Paul McJones

Paul McJones [Computer History Museum volunteer, born about 1950?] studied engineering mathematics at the University of California, Berkeley, from 1967 to 1971. He has worked since 1967 on operating systems, programming environments, transaction processing systems, and enterprise and consumer applications; for the University of California, IBM, Xerox, Tandem, Digital Equipment Corporations, and Adobe Systems. In 1982, he and coauthors received the Association for Computing Machinery *Programming Systems and Languages Paper Award* for the paper "The Recovery Manager of the System R Database Manager".

This article, published as an "anecdote" in *IEEE Annals of the History of Computing* (39:2, July 2017) https://ieeexplore.ieee.org/document/7974703/), tells the story behind a Computer History Museum project. The article is followed in this file by McJones' summary of the project report.

IEEE Editor's Note

Our anecdotes are typically on topics in computing history, but sometimes are about doing historical research; the current anecdote is the latter. At this time, it is more important than ever that historians and practitioners seek out original computer code and documentation before it is lost forever. The Computer History Museum (CHM) is one of the institutions encouraging such collection and archiving; for example, through its Center for Software History (http://www.computerhistory.org/softwarehistory), its Software Industry Special Interest Group (http://www.softwarehistory.org), and its Software Preservation Group (http://www.computerhistory.org/groups/spg/). Here we publish CHM volunteer Paul McJones' story of his collecting and archiving historical Fortran materials.

In Search of the Original Fortran Compiler

By Paul McJones

Software Preservation Group, Computer History Museum

In April 2002, Grady Booch sent out an email with the subject "Preserving classic software products." His appeal to provide top-10 lists, and to suggest where to look for the source code, received an enthusiastic response. Grady and the Computer History Museum organized a workshop in October 2003.[1] As a result, a month later, a group of Computer History Museum trustees, staff and volunteers (including me) established the Software Collection Committee[2] (SCC) to explore the software side of the museum's mission: "To preserve and present for posterity the artifacts and stories of the information age." I decided to take on the project of collecting Fortran materials.

John Backus's Fortran project at IBM developed one of the first higher-level programming languages as well as a set of techniques for code optimization that allowed Fortran to compete with hand-tuned assembly language.[3] The combination of ease-of-use and efficiency led to Fortran's rapid adoption and lasting impact on programming language design and implementation. Fortran's success on the IBM 704/709 led to Fortran compilers for many brands and models of computers, making it possible to port Fortran programs among different computers; Fortran program portability also made Fortran useful for non-numerical projects. While Fortran's popularity waned as other languages grew up to fill various niches, some high-performance computing users continued to use Fortran, with periodic updates, to the present.

The history of the Fortran project is well documented in papers by Backus and others, covering the project, its impact, and technical analysis of the compiler itself.[4] However, the artifact itself – the source code for the original Fortran compiler – seemed to me an appropriate starting point to gain experience with collecting, preserving, and presenting historic software. By spring 2006, I had assembled a collection including listings, source and executable code, documentation, and more and had made the collection publicly accessible via a website. Take a look at that website[5] now to have it in mind as you read this article, which describes my collection efforts. I hope my experience will be useful to other computing history researchers from the practitioner world considering similar efforts.

Getting Started

I started my search for Fortran by calling John Backus, with whom I had worked in the mid-1970s (on functional programming) but had not seen for about 25 years. He did not have a copy of the source code and directed me to Irv Ziller, who had been his first hire on the project. Irv responded to my request, saying that he didn't have the source code, but he recalled material being sent to the Smithsonian to become part of their collection. I visited the website of the Smithsonian National Museum of American History (NMAH) and located the group that had created the Computer History Collection,[6] but my initial attempts to establish communication with the staff were unsuccessful.[7]

One day I was chatting with a coworker, Jim King, who had worked at IBM Research for many years. Jim had used Fortran II and the Fortran Monitor System on an IBM 709 in college in the early 1960s and had some interesting anecdotes.[8] Jim suggested IBM's SHARE user group library as a potentially interesting source of material, and he also recalled the extensive IBM 7094-related library at Boeing when he worked there in the 1960s. Another member of the SCC, Dick Sites, recalled having seen "handwritten Fortran I documentation in a basement of the Sloan building at MIT in 1965. … As I remember, it described the overall compiler design and optimizing algorithms. I remember a fairly neat but slightly small handwriting with hand-drawn boxes and arrows for some of the algorithms." Also, Len Shustek, Computer History Museum founder, pointed out this sentence in Backus's article[9] about Fortran at the first History of Programming Languages conference: "Shortly after the distribution of the system, we distributed – one copy per installation – what was fondly known as the 'Tome,' the complete symbolic listing of the entire compiler plus other system and diagnostic information, an 11 by 15 inch volume about four or five inches thick." Presumably this was different from what Sites had seen, since it was not handwritten.

Digging Deeper

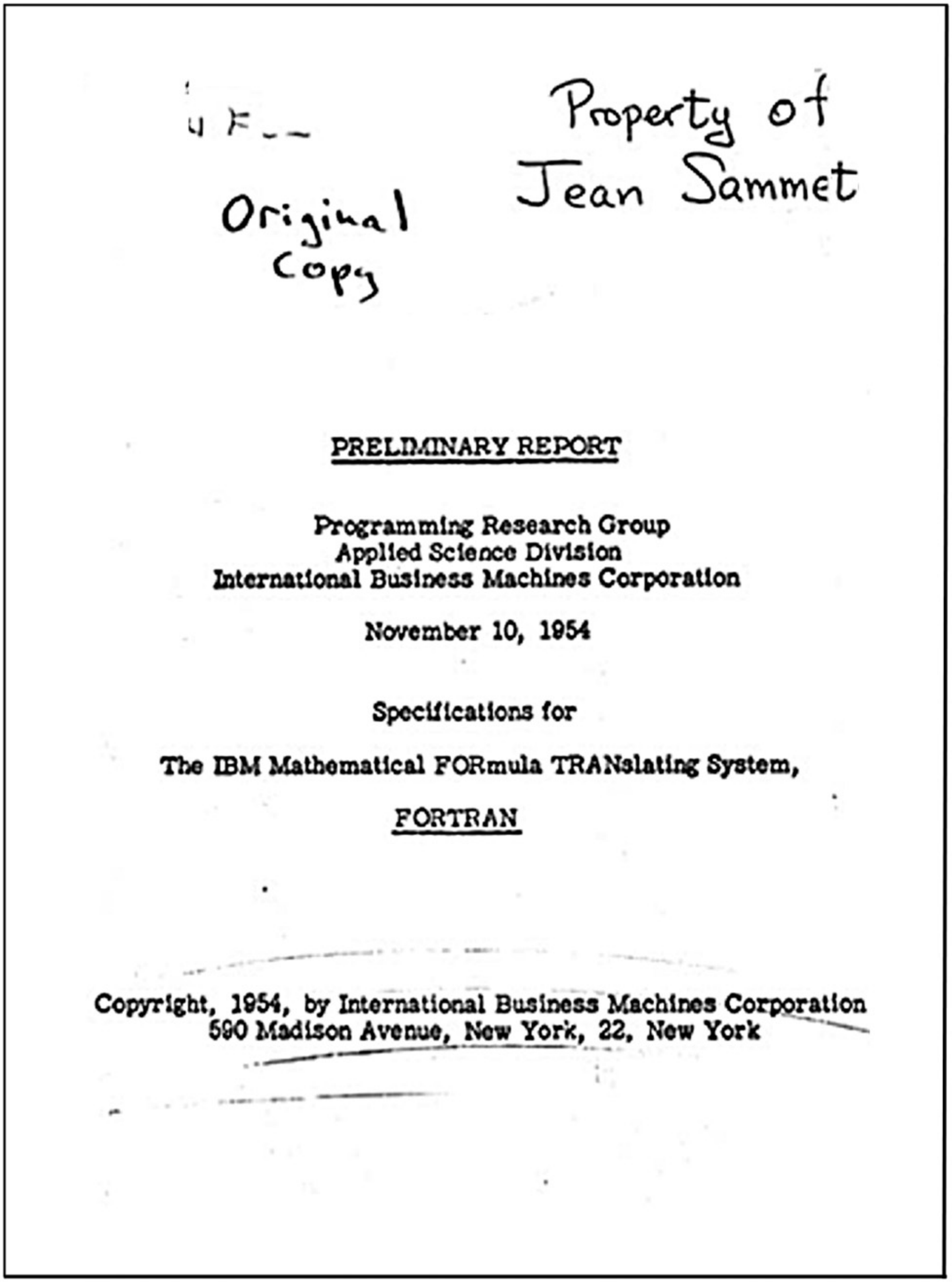
At the monthly SCC meeting in February 2004, my early-Fortran quest became the committee's first official test of collection and preservation efforts. In addition to the source code, I decided to start asking about the "Tome." The next person I contacted was Bob Bemer, who entered the computer business in 1949 and led the development of FORTRANSIT,[10] the second Fortran compiler, for the IBM 650. Bob's website included an article "Who Was Who in IBM's Programming Research? – Early FORTRAN Days"[11] that reproduced the IBM Programming Research Newsletter from January 1957 with short descriptions of the Fortran team members, including the sentence "Hal [Harold Stern] is currently working on 'TOME' describing FORTRAN internally." Bob told me he hadn't saved a copy of the "Tome." Sadly, Bob died four months later.

SCC member Dick Gabriel mentioned Paul Pierce's extensive private computer collection.[12] I contacted Paul. He didn't have a copy of the "Tome," but he had a number of useful suggestions for further research and volunteered to scan more of his SHARE and SOS (SHARE Operating System) materials.

Tom Van Vleck was at MIT in the 1960s, and I thought he might have heard of the "Tome."[13] He hadn't, but he suggested several leads for me to follow: Jean Sammet, Lynn Wheeler, and other IBM folks I'd worked with in the 1970s, Doug McIlroy, and Frank da Cruz, who maintains the Columbia University Computing History website.[14] McIlroy told me that after he joined Bell Labs in 1958, he used Fortran II, and the source code was available then. He also told me, "Another document that came with Fortran, which everybody got to know, was the 'stop book.' The compiler did not issue diagnostics. Instead it halted. The machine operator would record the instruction counter from the console lights. The stop book told what the cause of each stop was, often very cryptically: 'trouble in the tag table, or some other cause.'"

In May 2004, I came across the special issue of the *Annals of the History of Computing* dedicated to the 1984 National Computer Conference Pioneer Day celebration of "FORTRAN's Twenty-Fifth Anniversary," which contains a number of edited session transcripts, articles, anecdotes, and an annotated bibliography.[15]

Around this time, John Backus told me that he'd donated his personal papers to the Library of Congress, and he gave me a spreadsheet listing the donated items.[16] John also gave me extra copies of papers and photographs, including a photocopy of a 29-page memo,[17] "Preliminary Report: Specifications for the IBM Mathematical FORmula TRANslating System, FORTRAN," dated 10 November 1954 (see Figure 1).


4 F_ _

Property of Jean Sammet

Original Copy

PRELIMINARY REPORT

Programming Research Group
Applied Science Division
International Business Machines Corporation

November 10, 1954

Specifications for

The IBM Mathematical FORmula TRANslating System,

FORTRAN

Copyright, 1954, by International Business Machines Corporation
590 Madison Avenue, New York, 22, New York

Figure 1. Preliminary Report, 1954

J.A.N. Lee's annotated bibliography in the *Annals* special issue says the authors were probably John W. Backus, Harlan Herrick, and Irving Ziller. He notes, "This is the first formal proposal for the language FORTRAN. It lists the elements of the language that are proposed to be included in the eventual implementation, together with some suggestions for future extensions. It is interesting to match this proposal with the Programmer's Reference Manual[18] and to note that many of the ideas of later FORTRANs as well as Algol appear to have been given birth in this document."

Daniel N. Leeson's article in the same issue[19] mentioned that materials they'd used for the Pioneer Day event were from private collections, "two of which are unusually noteworthy": Roy Nutt and Harlan Herrick have both made a special effort to retain material from their early days in computing. Nutt possessed a microfilm of (allegedly) every document in the FORTRAN development offices at the time the product was released. He generously donated a copy to the IBM historical archives. Herrick's collection of memorabilia was also extensive. For example, he owned the only known copy of IBM's first FORTRAN film, made in Poughkeepsie about 1958, that showed how FORTRAN could be used to program a solution to 'The Indian Problem' (a calculation demonstrating the effect of compound interest on the $24 said to have been paid for Manhattan Island).

I tracked down Leeson and called him. He thought it was very unlikely that any copies of the source code survived. He used to run the organization that handled binary program distribution for IBM, and he said that IBM didn't even want to archive the binaries once a particular machine went out of production. Leeson also speculated that Roy Nutt's personal collection may have been lost when he died in 1990. Leeson later donated his copies of a Fortran marketing film released in 1957 and the retrospective video created by IBM for the Pioneer Day event.[20] I later talked to the IBM corporate archivist, who had recently inherited a vast but poorly cataloged archive. I never ended up getting access to any items in the IBM archive.

David Padua wrote an article[21] about the pioneering code optimization of the original Fortran compiler for a special issue of *Computing in Science and Engineering* on "the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century." I contacted David to see if, by any chance, he knew of someone with a copy of the source code. He didn't, but he and his colleague Sam Midkiff suggested contacting retired IBMers Fran Allen[22] or Marty Hopkins.

When I contacted Jean Sammet,[23] she also didn't know where to find the Fortran source, but suggested I contact Fran Allen and J.A.N. Lee.

J.A.N. Lee[24] responded to my email by saying, "I have asked about the original FTN compiler several times but without any success. Two sources would seem possible – John Backus himself … and the original recipient, Westinghouse-Bettis."[25] As it happened, I'd recently looked through several boxes of Fortran materials Lee had already donated to the Computer History Museum. They contained, among other things, photocopies of essentially all the papers included in his annotated Fortran bibliography.

First Results

During my search, I would periodically attempt to communicate with the Smithsonian. In June 2004, I was able to make contact with Alicia Cutler, a specialist in the Collection of Computers and Mathematics. She ran a search for me in their internal catalog, which produced a 12-page listing of documents related to Fortran. Three of them jumped out at me: three volumes totaling 1,321 pages, dated 1959, with the comment, "This is a FORTRAN program listing for the 704. Well documented." After further study, Alicia told me that these volumes relate to Fortran II, not Fortran, and they are accompanied by a letter from A.L. Harmon in May of 1959 stating that these are the SAP listings for the final Fortran II. This was a wonderful discovery: the Fortran II compiler had additional features, but it was essentially a superset of the original Fortran compiler code base. The question of how to get a copy of this listing would occupy me for the next 20 months.

One of the people Tom Van Vleck had mentioned was Frank da Cruz. From Frank's Columbia University Computing History website, I followed a link to the IBM 704 page of George A. Michael's website, "Stories of the Development of Large Scale Scientific Computing."[26,27] I sent him an email asking about the "Tome." He replied that although he was one of the early Fortran users, and Lawrence Livermore National Laboratory (LLNL) had four IBM 704s, he did not recall the "Tome." He noted that an LLNL programmer, Robert Hughes, had worked on the original Fortran project "on loan" from Livermore, and referred me to an oral history he had conducted with Hughes.[28] In that oral history, Hughes, who was one of the authors of the original 1956 IBM Fortran manual,[29] said: "I worked on what they called the 'first-level' documentation. And I made the biggest mistake of my life by not bringing a copy of that home. Now you understand why I missed making my first million dollars." So, apparently, he didn't bring a copy of the "Tome" back to LLNL. Hughes died in 2007 and Michael died in 2008.

I had learned from Alicia Cutler that Peter Zilahy Ingerman was the donor of the Fortran II listing at The Smithsonian.[30] I called Ingerman and had a very pleasant conversation. It turned out he'd donated the Fortran materials to The Smithsonian a number of years ago and did not remember the specific item I was interested in, but he volunteered to travel from his home in New Jersey to Washington if that would help.

In July 2004, I decided to try drawing attention to my search by creating an online journal called Dusty Decks. My initial post said, "I am using this weblog to discuss historic computer software and hardware, among other topics. For several months, I've been studying the early history of Fortran and trying to track down the source code for the original Fortran compiler. Although I just set up this weblog recently (June-July 2004), I've created backdated entries to document my quest in chronological order, starting here."[31]

As evidence for the power of the search engines, only three days after I published the blog, Micah Nutt, son of Fortran project participant Roy Nutt, commented on the post about Daniel N. Leeson. Micah refuted Leeson's speculation that Roy Nutt's personal collection may have been lost when he died: The family understood the significance of Roy Nutt's accomplishments and were carefully stewarding his papers. I followed up with Micah via email, and he told me, "The fiche I have is most likely the same as the set my father donated to the IBM library in 1982 for the 25th anniversary of FORTRAN. The contents appear to be the specifications, flowcharts, mathematical analysis and source code for FORTRAN along with (at least part of) the user's manual. The documents are a mix of handwritten and machine based (some with handwritten edits and notes)." Over the next months, Micah and I explored different approaches for having the microfiche digitized.

Another person who commented on the blog post was Leif Harcke, then a graduate student at Stanford as well as a knowledgeable practitioner of "retro-computing."[32] Leif explained, "Fortran II was a strange beast; it ran under the Fortran Monitor System (FMS). FMS could either run the machine stand-alone, or it could run under IBSYS. Fortran II was link-compatible with the FAP assembler, the IBM product which superseded UA-SAP. … I'm not sure the compiler itself will be of any use without the FMS monitor and the FAP assembler."

Around this time, Alicia Cutler at the Smithsonian suggested that the Fortran II listing could be loaned to the Computer History Museum, where we could scan it, and that process was initiated.

In October 2004, two months after I wrote a blog post about Micah Nutt and Leif Harcke, Bob Abeles commented, "The IBSYS tapes on Paul Pierce's site contain the source for FAP, FORTRAN II, FMS (version that ran under IBSYS), plus lots of other goodies." I followed up with Bob by email, who supplied me with a utility program he'd written for working with the tape images from Pierce. This was an amazing discovery: a machine-readable copy of the source code for a "superset" of the original Fortran compiler. Fortran II extended Fortran with separate compilation but preserved the original compiler's parsing, analysis, and code generation. In January 2005, Peter Capek, who had worked at IBM Research for many years and was friends with several of the original Fortran project members, told me about a document he'd come across in his files: "a detailed description of the FORTRAN compiler, dated in 1960, and explicitly distinguishing between the 704 and 709 versions, but covering both. … It's a couple of hundred pages, and describes each section of the compiler, including table structure, in considerable detail."

Peter kindly sent me a photocopy.[33]

Sharing with Others

By April 2005, I decided the items I'd located, including the Fortran II source code, the Preliminary Report, the Systems Manual, and a number of manuals from Al Kossow's Bitsavers collection,[34] justified creating a project website. The Computer History Museum had set up a webserver with a web-based content-management system for the use of the SCC, so it was simple to create and maintain a project website.[35] A week or so after I announced the website on the Dusty Decks blog, John Van Gardner posted a comment. He had been one of the IBM Customer Engineers at Lockheed Aircraft in Marietta, Georgia. In May 1956, he helped install IBM 704 serial number 13, and in April 1957 he helped get Fortran running on that machine. There was a problem that caused the compiler to print "Machine error" and halt. John noted, "To solve this problem we needed a source listing of the compiler. It took the Branch Manager several days to get one for us, and it was on a roll of 35 mm microfilm. This had to be under IBM control at all times, and when we finished the bug, it was sent back to the Branch Office." John also wrote a two-page anecdote about the resourceful techniques he used in 1957 to debug a hardware problem that resulted in the Fortran compiler behaving in a nondeterministic manner.[36]

Supporting a program as complex as the Fortran compiler was clearly challenging. By 1959, IBM had written the FORTRAN I, II, and 709 Customer Engineering Manual of Instruction, which contained many details about the internal structure of the compiler, the system tape, etc.[37] A copy of this manual was provided by Mark Halpern as part of a large donation to the Computer History Museum. Mark's first assignment after joining the IBM Programming Research Department in 1957 was to study and document (via flow-charts) the Fortran compiler. I got in touch with him after encountering his online memoirs.[38]

The original Fortran system had a variety of built-in library functions and allowed the programmer to write single-statement function statements or to add additional library functions written in assembler, but there were no separately compiled Fortran subroutines or functions. By August 2005, I had run across three versions of a 1957 memo (unsigned, but probably written by Irv Ziller), "Proposed Specifications for FORTRAN II for the 704," dated August 28, September 25, and November 18, that show the evolution of the design of the subroutine feature of FORTRAN II. Around this time, I got in touch with Dennis Hamilton, who had started his career in software right around the time the Fortran II compiler shipped. He had interesting observations about the impact of subroutines:

[T]he impact of small changes and improvements can be immense. The ability to build Fortran programs out of independently compilable modules and to have the ability to decompose into functions and subroutines using Fortran or any other tool that produced compatible code (usually the assembler, in those days) had an immense impact. In Fortran I, programs were one giant file, and there was no modularization structure. That small change in Fortran II was earthshaking in terms of software development and, I think, the endurance of Fortran as a technical-software programming tool.

It also changed the way that computers had to operate to make software building and use work more smoothly. I think it is no coincidence that this paralleled increased interest in operating systems (called things like tape monitors, at the time) and the use of the computer for organizing the data processing workflows. (There was also a lot of resistance to operating systems in those days.)

In February 2006 I finally obtained scans of the 704 Fortran II listing from the Smithsonian. The proposed intermuseum loan had been blocked because of a misunderstanding about an earlier loan of ENIAC modules to the Computer History Museum's predecessor in Boston. Instead, the Computer History Museum hired a researcher near the Smithsonian to scan the listing.[39]

Running Fortran Under Emulation

As soon as I began my quest for Fortran, I encountered people who were writing emulators.[40] Paul Pierce, who owned a real IBM 709[41] and had digitized the IBSYS tapes, had written an emulator for the IBM 709. Rob Storey wrote utilities to manipulate Pierce's tape images, and then wrote an IBM 7094 emulator. James Fehlinger took the lead in getting Fortran IV to run on Storey's emulator. The Fortran IV compiler does not share code with the IBM 704/709 Fortran/Fortran II compiler, but this was an impressive achievement in its own right.

Next, Dave Pitts wrote a cross-assembler and succeeded in assembling the source code for Pierce's copy of IBSYS into a bootable image. Using a modified version of Pierce's IBM 709 emulator, he was able to run IBSYS and the Fortran II compiler, but the object code generated by that compiler would not run.

Finally, Rich Cornwell succeeded in executing both the Fortran II compiler and the code it generated. Rich used a modified version of Bob Supnik's SimH emulator.[42] It was Rich's enthusiasm that inspired me to finally obtain the scanned copy of The Smithsonian's IBM 709 Fortran II compiler listing. Later, Rich, Fausto Saporito, James Markevitch, Bob Abeles, and Robert Cicconetti transcribed the source code from this listing to an assembly code file,[43] and Rich reassembled it using the SAP assembler running on an emulated IBM 7090 computer.[44]

Reflections

I failed to locate the original Fortran compiler: neither source code images (e.g., cards or magnetic tape) nor an assembly listing. However, I found two different versions of the Fortran II compiler source code, whose code generation and optimization sections were essentially identical to the original compiler. While searching for the source code, I encountered a rich variety of other primary and secondary materials, allowing me to create a website of interest to a broader audience than the dedicated researchers and computer scientists I was originally targeting. I was also fortunate to meet and participate in the retro-computing community, whose members write and run emulators, recreate source code from poor-quality listings, and pool their expertise in old hardware and software.

This project taught me that patience and persistence are necessary. Old software and related materials are in constant danger of being lost, damaged, or destroyed; the people who created that software are aging and dying. When contacted, people are usually friendly, but they are almost always busy. Multitasking was crucial, along with maintaining notes suggesting who had been contacted, what the response had been and an indication of when or how to follow up. My previous career in software gave me a pool of contacts to help me get started. The traditional literature research combined with Internet search is a synergistic combination. For example, I learned about the special FORTRAN 25th issue of *Annals* from an old USENET post, which led me to Daniel Leeson's article and J.A.N. Lee's annotated bibliography. I was then able to locate and call Leeson by using a search engine. And publicizing my search via the blog and the project website resulted in strangers contacting me with additional information. I have used these techniques with some success on several later projects.[45]

Acknowledgments

John Backus gave me an early taste of computer history by sharing stories of his early days.[46] J.A.N. Lee and Henry S. Tropp's work on the 1992 Pioneer Day and the 1984 special issue of *Annals* celebrating Fortran's 25th anniversary provided a sound foundation for all subsequent work on the history of Fortran. Grady Booch helped awaken an interest in software preservation. Thanks also to all the individuals and institutions named in this paper and the website for answering questions, providing documents, and getting old code to run again. Burt Grad encouraged me to submit this "after action review" to the *Annals*.

References and Notes

1. "Preserve Classic Software," Computer History Museum, 2003, https://web.archive.org/web/ 20080420173659/http://www.computerhistory.org/PreserveClassicSoftware/.

2. Now the Software Preservation Group; http://www.computerhistory.org/groups/spg/.

3. The compiler also served as the foundation for research on code optimization: F.E. Allen, “A Technological Review of the FORTRAN I Compiler,” *Proc. Nat’l. Computer Conf.*, 1982, pp. 805–809.

4. See, for example, J.A.N. Lee, “An Annotated Bibliography of FORTRAN,” *Annals of the History of Computing*, vol. 6, no. 1, 1984, pp. 49–58; doi:10.1109/MAHC.1984.10003.

5. The History of FORTRAN and FORTRAN II website contains design documents, source code, reference manuals, tutorials, papers, films, interviews, user stories, and other materials about this historic project. The Acknowledgements section of the website lists even more people who assisted my collection efforts than could be included in this paper.

6. See http://americanhistory.si.edu/comphist/.

7. I later learned they were in the final stage of mounting a major new exhibit: The Price of Freedom: Americans at War, http://americanhistory.si.edu/exhibitions/price-of-freedom .

8. For example, the compiler turned on a front panel indicator lamp once it determined there were no syntax errors.

9. J.W. Backus, “The History of FORTRAN I, II and III,” *Proc. of the First ACM SIGPLAN Conf. on the History of Programming Language*, 1978. Also published in R. Wexelblat, ed., *History of Programming Languages*, ACM Monograph Series, Academic Press, 1981. Reprinted in *Annals of the History of Computing*, vol. 1, no. 1, 1979, pp. 21–37, and vol. 20, no. 4, 1998, pp. 68–78.

10. See http://www.bobbemer.com/FTRANSIT. HTM.

11. See http://www.bobbemer.com/PRORES.HTM.

12. See http://www.piercefuller.com/collect/index. html.

13. Van Vleck is the creator and maintainer of the Multicians website (http://www.multicians. org/), which “presents the story of the Multics operating system for people interested in the system’s history.”

14. See http://www.columbia.edu/acis/history/; for example, http://www.columbia.edu/acis/ history/backus.html.

15. J.A.N. Lee and H.S. Tropp, eds., “Special issue: FORTRAN’s Twenty-Fifth Anniversary,” *Annals of the History of Computing*, vol. 6, no. 1, 1984.

16. See http://www.softwarepreservation.org/ projects/FORTRAN/Backus%20-%20LOC%20-%20catalogue%20of%20papers.pdf (there is now a formal – and less gossipy! – finding aid: John W. Backus papers, 1951–2001, Library of Congress Manuscript Division, Washington, D.C.; http://hdl.loc.gov/loc.mss/eadmss.ms007001).

17. http://www.softwarepreservation.org/projects/ FORTRAN/BackusEtAl-Preliminary%20Report* 1954.pdf.

18. J.W. Backus, et al. *The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Reference Manual*. Applied Science Division and Programming Research Department, IBM Corp., 1956, http://archive.computerhistory.org/ resources/text/Fortran/102649787.05.01.acc.pdf.

19. <oo> D. N. Leeson. "IBM FORTRAN Exhibit and Film," *Annals of the History of Computing*, vol. 6, no. 1, 1984, pp. 41–48.

20. Lot X2843.2005, Computer History Museum. See also http://www.softwarepreservation.org/ projects/FORTRAN/index.html#Films/video.

21. D. Padua, "The Fortran I compiler," *Computing in Science and Eng.*, vol. 2, no. 1, 2000, pp. 70–75; doi: 10.1109/5992.814661.

22. See F.E. Allen, "The History of Language Processor Technology in IBM," *IBM J. of Research and Development*, vol. 25, no. 5, 1981, pp. 535–548; F.E. Allen, "A Technological Review of the FORTRAN I Compiler," *Proc. Nat'l Computer Conf.*, 1982, pp. 805–809.

23. Sammet has published extensively on programming languages and their history. Her book, *Programming Languages: History and Fundamentals*, Prentice-Hall, Inc., 1969, has been called the definitive work on early computer language development.

24. Lee had successive careers in civil engineering and computer science and has been active in the history of computing for many years, for example, as co-editor of the previously mentioned special issue on FORTRAN's Twenty-Fifth Anniversary, and a previous editor-in-chief of the *Annals*.

25. H.S. Bright, "FORTRAN Comes to Westinghouse-Bettis," *Computers and Automation*, vol. 20, no. 11, 1971, pp. 17–18.

26. See http://www.computer-history.info and http://www.computer-history.info/Page4.dir/ pages/IBM.704.dir/index.html.

27. Michael was a computational physicist at Lawrence Livermore National Laboratory (LLNL) who worked in high-performance computing for most of his career.

28. G. A. Michael, "An Interview with Bob Hughes," May 9, 1997, http://www.computer-history. info/Page1.dir/pages/Hughes.html.

29. Backus, et al. *The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Reference Manual*.

30. Ingerman has published a number of books and papers in the area of programming languages and compilers.

31. The earliest post is http://www.mcjones.org/ dustydecks/archives/2003/11/19/11/, and the introductory post is http://www.mcjones.org/ dustydecks/archives/2004/07/04/2/.

32. See http://insar.stanford.edu/~lharcke/ programming/.

33. Anonymous, *Systems Manual for 704 FORTRAN and 709 FORTRAN*, Applied Programming Department, IBM Corp., April 1960, http:// archive.computerhistory.org/resources/text/ Fortran/102653992.05.01.acc.pdf.

34. See http://www.bitsavers.org/.

35. The current location is http://www. softwarepreservation.org/projects/FORTRAN.

36. J. Van Gardner. "Fortran And The Genesis of Project Intercept," 2005, http://www. softwarepreservation.org/projects/FORTRAN/ paper/John%20Van%20Gardner%20-%20 Fortran%20And%20The%20Genesis%20 Of%20Project%20Intercept.pdf.

37. Anonymous, *FORTRAN I, II, and 709: Customer Engineering Manual of Instruction*, IBM Corp., Form R23-9518-0, February 1959. Mark Halpern papers, Lot number X3762.2007, Computer History Museum, http://www. softwarepreservation.org/projects/FORTRAN/ R23-9518-0.pdf.

38. M. Halpern, "On the Heels of the Pioneers – A Memoir of the Not-Quite-Earliest Days of Programming," *Annals of the History of Computing*, vol. 13, no. 1, 1991, pp. 101–111.

39. Catalog number 102660000, Lot number X3435.2006, Computer History Museum. See http://www.computerhistory.org/collections/ catalog/102660000 and http://www. softwarepreservation.org/projects/FORTRAN/ index.html#Source_code.

40. See http://www.softwarepreservation.org/ projects/FORTRAN/index.html#Emulators.

41. Now catalog number 102728984 of the Paul Pierce Collection, Lot X7021.2014, Computer History Museum; see http://www. computerhistory.org/collections/catalog/ 102728984.

42. B. Supnick, "The Story of SimH," *Annals of the History of Computing*, vol. 37, no. 3, pp. 78–80. See also http://simh.trailing-edge.com/ and https://github.com/simh/simh.

43. See http://www.softwarepreservation.org/ projects/FORTRAN/index.html#Source_code.

44. For a detailed description of techniques developed by Markevitch and others to convert a scan of an old listing into an accurate source file, see D. Walden and the

IMP Software Guys, “The Arpanet IMP Program: Retrospective and Resurrection,” *Annals of the History of Computing*, vol. 36, no. 2, 2014, pp. 28–39.

45. See for example http://www. softwarepreservation.org/projects/LISP, http:// www.softwarepreservation.org/projects/ ALGOL, and http://www.softwarepreser-vation. org/projects/c plus plus.

46. John Backus died in March 2007, the year of Fortran’s 50th anniversary.

Paul McJones retired in 2009 from a career alternating between software research and product development. In retirement, he has been a Computer History Museum volunteer in the area of software preservation; see http://www.mcjones.org/paul/, which also includes contact information for him.

Computer History Museum – Software Presentation Group

History of FORTRAN and FORTRAN II

by Paul McJones – last modified 2014-08-12 16:22

Paul McJones, editor: paul@mcjones.org – http://www.mcjones.org/dustydecks/

Abstract:

The goal of this project is to preserve source code, design documents, and other materials concerning the original IBM 704 FORTRAN/FORTRAN II compiler. FORTRAN was the first high-level programming language and the first high-quality optimizing compiler. This is a project of the Computer History Museum's Software Preservation Group to develop expertise in the collection, preservation, and presentation of historic software. Comments, suggestions, and donations of additional materials are greatly appreciated.

Contents:

Acknowledgements

* Bob Abeles, Allison Akbay, Fran Allen, David K. Allison, Association for Computing Machinery (ACM), Bob Ayers, John Backus, John T. Bagwell Jr., Bob Bemer, Grady Booch, Nance L. Briscoe, Leonard Bruno, Sharon Brunzel, Peter Capek, Alicia Cutler, Frank da Cruz, John Van Gardner, Mark Halpern, Dennis E. Hamilton, Leif Harcke, Norm Hardy, Bob Hughes, Peter Zilahy Ingerman, Paula Jabloner, Jim King, Al Kossow, Paul Lasewicz, J.A.N. Lee, Daniel N. Leeson, Doug McIlroy, George A. Michael, Anel Rodriguez Muller, David Padua, Paul Pierce, Dave Pitts, Micah Nutt, Bernard L. Peuto, Roger Roach, Jean Sammet, Len

Shustek, Dick Sites, Dag Spicer, Rob Storey, Kirsten Tashev, Tom Van Vleck, Michael R. Williams, Irving Ziller.

Source code

* FORTRAN II compiler source listing. Smithsonian Institution-National Museum of American History, Catalog number 304,349. Originally donated by Peter Zilahy Ingerman. Assembly language (SAP) listing of 4K and 8K drum versions of the final IBM 704 FORTRAN II compiler. Three volumes. Pages I-III of Volume I are a table of contents. Digital scan is Computer History Museum Lot X3435.2006.

o Cover letter from A.L. Harmon to Peter Zilahy Ingerman, May 13, 1959. PDF

o Volume I. PDF (28.5 MB, 300dpi)

o Volume II. PDF (38.8 MB, 400dpi)

o Volume III. PDF (21.4 MB, 400dpi)

o For details about how this listing was discovered, see these Dusty Decks articles: The Smithsonian, The Smithsonian redux, As you sow so shall you reap, and 704 FORTRAN II listing available.

* Microfiche with “specifications, flowcharts, mathematical analysis and source code for FORTRAN”. Two sets, each consisting of 23 microfiche of size 5.8”x4.1” (14.7cm x 10.5cm), each microfiche containing 50 images of size .51”x.58” (1.3cm x 1.5cm). Property of the family of Roy Nutt.

o 300 dpi scan of microfiche #1: JPEG

o 2400 dpi scan of image 6 of microfiche #1: JPEG

o For background, see the Dusty Decks article As you sow so shall you reap.

* 32K 709/7090 FORTRAN II in executable and source form, in 7-track IBSYS distribution. Digitized by Paul Pierce.

o For an image of the original BCD-encoded 7-track magnetic tape digitized by Pierce, see: pr130-3.bcd

o For an ASCII transliteration, see the FORTRAN subdirectory of the archive prepared by Dave Pitts: ibsys.tar.gz at www.cozx.com ibsys.tar.gz at www.bitsavers.org

Here are the individual components; for the compiler proper, see 9F13 through 9F30:

* FORTRAN COMMON I/O PACKAGE-IOEX VERSION 9F00

* FORTRAN II MONITOR-TAPE POSITIONING RECORD 9F01

* FORTRAN II MONITOR-DUMP RECORD. 9F02

* 32K 709/7090 FORTRAN MONITOR-SIGN-ON RECORD. 9F03

* 32K 709/7090 FORTRAN ASSEMBLY PROGRAM-MACRO-FAP. 9F04

* 32K 709/7090 FORTRAN MONITOR-SCAN. 9F06

* 32K 709/7090 FORTRAN MONITOR-DEBUG, BSS CONTROL AND LIBRARY SEARCH. 9F07

* 32K 709/7090 FORTRAN MONITOR-MACHINE ERROR RECORD. 9F10

* 32K 709/7090 FORTRAN MONITOR-SOURCE ERROR RECORD. 9F11

* 32K 709/7090 FORTRAN MONITOR-DUMMY RECORD (RECORD 12). 9F12

* 32K 709/7090 FORTRAN-SECTION ONE, ONE PRIME, ONE DOUBLE PRIME (RECORDS 13-17). 9F13

* 32K 709/7090 FORTRAN-SECTION TWO. 9F18

* 32K 709/7090 FORTRAN-SECTION THREE. 9F22

* 32K 709/7090 FORTRAN-SECTIONS FOUR AND FIVE. 9F23

* 32K 709/7090 FORTRAN-SECTION SIX. 9F30

* 32K 709/7090 FORTRAN MONITOR-DEBUG, BSS CONTROL AND LIBRARY SEARCH. 9F32

* IBSFAP VERSION 3/FORTRAN IBSFAP MODE LOADER. IBSFAP

o For details about how these tape images were discovered and progress executing them on a simulator, see these Dusty Decks articles: Fortran II source in Paul Pierce's collection and Dave Pitts is making progress running Fortran II.

* FORTRAN II compiler source listing belonging to John T. Bagwell Jr. This version was modified to use 7 index registers and ran under the 7090/7094 Porthos operating system at the University of Illinois 7090/94. Porthos was derived from the University of Michigan operating system and supported a hard disk.

o For details about how this listing was discovered, see this comment at Dusty Decks.

* John Van Gardner, who was an IBM Customer Engineer at Lockheed Aircraft in Marietta, Georgia, when FORTRAN was shipped in April 1957 notes that they were able to obtain a source listing of the compiler on 35mm microfilm in order to debug a hardware problem. Norm Hardy, who was at Livermore, has also mentioned the microfilm images of listings distributed by IBM. See Memoirs and user stories below. [Personal communication from John Van Gardner to Paul McJones, May 2005.]

* Matrix Inversion Order 5. FORTRAN source program, generated assembly program, output from program execution, compiler listing summary page. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. Approx. 1958? PDF

Documentation

Author names in [brackets] are not listed in the document itself, but are "as remembered" according to Lee 1984.

* [J.W. Backus, H. Herrick and I. Ziller.] Preliminary Report: Specifications for the IBM Mathematical FORmula TRANSlating System, FORTRAN. Programming Research Group, Applied Science Division, International Business Machines Corporation, November 10, 1954, 29 pages.

o Copy 1: Photocopy with "Property of Jean Sammet" and "Original Copy" on first page; missing page numbered 12. Given to Paul McJones by John Backus. PDF

o Copy 2: Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

"This is the first formal proposal for the language FORTRAN and lists the elements of the language which are proposed to be included in the eventual implementation together with some suggestions for future extensions. It is interesting to match this proposal with the Programmer's Reference Manual published in 1957 (below) and to note that many of the ideas of later FORTRANs as well as ALGOL appear to have been given birth in this document." [Lee 1984]

"The FORTRAN 0 document represents the first attempt to define the syntax of a programming language rigorously. Backus's important notation, which eventually became `BNF' can be seen in embryonic form here." [Knuth and Trabb Pardo 1977]

* Anonymous. IBM 704 Electronic Data Processing Machine Manual of Operation. IBM Corporation, 1954-1955, Form 24-6661-2. PDF

This manual describes the machine on which FORTRAN ran: a memory of 4096, 8192, or 32768 words of 36 bits each; a register complement including an accumulator, a multiplier-quotient, and 3 index registers; integer and floating-point arithmetic; and programmed I/O (no I/O channels or interrupts) with magnetic tape units, magnetic drums, punched card readers and punches, and line printers.

* J.W. Backus, R.J. Beeber, S. Best, R. Goldberg, H.L. Herrick, R.A. Hughes, L.B. Mitchell, R.A. Nelson, R. Nutt, D. Sayre, P.B. Sheridan, H. Stern, I. Ziller. The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Reference Manual. Applied Science Division and Programming Research Department, International Business Machines Corporation, October 15, 1956, 51 pages. From bitsavers.org. PDF. Apparently assigned Form number 32-7026 after printing, as referred to on the last page of the Primer.

* Anonymous. Addenda to the FORTRAN Programmer's Reference Manual. Reproduced typescript, undated, 7 pages. Computer History Museum, gift of Dick Sites, accession number 102655104. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005. PDF

"For 'late 1956' read 'early 1957'." Describes FUNCTION statement.

* [Grace E. Mitchell?] FORTRAN Introductory Programmer's Manual. Programming Research Department, International Business Machines Corporation, 590

Madison Avenue, New York, New York. Reproduced typescript. Computer History Museum, gift of Dick Sites, accession number 102655104. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005.

o Section I. March 20, 1957, 1+37 pages. PDF

o Section II. April 10, 1957,1+ 31 pages. Page 12 was blank. PDF

o Section III. June 7, 1957, 1+28 pages. PDF

* [Grace E. Mitchell.] The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer's Primer. International Business Machines Corporation, 1957, Form 32-0306, 64 pages. PDF

* [Grace E. Mitchell.] General Information Manual: Programmer's Primer for FORTRAN Automatic Coding System for the IBM 704. IBM Corporation, New York, 1957, Form F28-6019, 64 pages. From bitsavers.org. PDF

"This edition is a reprint of Form Number 32-0306-1 and does not obsolete it or 32-0306. No changes have been made to 32-0306-1."

* Anonymous. The FORTRAN Automatic Coding System for the IBM 704 EDPM: Preliminary Operator's Manual. Programming Research Department, International Business Machines Corporation, April 8, 1957.

o Copy 1. 6+31 pages. Computer History Museum, gift of Dick Sites, accession number 102655104. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005. PDF

o Copy 2: 6+7 pages. Photocopy with "I. Ziller" written on first page. Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

"This manual describes the use of FORTRAN 4-1-4-1." The 704 did not have an operating system; FORTRAN ran on the bare machine. This manual has sections Preparing the System Tape, Using the System Tape, Error Detection, Running the Object Program, Error Stops in Object Programs, Maintaining the Library Functions, and then a 7-page list of compiler error stops.

* J. W. Backus. Letter to Greenstadt. [FORTRAN Letter #1?]. April 24, 1957. 4 pages. Computer History Museum, gift of Dick Sites, accession number 102655104. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005. PDF

Describes materials sent to SHARE members owning IBM 704s with initial FORTRAN release.

* J. W. Backus. Letter to Ross: FORTRAN Letter #2. May 7, 1957. 3 pages. Computer History Museum, gift of Dick Sites, accession number 102655104. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005. PDF

Describes upgrade for IBM 704s with 8192 words of memory; includes writeup of FNEDT 1 program for editing FORTRAN system tapes.

* International Business Machines Corporation and R.A. Hughes and R. Nutt. FORTRAN Automatic Coding System for the IBM 704 Data Processing System: Reference Manual. Form C28-6003, October 1958, 56 pages. From bitsavers.org. PDF

This is very similar to the October 1956 manual, but includes the description of the FUNCTION statement (see page 17), which was initially documented in an addendum (see above).

* Anonymous. FORTRAN Programmation Automatique de L'Ordinateur IBM 704: Manuel du Programmeur. IBM France, Institut de Calcul Scientifique, Paris. No date, 51 pages. Given to Paul McJones by John Backus. PDF

* Anonymous. FORTRAN II for the IBM 704 Data Processing System: Reference Manual. Form C28-6000-2, 1958, 63 pages. PDF

This manual is a supplement to the original FORTRAN Reference Manual and Primer, describing the new features, including the CALL, SUBROUTINE, FUNCTION, COMMON, and END statements and the new Binary Symbolic Subroutine Loader.

* Anonymous. The FORTRAN II Automatic Coding System for the IBM 704: Operations Manual. No date, 62 pages. Scanned from photocopy loaned by Roger Roach. PDF

* Various authors. Miscellaneous FORTRAN memos, 1958-1959. From bitsavers.org. PDF

o Anonymous. 704 FORTRAN II: Error Halt List for BSS Loader. November 28, 1958, 1 page.

o Anonymous. FORTRAN Subprogram Types [with calling sequences]. 2 pages.

o Anonymous. Memorandum for All 704 Users: FORTRAN Library and END Cards. Poughkeepsie South Road Laboratory, Department 535, November 26, 1958.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [PE RAND pseudo random number generator]. Poughkeepsie South Road Laboratory, Department 535, October 28, 1958, 2 pages.

o H.S. Long and L.O. Nippe. Memorandum for All 704 Users: FORTRAN Library Addition [PK CERF complementary error function]. Poughkeepsie South Road Laboratory, Department 535, November 14, 1958, 1 page.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [PK LGAM natural logarithm of the Gamma Function]. Poughkeepsie South Road Laboratory, Department 535, October 28, 1958, 1 page.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [PE OVFL reset or test overflow triggers]. Poughkeepsie South Road Laboratory, Department 535, November 25, 1958, 1 page.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [EL SAVE 1 and 2 save and restore cores and drums]. Poughkeepsie South Road Laboratory, Department 535, October 28, 1958, 1 page.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [LA S885 to solve the matrix equation AX = B for X and to evaluate the determinant A]. Poughkeepsie South Road Laboratory, Department 535, October 28, 1958, 1 page.

o L.O. Nippe. Memorandum for All 704 Users: FORTRAN Program Library Addition [NU BES1 Bessel functions for real argument and order]. Poughkeepsie South Road Laboratory, Department 535, October 28, 1958, 2 pages.

o Anonymous. Master IBM CARD Layouts: FORTRAN relocatable and absolute binary instruction or data cards. 2 pages.

o Anonymous. FORTRAN Modifications/Boolean Expressions, Function and Subroutine Names as Arguments, Additional Format Features. IBM Applied Programming, November 1959, 6 pages.

o Anonymous. FORTRAN Modifications/Machine Language Instructions in FORTRAN Programs, IBM Applied Programming, January 1960, 12 pages.

o L.O. Nippe. FORTRAN Program Cards [control card for SAP relocatable binary deck]. Department 535, October 19, 1959, 1 page.

* [I. Ziller]. Description of Source Language Additions to the FORTRAN II System. Programming Research, International Business Machines Corporation, 1958?, 1+12 pages. Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

"This is the document which proposes the extensions to FORTRAN II to create FORTRAN III which was a very short-lived system. The fundamental addition was the allowance of symbolic statements intermixed with FORTRAN statements." [Lee 1984]

* Grace E. Mitchell, "The 704 FORTRAN II Automatic Coding System. Research Report RC-136, IBM Research Center, Yorktown Heights, New York, September 4, 1959, 13 pages. Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

"Abstract: This paper discusses the addition made in the FORTRAN I translator to produce the FORTRAN II translator. The new source language statements, debugging facilities and loader are described."

* Anonymous. IBM 709 Data Processing System Reference Manual followed by 709 FORTRAN Automatic Coding System. Cover, 24-72, 121-124, 129-166, 182 pages, no date. From bitsavers.org. PDF

A combination of sections of a hardware reference manual and a FORTRAN II reference manual. Perhaps this was assembled from separate manuals for distribution to students learning FORTRAN?

* Anonymous. FORTRAN I, II, and 709: Customer Engineering Manual of Instruction. IBM Corporation, Form R239518-0, February 1959, 67 pages. Computer History Museum Lot X3762.2007. Donated by Mark Halpern. PDF

Includes overview of the structure of the compiler, many details of the FORTRAN systems tape, and the record structure of the compiler intermediate tape. Section 3 is reprint of Backus et al. 1957.

* Anonymous. Systems Manual for 704 FORTRAN and 709 FORTRAN. Applied Programming Department, International Business Machines Corporation, April 1960, 264 pages. Copy belonging to Peter Capek. PDF

"Prefatory Note: This manual is an attempt to fulfill a long standing, much-pressed request. That is, a request for an over-all, comprehensive explanation of the workings of the entire Fortran System. This includes, in addition to the compiler proper, the monitor, the editor programs, and other corollary routines. ..."

* Anonymous. 709/7090 32K FORTRAN: Programming Systems Analysis Guide. IBM Corporation, Form R23-9673, 1961, 134 pages. Computer History Museum Lot X3762.2007. Donated by Mark Halpern. PDF

This appears to be an evolution of the 1960 Systems Manual for 704 FORTRAN and 709 FORTRAN.

* Anonymous. FORTRAN II General Information Manual. IBM Systems Reference Library. Form F28-8074-3, December 1963, viii+103 pages. From bitsavers.org. PDF

* Anonymous. FORTRAN II reference manual.

o IBM 709/7090 FORTRAN Programming System: Reference Manual. Form C28-6054-2, January 1961, 105 pages. Original belonging to Bob Ayers. PDF

o IBM 7090/7094 Programming Systems: FORTRAN II Programming. IBM Systems Reference Library. Form C28-6054-4, August 1963, 51 pages. From bitsavers.org. PDF

o IBM 7090/7094 Programming Systems: FORTRAN II Programming. IBM Systems Reference Library. Form C28-6054-5, April 1964, 51 pages. From bitsavers.org. PDF

* Donald P. Moore. FORTRAN ASSEMBLY PROGRAM (FAP) for the IBM 709/7090.

IBM 709/7090 Data Processing System Bulletin. Form J28-6098-1, 1961, 90 pages including addenda. Original belongs to Bob Ayers. PDF

* [Donald P. Moore?] FORTRAN ASSEMBLY PROGRAM (FAP) for the IBM 709/7090:

Supplementary Information for the 32K Version: Preliminary Bulletin. IBM 709/7090 Data Processing System Bulletin. Form J28-6186, circa 1961, 50 pages. Computer History Museum Lot X3762.2007. Donated by Mark Halpern. PDF

* Anonymous. IBM 7090/7094 Programming Systems: FORTRAN II Assembly Program (FAP). IBM Systems Reference Library.

o Form C28-6235-2, October 1963. 75 pages. Followed by Technical Newsletter N28-0084 (addenda and errata to C28-6235-2), November 29, 1963. 18 pages. From bitsavers.org. PDF

o Form C28-6235-3, April 1964. 75 pages. From bitsavers.org. PDF o Form C28-6235-5, April 7, 1965, 75 pages. PDF

* Anonymous. IBM 7090/7094 Programming Systems: FORTRAN II Operations. IBM Systems Reference Library. Form C28-6066-6, 1963. From bitsavers.org. PDF

"This publication contains a description of the FORTRAN II Processor operating under the FORTRAN II Monitor (Part 1 of this publication) and of the FORTRAN II Processor operating under IBSYS (Part 2 of this publication.)"

* Anonymous. IBM 7090/7094 Programming System: FORTRAN II Input/Output Package. IBM Systems Reference Library. Form J28-6190-3, October 16, 1964, 24 pages. PDF

* Anonymous. IBM 7090/7094 IBSYS Operating System: System Monitor (IBSYS). IBM Systems Reference Library.

o Version 11. Form C28-6248-1, February 1964. 77 pages. From bitsavers.org. PDF

o Version 13. Form C28-6248-7, December 30, 1966. 93 pages. Followed by Technical Newsletter N28-0242 (addenda and errrata to C28-6248-7), 3 pages. From bitsavers.org. PDF

<u>Correspondence and memoranda</u>

* John W. Backus Papers, Manuscript Division, Library of Congress, Washington, D.C. LC Control Number mm2003084968.

o Finding aid available in the Library of Congress Manuscript Reading Room. DRAFT version. PDF

o John Backus. Items for the Library of Congress. 13 pages, 2003. PDF

"Summary: Correspondence, memoranda, speeches, writings, reports, notes, slides, photographs, and other papers relating to Backus's work as a computer scientist at IBM on programming languages, particularly FORTRAN. Includes material relating to Backus's early work on the IBM Selective Sequence Electronic Calculator and Backus-Naur Form (BNF). Documents Backus's pursuit as IBM Fellow, 1963-1991, of his own research projects relating to mathematical theories of programming and the development of functional programming languages. Also includes files pertaining to Backus's political activism as a member of Computer

Professionals Against ABM in the early 1970s and as an opponent of the strategic defense initiative in the 1980s."

* "TWENTY-FIVE YEARS OF FORTRAN" (EXHIBIT, 1957-82). INTERNATIONAL BUSINESS MACHINES CORPORATION. Manuscript number Ms83-003, Special Collections Department, University Libraries, Virginia Polytechnic Institute and State University. Directory

"5.0 cu. ft., 10 – 3'x8' panels and 1 videotape. Pioneer Day was celebrated on June 9, 1982, at the National Computer Conference in honor of the 25th anniversary of the delivery of the first FORTRAN compiler. As part of the celebration IBM created and displayed this exhibit. Contains photographs of FORTRAN pioneers, facsimiles of documents, textual analysis, flow charts, memorabilia, FORTRAN manuals and other publications, and a twelve-minute videotape on the history of FORTRAN starring the members of the original FORTRAN development team: John Backus, Sheldon Best, Richard Goldberg, Lois

Mitchell Haibt, Harlan Herrick, Grace Mitchell, Robert Nelson, Roy Nutt, David Sayre, Peter Sheridan, and Irving Ziller."

* Harwood G. Kolsky. Notes from "Inside Fortran" presentation by Bill Heising. July 7, 1958. Handwritten notes, 5 pages. Computer History Museum accession number 102635018. Donated by Harwood G. Kolsky. PDF

* B.G. Oldfield. FORTRAN Experience at the New York Data Processing Center. Presented to SHARE meeting, Dallas, Texas, July 1957. Includes cover letter from Oldfield to John Backus, July 30, 1957. 1+3 pages. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

* [Irv Ziller?] Preliminary Report: Proposed Specifications for FORTRAN II for the 704. August 28, 1957, 3 pages. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

In this version a subroutine call begins with the name of the procedure followed by the parenthesized argument list – there is no reserved word CALL. A subroutine begins with a SUB DEF statement. There is a RETURN statement but no END statement.

* [Irv Ziller.] Proposed Specifications for FORTRAN II for the 704. Programming Research Department, International Business Machines Corporation, September 25, 1957, 1+7 pages. Ditto (spirit duplicator). Given to Paul McJones by John Backus. PDF

In this version a subroutine begins with a SUBROUTINE DEFINITION statement, and there is still no reserved word CALL. An UPPER statement is described, with the same semantics as the COMMON statement introduced in the November 18 version. The END statement appears.

* [Irv Ziller.] Proposed Specifications for FORTRAN II for the 704. Programming Research Department, International Business Machines Corporation, November 18, 1957, 1+7 pages. Ditto (spirit duplicator) with pen and ink corrections. Given to Paul McJones by John Backus. PDF

In this version a subroutine begins with a SUBROUTINE statement and a subroutine call begins with the reserved word CALL. The UPPER statement becomes the COMMON statement.

* J.W. Backus. Present Status of FORTRAN: Memorandum to Mr. J.C. McPherson. Programming Research, September 1, 1957, 1 page. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

* John C. McPherson. Memorandum to Mr. J.W. Backus. WHQ, September 23, 1957, 1 page. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

* John C. McPherson. Memorandum: FORTRAN II. November 1, 1957, 1 page. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

* J.W. Backus. Memorandum to Mr. J.C. McPherson: FORTRAN II. Programming Research, November 5, 1957, 1 page. Computer History Museum Lot X3853.2007, Box 15, Programming Presentation Papers folder, McPherson/Polivka donation. PDF

<u>Papers and lecture notes: Precursors of FORTRAN</u>

* A.C. Glennie. Automatic Coding of an electronic computer. Photocopy of typewritten manuscript with handwritten corrections, 15 pages. First page has date “14/12/52” and handwritten annotation “Lecture was delivered at University of Cambridge mid-February 1953”. Computer History Museum Lot X2677.2004, Box 3 of 6, blue 3-inch binder. Donated by J.A.N. Lee. PDF

* J.H. Laning and N. Zierler. A Program for Translation of Mathematical Equations for Whirlwind I. Engineering Memorandum E-364, Instrumentation Laboratory, Massachusetts Institute of Technology.

  o Copy 1: Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

  o Copy 2: Alternative version in Rich Katz’s NASA Goddard Site. PDF

“The Laning and Zierler system was quite a different story: it was the world’s first operating algebraic compiler, a rather elegant but simple one. Knuth and Trabb Pardo (1977) assign this honor to Alick Glennie’s AUTOCODE, but I, for one, am unable to recognize the sample AUTOCODE program they give as “algebraic”, especially when it is compared to the corresponding Laning and Zierler program.” [Backus, The history of FORTRAN I, II and III]

* John W. Backus. The IBM 701 Speedcoding System. Journal of the ACM, Volume 1, Number 1 (January 1954), pages 4-6. [Posted here by permission of ACM] PDF ACM Digital Library

"Speedcoding is a floating point three-address system which greatly simplifies programming, and checking out a program. Speedcoding provides convenient input-output operations, built-in checking, easy loading and printing. Therefore, Speedcoding reduces programming and testing expenses considerably. These expenses are often a large part of the cost of operating a computing installation. Thus Speedcoding is economical as well as convenient to use."

* J.W. Backus and H. Herrick. IBM 701 Speedcoding and other automatic programming systems. In Proc. Symp. on Automatic Programming for Digital Computer, Washington DC, The Office of Naval Research, May 1954, pp. 106-113. Computer History Museum Lot X2677.2004, Box 3 of 6, black 3-inch binder. Donated by J.A.N. Lee. PDF

<u>By FORTRAN project members</u>

* J.W. Backus, R.J. Beeber, S. Best, R. Goldberg, L.M. Haibt, H.L. Herrick, R.A. Nelson,

D. Sayre, P.B. Sheridan, H.J. Stern, I. Ziller, R.A. Hughes, and R. Nutt, The FORTRAN automatic coding system. Pages 188-198. In Proceedings Western Joint Computer Conference, Los Angeles, California, February 1957.

o Copy 1: Preprint, 50 pages. Acquired by Al Kossow on eBay. PDF preprint at bitsavers.org

o Copy 2: Identical to Copy 1? Computer History Museum Lot X3143.2005. Donated by Dick Sites. Part of Lot 58 of The Origins of Cyberspace auction, Christie's, February 23, 2005.

o Copy 3: Typeset reprint in original blue cover. Given to Paul McJones by John Backus. [Posted here by permission of ACM] PDF

"This is the first formal paper on the completed FORTRAN implementation reporting both on the language and on the design of the compiler. It includes an overview of the language, a description of each stage of the processor (with attributions of responsibilities amongst the authors), and conclusions regarding the success of the project. The final statement is worthy of repeating: "... the intellectual satisfaction of having formulated and solved some difficult problems of translation and the knowledge and experience acquired in the process are themselves almost a sufficient reward for the long effort expended . . . " [Lee 1984]

* J.W. Backus. Automatic programming: properties and performance of FORTRAN systems I and II. Session 2, Paper 3, Proceedings Symposium on the Mechanisation of Thought Processes, Teddington, Middlesex, England, The National Physical Laboratory, Nov. 1958, Her Majesty's Stationary Office (HMSO), pp. 232-255. PDF

“Written at a distance of one year after the delivery of the first FORTRAN processor for the 704, this paper is significant in its presentation of FORTRAN as an ‘automatic programming system’ in the environment of a symposium of the mechanisation of thought processes. Other attendees at the meeting included Jan Garwick (Norway), John McCarthy (USA), Grace Murray Hopper (USA) and Christopher Strachey (GB), each of whom commented on the presentation by Backus. Obviously Garwick was much more interested in telling the audience of developments by Ole Johan Dahl while McCarthy (the author of LISP) praised FORTRAN for its ability to express ‘... quite lengthy algebraic expressions ...’ and the implementation of separate compilation of subroutines (presumably in FORTRAN II). Hopper states that ‘... there is a lack of understanding of the systemizsation [sic] of FORTRAN’ and asks Backus to emphasize that Fortran does more than just the ‘housekeeping’ for the programmer.” [Lee 1984]

* P.B. Sheridan. The arithmetic translator-compiler of the IBM FORTRAN automatic coding system. Comm. ACM, Volume 2, Number 2, February 1959, pages 9-21. [Posted here by permission of ACM] PDF ACM Digital Library

[From the introduction:] The present paper describes, in formal terms, the steps in translation employed by the FORTRAN arithmetic translator in converting FORTRAN formulas into 704 assembly code. The steps are described in about the order in which they are actually taken during translation.”

* J.W. Backus and W.P. Heising. FORTRAN. IEEE Transactions on Electronic Computers, EC-13, Number 4, August 1964, pages 382-385. PDF

[From the summary:] The fundamental concepts of FORTRAN, the most widely used high-level, scientific programming language, are set forth and the significant characteristics are described in historical order from inception ... in 1954 to [1964] ... The basic problem of how to get high quality programming from an easy-to-write high-level language is emphasized. “Looking back after 10 years, Backus recalls the objectives of the FORTRAN effort and many of the frustrations which accompanied the development of the first processor. Of particular note is the commentary on the time taken during compilation to ensure the production of optimum code, time which is often fruitlessly wasted on simple programs. Mention is made of the technique of flow analysis used in the first compiler which was based on a Monte Carlo analysis of the frequency of execution of sections of the program. Regrettably (then and ever since) no documentation of this technique is provided.” [Lee 1984]

* J. W. Backus, The history of FORTRAN I, II and III. Proceedings First ACM SIGPLAN conference on History of programming languages, Los Angles, 1978. [Posted here by permission of ACM]

o Preprint. PDF ACM Digital Library

o Final version, in: R. Wexelblat, editor. History of Programming Languages, ACM Monograph Series, Academic Press, 1981.

* Paper. Pages 25-45. PDF ACM Digital Library

* Transcript of presentation. Pages 45-66. PDF ACM Digital Library

* Transcript of discussant's remarks. Pages 66-68. PDF ACM Digital Library

* Transcript of question and answer session. Pages 68-71. PDF ACM Digital Library

* Full text of all questions submitted. Pages 71-73. PDF ACM Digital Library

* Biography of John Backus. Page 74. PDF ACM Digital Library

o Reprinted in Annals History of Computing, Volume 1, Number 1, July 1979, pages 21- 37 and in Volume 20, Number 4, October-December 1998 IEEE CS Digital Library

"This article summarizes the history of the development of FORTRAN I, II and III. The author, who was the leader of the groups which developed the first two compilers, explains the economic factors leading to the establishment of the FORTRAN project, its goals, and the mode of working of its implementations. The article makes it clear that the early FORTRAN efforts were efforts of compiler development rather than language design. The language was designed as the compiler was written and the compiler design was considered [to be] the hard job. This lucidly written article is interesting not only for the facts presented about the history of FORTRAN (e.g., that efficiency of object code was more important in getting FORTRAN accepted than the design of the language) and the insight given into design of the language (e.g., that subscripts in a subscript variable were limited to three to increase compiler efficiency rather than because the IBM 704 has only three index registers), but also for its revelation of the mixture of clairvoyance, inventiveness, and naivete possessed by the implementation team (e.g., common expression elimination, the actual degree of optimization exhibited in the object code, and the feeling that debugging would all but [be] eliminated by the use of FORTRAN). The article is must reading for anyone considering language design today; it raises serious questions as to whether there is anything new under the sun. The kinds of things being said today are hauntingly reminiscent of the kinds of things quoted in the article as being said in 'those' days." [D. Berry, Los Angeles CA, Computing Review 35,907]

* J. W. Backus. Programming in the 1950's – some personal impressions. In A History of Computing in the Twentieth Century, N. Metropolis et al, Eds., Academic Press, New York, 1980, pages 125-135. Preprint. PDF

"Backus has been known to suggest that as much as anything else he would like to forget FORTRAN, and in this paper he almost does just that. The paper represents a personal view of the world as it existed prior to the FORTRAN development. Following a short presentation on the actual production of the first FORTRAN compiler (including the admission that he had been wrong in ascribing the early notions of algebraic input to Laning and Zierler in previous presentations) Backus shows how the concepts introduced by FORTRAN fulfilled the goals of the period.” [J.A.N. Lee, Annals of the History of Computing, 1981.]

By others

* John Cocke and J.T. Schwartz, Programming Languages and their Compilers, Preliminary Notes. 2nd revised edition, Courant Institute, New York, April 1970, pages 510-515. PDF

“These five pages in the otherwise unpublished manuscript contain a review of the techniques of optimization that were used in the original FORTRAN compiler, and based on assistance from Sheldon Best, are a more detailed account of the processes used than were published previously.” [Lee 1984]

* H.S. Bright. FORTRAN comes to Westinghouse-Bettis. Computers and Automation,

Volume 20, Number 11, 1971, pages 17-18. PDF

“This is an anecdote regarding a strange and wonderful package that arrived unannounced at Westinghouse-Bettis and which turned out to be a binary deck of the original FORTRAN processor for the IBM 704. Included is a copy of the first program run and the output (including the first error message.) There may be one erroneous report in this anecdote, that is that April 20, 1957 is said to be a Friday (in the first sentence); the 20th was in fact a Saturday.” [Lee 1984]

The PDF includes a letter from Bright to J.A.N. Lee dated February 8, 1978 and a letter from Bright to John Backus dated February 2, 1978. Scanned from original in Computer History Museum Lot X2677.2004, Box 3 of 6. Donated by J.A.N. Lee.

* Donald Ervin Knuth and Luis Trabb Pardo. The early development of programming languages. Pages 419-493 in J. Belzer, A. G. Holzman, A. Kent, editors. Encyclopedia of Computer Science and Technology, Marcel Dekker, Inc., New York. 1977. Republished in: Donald Knuth. Selected Papers on Computer Languages, CSLI, Stanford, 2003.

o Originally released as Report STAN-CS-76-562, Computer Science Department, Stanford University. PDF at bitsavers.org

o Basis for lecture: “A Dozen Precursors of Fortran”. Video, Computer History Museum, December 03, 2003. Online at CHM

* F.E. Allen. The History of Language Processor Technology in IBM. IBM Journal of Research & Development, Volume 25, Number 5 (September 1981), pages 535-548. PDF at IBM

The section “Early history” describes pre-FORTRAN programming: machine language, subroutine libraries, symbolic operation codes, relative addressing for instructions and data, labels, macros, and interpreters for virtual machines with floating-point and index registers such as Backus’s Speedcode. The section “FORTRAN I” describes the compiler structure in some detail and concludes: “The real results of the project are the influences it had on future compilers and theory. Some of these effects have already been mentioned; more will be discussed later. Suffice it to say that the technological fallout from this project has been extensive."

* F.E. Allen. A Technological Review of the FORTRAN I Compiler. Proceedings National Computer Conference, 1982, AFIPS Press, pages 805-809. [Posted here by permission of ACM] PDF

"In some cases, it produced code which was so good that users thought it was wrong, since it bore no obvious relationship to the source. It set a standard for object program efficiency that has rarely been equalled. The FORTRAN I compiler, completed in 1957, established modern compiler tasks, structure, and techniques."

* J.A.N. Lee and Henry Tropp. FORTRAN and its Progeny (unpublished book). Computer History Museum Lot X2677.2004, Box 3 of 6. Donated by J.A.N. Lee.

* J.A.N. Lee and H.S. Tropp, editors. FORTRAN's Twenty-Fifth Anniversary. Special issue of the Annals of the History of Computing, Volume 6, Number 1 (January 1984). IEEE Digital Library

o John Backus, session chair. Early Days of FORTRAN.

* J.A.N. Lee. Pioneer Day, 1982. Pages 7-14.

* John C. McPherson. Early Computers and Computing Institutions. Pages 15-16.

* Robert W. Bemer. Computing Prior to FORTRAN. Pages 16-18.

* Richard Goldberg. Register Allocation in FORTRAN I. Pages 19-20.

* Roy Nutt. Compiler Techniques Available in 1954. Pages 20-22.

* Frances E. Allen. A Technological Review of the Early FORTRAN Compilers. Pages 22-26.

* John Backus. Afterward. Pages 26-27.

o Jeanne Adams, session chair. Institutionalization of FORTRAN.

* Herbert S. Bright. Early FORTRAN User Experience. Pages 28-30.

* Robert A. Hughes. Early FORTRAN at Livermore. Pages 30-31.

* William P. Heising. The Emergence of FORTRAN IV from FORTRAN II. Pages 31-32.

* Martin N. Greenfield. The Impact of FORTRAN Standardization. Page 33.

* Daniel D. McCracken. The Early History of FORTRAN Publications. Pages 33-34.

* Charles Davidson. The Emergence of Load-and-Go Systems for FORTRAN. Pages 35- 37.

* James M. Sakoda. A Dynamic Storage Allocation Language–DYSTAL. Pages 37-38.

* Bruce Rosenblatt. The Successors to FORTRAN–Why Does FORTRAN Survive? Pages 39-40.

o Daniel N. Leeson. IBM FORTRAN Exhibit and Film. Pages 41-48.

o J.A.N. Lee. An Annotated Bibliography of FORTRAN. Pages 49-58.

o Henry S. Tropp, editor. FORTRAN Anecodotes. Pages 59-64.

o Anonymous. Meetings in Retrospect. Pages 65-69

* FORTRAN Celebration at IBM Santa Theresa Laboratory.

* FORTRAN Activities at SHARE Meeting.

* David Padua. The FORTRAN I Compiler. Computing in Science and Engineering, Volume 2, Number 1, January/February 2000, pages 70-75. PDF at uiuc.edu IEEE Digital Library. Part of a special issue with the theme of "the 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century." [Editors, page 22. PDF at stanford.edu IEEE Digital Library]

<u>Interviews and biographies of John Backus</u>

* Transcript of Larry Saphire Interview of John Backus, San Francisco, California, December 15, 1967. Number TC-46 of a series in the IBM Oral History of Computer Technology. Topical Outline, Track 1, Track 2, 1+32+22 pages. Handwritten annotation "COPY orig to LOC". Given to Paul McJones by John Backus.

* Henry Tropp, editor. Transcript of discussion, SHARE meeting, Hilton Hotel, San Francisco, March 1972. Computer History Museum Lot X2677.2004, Box 3 of 6, blue 3-inch binder. Donated by J.A.N. Lee.

"The complete transcript of this discussion covers much more than FORTRAN, but there are several pages of very frank and open comments about the development of FORTRAN and some of the vexations of getting the system into the hands of users. Participants in the discussion include John Backus, Tom Steel, Jr., Frank Engel, Jr., Betty Ryckman, George Ryckman, Frank Wagner, William Gautney, John Greenstadt, Harry Cantrell, Ted Dollata, Arnold Smith, and Mort Bernstein. The original tapes of the discussion are in the Smithsonian's National Museum of American History." [Lee 1984]

* Claire Stegmann. Pathfinder. THINK, IBM Corporation, July/August 1979, pages 18-27. PDF

"An interview with John Backus on the 25th anniversary of the beginning of the FORTRAN project." [Lee 1984]

* ACM Turing Award citation

* Charles E. Larson, interviewer. Pioneers of Science and Technology Video Tape Project, San Francisco, CA, Oral History 262, Charles Babbage Institute, University of Minnesota, January 3, 1984.

* Robert Slater. The Man Who Invented Fortran. In Portraits in Silicon, The MIT Press, 1987.

* Dennis Shasha and Cathy Lazere. John Backus. In Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists, Copernicus/Springer-Verlag, 1995.

* Steve Lohr. FORTRAN: the “Early Turning Point”. In Go To, Basic Books, 2001. Online at inventors.about.com

* Steve Lohr. Pioneers of the ‘Fortran’ Programming Language. New York Times, June 13, 2001. Online at fortran.bcs.org

* Grady Booch, interviewer. Oral History of John Backus, Ashland, Oregon, September 5, 2006. Computer History Museum reference number: X3715.2007 (accession number 102657970), 42 pages. Online

o “Abstract: In this interview, John Backus reflects on his life and career from his earliest days in the U.S. Army during World War II through his career at IBM. He discusses IBM culture and the environment that led to the creation of Fortran, analyzes the development and influence of Fortran, and reviews his work in functional programming.”

* The History of Computing Project

* The MacTutor History of Mathematics archive

* University of Pittsburgh Information Science Hall of Fame

* René Gabriëls, Dirk Gerrits, and Peter Kooijmans. John W. Backus: 3 December 1924-17 March 2007. Report for class 2R930 Geschiedenis van Informatica (History of Computing), Faculteit Wiskunde & Informatica, Technische Universiteit Eindhoven, May 29, 2007. PDF at tue.nl

<u>Memoirs and user stories</u>

* Bob Bemer. Web site: Computer History Vignettes. Online at bobbemer.com

Bemer joined IBM in 1955 and worked for Backus in the Programming Research Department. FORTRANSIT translated FORTRAN to IT, then used a modified version of Alan Perlis’s IT compiler to generate IBM 605 SOAP assembly language. The main memory of the IBM 650 was a drum, so the instruction placement optimization performed by SOAP was crucial to getting reasonable performance.

o FORTRANSIT – Making FORTRAN a Winner. V10. Online

o Who Was Who in IBM’s Programming Research? – Early Fortran Days. V11. Online

* Robert Hughes. An Interview with Bob Hughes. May 9, 1997. Part of: George A. Michael. Web site: Stories of the Development of Large Scale Scientific Computing at Lawrence Livermore National Laboratory. Online at computer-history.info

* John Van Gardner. Fortran and The Genesis of Project Intercept, May 2005, 2 pages. PDF

Gardner was one of the IBM Customer Engineers who installed 704 serial number 13 at Lockheed Aircraft in Marietta, Georgia in May 1956. This memoir describes how in 1957 he debugged a hardware problem that had resulted in the Fortran compiler behaving in a nondeterministic manner.

* Mark Halpern. On the Heels of the Pioneers-A Memoir of the Not-quite-earliest Days of Programming. Annals of the History of Computing, Volume 13, Number 1, 1991, pages 101-111. Online at rules-of-the-game.com IEEE Digital Library

His first assignment after joining the IBM Programming Research Department in 1957 was to study and document (via flow-charts) the FORTRAN compiler.

* Dennis E. Hamilton. Impact of FORTRAN II language changes. Personal communication to Paul McJones, April 2005:

“However, the impact of small changes and improvements can be immense. The ability to build Fortran programs out of independently-compilable modules and to have the ability to decompose into functions and subroutines using Fortran or any other tool that produced compatible code (usually the assembler, in those days) had an immense impact. In Fortran I programs were one giant file and there was no modularization structure. That small change in Fortran II was earthshaking in terms of software development and, I think, the endurance of Fortran as a technical-software programming tool.

It also changed the way that computers had to operate to make software building and use work more smoothly. I think it is no coincidence that this paralleled increased interest in operating systems (called things like tape monitors, at the time) and the use of the computer for organizing the data processing workflows. (There was also a lot of resistance to operating systems in those days.)”

* Norm Hardy. Fortran at Livermore. Web page, undated. HTML at cap-lore.com
“Here are a few recollections on Fortran when it appeared at Livermore. ...

Fortran’s optimization was far ahead of its time. Indeed other much smaller and faster Fortrans were soon written for other machines, but I recall observing that it was about 10 years before any compiler optimized as well as the original.

Some months after the compiler was shipped, IBM released the ‘source’ for the compiler in the form of microfilm images of assembler listings of the compiler, carefully hand annotated with the patches. A year or so later IBM shipped a clean assembly of the compiler with some of the less significant optimizations absent. The newer compiler was yet faster and more reliable and we were able finally to compile our large production code. The newer compiler still led the pack regarding optimization.”

* Peter Crouch, editor. Fifty Years of Fortran. British Computer Society Fortran Specialist Group and Computer Preservation Society, January 25, 2007. Online at bcs.org

Proceedings of a meeting held at the BCS London Office, January 25, 2007.

* Gareth Mitchell, presenter. Fortran is 50. Digital Planet programme, BBC Radio World Service, December 18, 2007, 6':40". MP3 (7.6 megabytes)

Mitchell interviews Paul McJones on the occasion of the 50th birthday of Fortran; additional commentary by Bill Thompson; produced by Helena Selby.

Photographs

* John Backus and John Cocke. Match book cover labeled "Capt. Starn's Restaurant and Bar, Atlantic City, New Jersey", undated. Given to Paul McJones by John Backus. JPEG

* Richard Goldberg, Robert Nelson, Lois Haibt, Roy Nutt, Irv Ziller, Sheldon Best, Harlan Herrick, John Backus, Peter Sheridan. Pioneer Day Banquet, National Computer Conference, Houston, Texas, June 9, 1982. Photographer unknown. Given to Paul McJones by John Backus. JPEG

Films/video

* FORTRAN films. Computer History Museum Lot X2843.2005. Donated by Daniel N. Leeson.

o FORTRAN. IBM Department of Education, Poughkeepsie, New York, 1958?, 16':27".

* Windows Media Video (100.9 megabytes)

* QuickTime Video (63.2 megabytes)

o FORTRAN 25th anniversary film, 1982, 12.5 minutes. Copy belonging to Micah Nutt. Windows Media Video (12.8 megabytes)

Daniel N. Leeson describes both these films in his article "IBM FORTRAN Exhibit and Film" in Annals of the History of Computing, Volume, 6, Number 1, January 1984, pages 41-48. He says the first FORTRAN film was made in Poughkeepsie about 1958, and that Harlan Herrick owned the only known copy. He describes the production of the 1982 film in some detail, and includes a complete transcript.

* Excerpts from FORTRAN session. History of Programming Languages Conference, 1-3 June 1978, Los Angeles, California. Audio plus still pictures in Real Media and QuickTime format. Online at Virginia Tech

Bibliographies

* Nelson H. F. Beebe. Bibliography on Fortran. Department of Mathematics, University of Utah, version 1.44, April 12, 2005.

o Part 1: 1956-1980. HTML

o Part 2: 1981-1989. HTML

o Part 3: 1991-present. HTML

* J.A.N. Lee. An Annotated FORTRAN Bibliography. Annals of the History of Computing, Volume 6, Number 1, January 1984, pages 49-58. IEEE Digital Library

Simulators

* Paul Pierce. 709/7090/7094 simulator and tools. http://www.pierce-fuller.com/oldibm- shadow/709x.html

* Dave Pitts. 7090 simulator and tools. http://www.cozx.com/~dpitts/ibm7090.html

Dave notes that Paul Pierce's version won't run IBSYS at all, because it lacks channel changes and 7094 instructions that Dave added to his version. Rob Storey's is also not currently as far along as Dave's. [Personal communication, May 5, 2005]

Screenshot of Dave's emulator after FORTRAN II compilation. JPEG

* Rob Storey. 7094 emulator. http://members.optushome.com.au/in-taemul/Emul7094.htm

Related resources

* Al Kossow. Bitsavers' Software Archive and PDF Document Archive.

o Home: http://www.bitsavers.org/

o Mirrors: http://bitsavers.trailing-edge.com/-http://www.classiccmp.org/bitsavers http://bitsavers.vt100.net/-http://computer-refuge.org/bitsavers-ftp://zerkalo.cu-rie.fr./mirrors/www.bitsavers.org/pdf/

"As of Mar, 2008 there are over 13,300 documents containing over 1,228,400 pages in the archive."

* Fortran Archive Catalog. Computer History Museum. Online at CHM

This is a predefined query on the Museum's in-house MIMSY collection database that produces catalog entries for some 71 items, mostly from this web site.

* Paul Pierce. Computer Collection. http://www.piercefuller.com/collect/in-dex.html

The web site describes Pierce's collection of computer hardware, which includes an IBM 709 and an IBM 7094. The web site also includes a library with scanned copies of manuals as well as machine readable images of card decks and magnetic tapes including SHARE, IBSYS, and CTSS distributions.

* Diarmuid Pigott. HOPL: an interactive Roster of Programming Languages. http://hopl.murdoch.edu.au/

"This site is concerned with the idea – historical treatment of the development of programming languages as a means of human expression and creation. In 1976, at the History of Computing Conference in Los Alamos, Richard Hamming described why we might be interested in the history of computing: "We would know what they thought when they did it".

This site is all about why they did it – why people designed and implemented languages and what influenced them when they did so (historically, philosophically, politically as well as theoretically).

This site lists 8238 languages, complete with 17444 bibliographic records featuring 10624 extracts from those references. It is in effect a family tree of languages with 5314 links, making it not only the biggest programming language family tree around, but also one of the largest idea-genealogical projects undertaken."

* Ed Thelen. On-Line Documents. http://ed-thelen.org/comp-hist/on-line-docs.html

The web site includes scanned (and in many cases OCRed) copies of useful reference works for the computer historian including books, reference manuals, directories, etc. Here is an example relevant to FORTRAN: a document that lists all the IBM 704s that were sold:

o Martin H. Weik. "A Third Survey of Domestic Electronic Digital Computing Systems" Report Number 1115, Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland, March 1961. HTML

* Frank da Cruz. Columbia University Computing History. http://www.columbia.edu/acis/history/

This web site is a rich source of information about computing activities at Columbia, which hosted IBM's Watson Laboratory from 1945 to 1970. Examples of pages relevant to FORTRAN include John Backus and the IBM 704.

* Peter Crouch, Chairman. Fifty Years of Fortran. Meeting organized jointly by the British Computer Society Fortran Specialist Group and the Computer Conservation Society, January 25, 2007. http://www.fortran.bcs.org/2007/jubileeprog.php

o Program with slides.

o Written contributions.

o A list of FORTRAN implementations, 1957-1967.

o Further information.

* Simulators for other historic machines [these links belong somewhere on the Software Collection Committee web site, but aren't specifically FORTRAN-related.]:

o Martin Campbell-Kelly. The Edsac Simulator. http://www.dcs.warwick.ac.uk/~edsac/

o David Richard Green. SILLIAC Emulator. http://members.iinet.net.au/~dgreen/sil- liac.html

SILLIAC was Sydney University's almost-exact copy of the ILLIAC. Green's web site includes a 1958 SILLIAC Programming Manual and links to SILLIAC and ILLIAC software.

o Peter Zilahy Ingerman. UNIVAC I and II Simulator. http://www.ingerman.org/niche.htm#UNIVAC

o Leif Jon Harcke. Atlas – an 1103A emulator for UNIX systems. http://insar.stanford.edu/~lharcke/programming/

o Bob Supnik. Computer History Simulation Project (SIMH). http://simh.trailing-edge.com/

Various machines by Data General, DEC, GRI, IBM, Interdata, Hewlett-Packard, Honeywell, MITS, Royal-Mcbee, and Scientific Data Systems, plus software kits and some papers on simulation.

o Ron Burkey. Virtual AGC and AGS (Apollo Guidance Computer and Abort Guidance System). http://www.ibiblio.org/apollo/

o Jay Maynard. The Hercules System/370, ESA/390, and z/Architecture Emulator. http://www.hercules-390.org/

Tom Hunter. Desktop CYBER Emulator (CDC CYBER 6600, 7x, 17x). http://members.iinet.net.au/~tom-hunter/