, exemplifying Rice's commitment to the creation of problem-solving environments.[41] According to Rice,

> Parallel ELLPACK has, I believe, a million and a half lines of code in it. So it was really a major operation of integrating very large software packages and it had FORTRAN, it had C, it had LISP, it had assembler, it had all kinds of languages there. Portability went out the window, and there was no idea of having other people use it.... It hasn't been widely used because it won't move.[42]

### *Recent Career and Professional Service*

The career of a distinguished scientist often broadens to include involvement with science policy and advocacy. For Rice, that transition took place with his participation in the Computer Science and Engineering Research Study (COSERS) initiative of the late 1970s. This high-powered committee—which included participants Alan Newell, Jean Sammet as ACM chair, Bernie Galler, Juris Hartmanis, Peter Denning, Richard Karp, and Donald Knuth—delivered an influential report intended to demarcate the full scope of computing research. Rice chaired its panel on numerical computation, acting as the main author for this section of the report.[43] He was also an enthusiastic participant in the Computing Research Association, a group founded in the 1970s to represent the common interests of elite computer science departments and professional associations. As Rice put it, "it took me quite a long time to really understand that computer science as the new kid on the block ... was getting the short end of the stick from federal funding."[44] This mirrored his concerns about resource discrimination against computer science within Purdue. Rice helped the organization to expand to hire a full-time executive director and boost the scale of its lobbying work on behalf of computing research. He was its chair from 1991 to 1993.

Rice also made major contributions to the *Computing in Science & Engineering* (formerly *IEEE Computational Science and Engineering*) and the International Association for Mathematics and Computers in Simulation.

One milestone in Rice's career came when an international symposium was held in honor of his 65th birthday in 1999.[45] In the opening essay of its proceedings, Ronald F. Boisvert suggested that Rice's contributions followed three main themes: the architecture of scientific software systems (including the concept of polyalgortihms and a push for software components), raising the level of abstraction (through techniques such as high-level languages and problem-solving environments), and an experimental approach toward understanding mathematical software through performance evaluation and testing.[46]

Rice retired from Purdue in 2004, transitioning to emeritus status. That year he told me that "currently I'm spending almost all of my time on computer security."[47] His interest in the field began with a collaboration with fellow Purdue professor Mikhail J. Atallah to investigate ways to outsource computation work without revealing the content of the problem being solved.[48] This led to a company Arxan Technologies, backed by venture capital, to create tamperproof software. Rice was then spending about half his time working to get the company up and running. Today, Arxan Technologies has a professional management team, but Rice and Atallah remain members of its Technical

Advisory Board. His personal and professional life remain active. As I write, he is on an extended honeymoon trip to China, having recently remarried.

## References and notes


1. J.R. Rice, ''Oral History Interview with Thomas Haigh,'' 2004; http://history.siam.org/pdfs2/Rice_returned_SIAM_copy.pdf. Facts in this biography not otherwise attributed are derived from this oral history.
2. J.R. Rice, ''The Characterization of Best Nonlinear Tchebycheff Approximations,'' *Trans. Am. Mathematical Soc.,* vol. 96, no. 2, 1960, pp. 322–340.
3. J.R. Rice, ''Split Runge-Kutta methods for simultaneous equations,'' *J. Research of the National Bureau of Standards,* vol. 64B, no. 3, 1960, pp. 151–170.
4. The project is mentioned in W. Kahan ''In Memoriam: Hirondo Kuki,'' *ACM SIGNUM Newsletter,* vol. 7, no. 1, 1972, pp. 8–10. Its records are preserved in the archives of the Nat'l Museum of Am. History.
5. Rice oral history interview, p. 40.
6. J.K. Rice and J.R. Rice, *Introduction to Computer Science,* Holt, Rinehart, and Winston, 1969. The other author was Rice's father.
7. Rice oral history interview, p. 66.
8. Rice oral history interview, p. 67.
9. The first of these was R.E. Lynch and J.R. Rice, *Computers: Their Impact and Use – with Basic Programming,* Holt, Rinehart, and Winston, 1975.
10. T. Aird, ''Oral History Interview by Thomas Haigh,'' 3 and 4 June, 2004, Society for Industrial and Applied Mathematics (SIAM), http://history.siam.org/aird.htm.
11. J.R. Rice, *The Approximation of Functions, vol. 1: Linear Theory,* Addison Wesley, 1964. A second book explored less well established methods closer to his own research. J.R. Rice, *The Approximation of Functions, vol. II: Nonlinear and Multivariate Theory,* Addison Wesley, 1969.
12. Rice oral history interview, p. 40.
13. See also S. Rosen and J.R. Rice, ''The Origins of Computing and Computer Sciences at Purdue University,'' *Studies in Computer Sciences: In Honor of Samuel D. Conte,* R. De Millo and J.R. Rice, eds., Plenum, 1994, pp. 31–44, and J.R. Rice and S. Rosen, ''History of the Computer Sciences Department of Purdue University,'' *Studies in Computer Sciences: In Honor of Samuel D. Conte,* R. De Millo and J.R. Rice, eds., Plenum, 1994, pp. 45–72.
14. J.R. Rice and S. Rosen, ''Computer Sciences at Purdue University—1962 to 2000,'' *IEEE Annals of the History of Computing,* vol. 26, no. 2, 2004, pp. 48–61.
15. Rice oral history interview, p. 72.
16. Rice discussed his own role in more detail in a 1997 oral history interview J.R. Rice, ''Oral History Interview by William Aspray,'' OH 320, 3 October 1997, Charles Babbage Inst., Univ. of Minnesota. This has not been transcribed, but tapes can be ordered.
17. Rice oral history interview, p. 68.
18. See vol. 1, no. 1 of the newsletter (June 1966), accessible via the ACM Digital Library.
19. Rice oral history interview, p. 21.
20. J.R. Rice, ed., *Mathematical Software,* Academic Press, 1971.
21. Rice's, *Mathematical Software,* pp. xv–xvi.
22. Rice oral history interview, p. 34.
23. R. Boisvert, W. Dyksen, and E. Houstis, ''Editorial: Special Issue in Honor of John Rice's 65th Birthday,'' *ACM Trans. Mathematical Software,* vol. 26, no. 2, 2000, p. 223.
24. Rice oral history interview, p. 20–24, quotes, p. 23, 24.
25. Rice oral history interview, p. 20.
26. Although the quotations here are from my oral history interview with Rice, he has also told the story (with supporting data) in J.R. Rice, ''Mathematical software and ACM publications,'' *A History of Scientific Computing,* S. Nash, ed., Addison-Wesley, 1990, pp. 217–227.
27. J.J. Dongarra, ''Oral History Interview by Thomas Haigh,'' 26 April 2004, Society for Industrial and Applied Mathematics, http://history.siam.org/oralhistories/dongarra.htm. On the Netlib story, see J. Dongarra et al., ''Netlib and NA-Net: Building a Scientific Computing Community,'' *IEEE Annals of the History of Computing,* vol. 30, no. 2, 2008, pp. 30–41.
28. W.J. Cody, ''Oral History Interview by Thomas Haigh,'' 3 and 4 August 2004, Soc. for Industrial and Applied Mathematics http://history.siam.org/oralhistories/cody.htm.
29. L.D. Fosdick, ed., *Performance Evaluation of Numerical Software: Proc. IFIP TC 2.5 Working Conf.,* North-Holland, 1979.
30. J.R. Rice, ''Ellpack: A Research Tool for Elliptic Partial Differential Equations Software,'' *Mathematical Software III,* J.R. Rice, ed., Academic Press, 1977, pp. 319–342. The history of Ellpack is also discussed in W. Cowell ed., *Sources and Development of Mathematical Software,* Prentice-Hall, 1984, pp. 231–237.
31. J.R. Rice and R.F. Boisvert, *Solving Elliptic Problems Using ELLPACK,* Springer-Verlag, 1985.
32. Rice oral history interview, p. 42.
33. Rice oral history interview, p. 44.
34. Matlab originated for instructional use on timesharing systems around this time, with a specialized command language. T. Haigh, ''Cleve Moler: Mathematical Software Pioneer and Creator of Matlab,'' *IEEE Annals of the History of Computing,* vol. 30, no. 1, 2008, pp. 87–91.

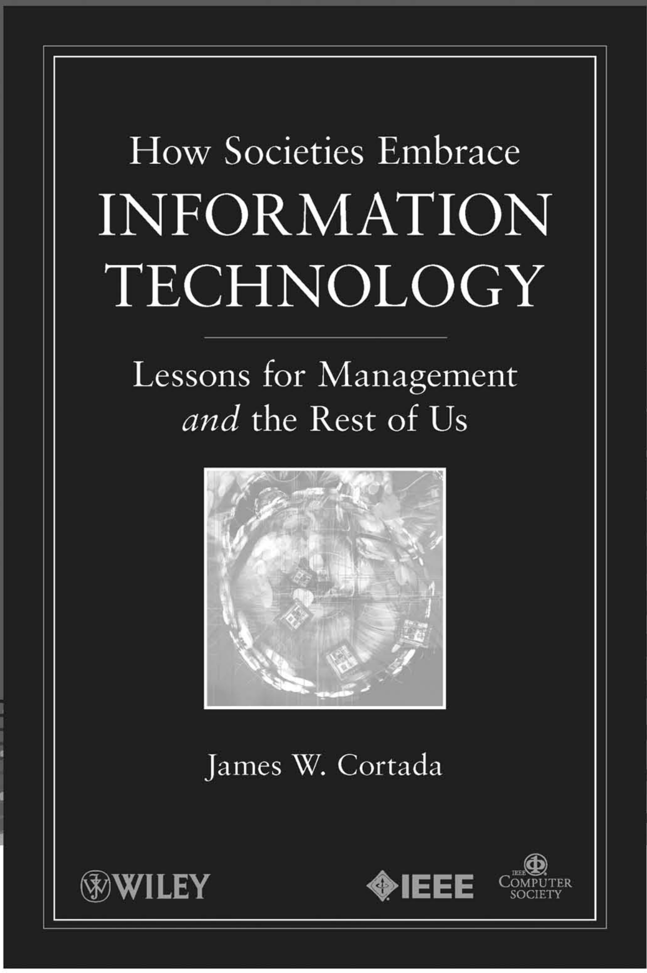
## Correction

Page 78 of the biography of Heinrich Welker published in the April–June 2010 issue (vol. 32, no. 2) incorrectly credited Welker with ideas leading to ‘‘the 1970 inauguration of the Arpanet, precursor of the Internet.’’ This mistaken claim was introduced during the editing process. Thanks to Alex McKenzie for drawing the error to our attention.


35. Rice oral history interview, p. 48.
36. J.R. Rice and R.F. Boisvert, *Solving Elliptic Problems Using ELLPACK,* Springer-Verlag, 1985.
37. See B.J. Ford and F. Chatelin, ed., *Problem Solving Environments for Scientific Computing: Proc. IFIP TC2 Wg 2.5 Working Conf. Problem Solving Environments for Scientific Computing,* Elsevier Science, 1987.
38. The first of three edited volumes to emerge from these meetings were in E.N. Houstis, J.R. Rice, and R. Vichnevetsky, eds., *Intelligent Mathematical Software Systems,* North Holland, 1990.
39. J.R. Rice, *Numerical Methods, Software and Analysis,* McGraw-Hill, 1983.
40. T. Aird, ‘‘Oral History Interview by Thomas Haigh,’’ p. 56.
41. E. N. Houstis et al., ‘‘//ELLPACK: A Numerical Simulation Programming Environment for Parallel MIMD Machines,’’ *Proc. 4th Int’l Conf. Supercomputing* (ICS 90), ACM Press, 1990, pp. 96–107.
42. Rice oral history interview, p. 60.
43. B.W. Arden, ‘‘The Computer Science and Engineering Research Study (COSERS),’’ *Comm. ACM,* vol. 19, no. 12, 1976, pp. 670–673.
44. Rice oral history interview, p. 71.
45. R.F. Boisvert and E. Houstis, eds., *Computational Science, Mathematics and Software: Proc. International Symp. Computational Science in Celebration of the 65th Birthday of John R. Rice,* Purdue Univ. Press, 2002.
46. R.F. Boisvert, ‘‘Mathematical Software: Past, Present, and Future,’’ *Computational Science, Mathematics and Software,* R. Boisvert and E. Houstis, eds., Purdue Univ. Press, 2002, pp. 3–27.
47. Rice oral history interview, p. 75.
48. M.J. Atallah et al., ‘‘Secure Outsourcing of Scientific Computations,’’ *Advances in Computers, vol. 54: Advances in Software Engineering,* M.V. Zelkowitz, ed., Academic Press, 2001, pp. 215–272.


**Thomas Haigh** is a historian of computing and an associate professor in the School of Information Studies at the University of Wisconsin, Milwaukee. Contact him at thaigh@computer.org.

# NEW TITLE FROM WILEY & IEEE CSPress




The new book by James W. Cortada, author of *The Digital Hand*, discusses the role of corporations and governments in the over $2 trillion annual world-wide expenditure in acquisition and deployment of IT.

The author's royalties will be donated to the IEEE Computer Society's Educational Activities Board.


978-0-470-53498-4 • November 2009
Paperback • 272 pages • $29.95
A Wiley-IEEE Computer Society Press Publication


**To Order**

**North America**
1-877-762-2971

**Rest of the World**
+ 44 (0) 1243 843291

**15% Off for CS Members**

WILEY *Publishers Since 1807*

IEEE

IEEE computer society
www.computer.org