AIPS Memo 100

The Creation of AIPS

Eric W. Greisen

July 27, 1998

Abstract

At this writing, the AIPS package of software has been in active development and use for over 19 years. The present manuscript is an attempt to summarize the discussions and earlier software packages that led to the creation of AIPS and to describe what AIPS was like during its formative years.

1 Introduction

The NRAO Astronomical Image Processing System (AIPS) is a software package for interactive (and, optionally, batch) calibration and editing of radio interferometric data and for the calibration, construction, display and analysis of astronomical images made from those data using Fourier synthesis methods. Design and development of the package began in Charlottesville, Virginia in 1978. It presently consists of over 1,000,000 lines of code, 100,000 lines of on-line documentation, and 300,000 lines of other documentation.[1] It contains over 386 distinct application "tasks," representing *very* approximately 70 man-years of effort since 1978.

In contrast with modern practices, AIPS was not designed on paper and then translated into code. It was not accompanied by code and documentation management systems and even reports to management and oversight committees were essentially informal and irregular. The first *AIPSLetter* did not appear until November 1981, the AIPS Memo Series was not begun until April 1983, a code checkout system was not in place until June 1984, and a proper code management system with full accountability and recoverability was not instituted until December 1990. Files recording transactions in the earlier years were allowed to disappear when they were judged "obsolete." In fact, records of the export of AIPS were originally kept only in the form of the n^{th} copy of the tape shipping order forms thrown into a table drawer. As a consequence of this casual attitude, the historic record for this project is spotty. The files have yielded remarkable original design documents in some areas and no hint that major committees even existed in others.

Nonetheless, it has been fun and instructive to delve into the historic record and I hope people who I may slight will forgive me for the following summary. Further current information on AIPS can be obtained by writing by electronic mail to `aipsmail@nrao.edu` or by paper mail to the AIPS Group, National Radio Astronomy Observatory, Edgemont Road, Charlottesville, VA 22903-2475, U.S.A. AIPS information is also available on the the World-Wide Web at URL `http://www.cv.nrao.edu/aips`. AIPS Memos 61 and 87 are particularly helpful, along with an early article by Don Wells.[2]

[1] Counted on 29-May-1997 and omitting the GNU copyrights, PostScript files, and obsolete areas.

[2] Greisen, E. W., AIPS Memo No. 61, "The Astronomical Image processing System," September 1988 and Bridle, A. H., Greisen, E. W., AIPS Memo No. 87, "The NRAO AIPS Project – a Summary," : April, 1994 and Wells, D. C., "NRAO's Astronomical Image Processing System (AIPS)," *Data Analysis in Astronomy*, Eds. Di Gesù, V., Scarsi, L., Crane, P., Friedman, J. H., Levialdi, S., Plenum Press, New York and London, 1984.

2 Early Committees

The debates about software for the VLA probably began with the first design documents for the telescope if not sooner. The January 1967 VLA Proposal Volume II contains an interesting and insightful view of the computing problem in a chapter attributed primarily to Barry Clark. Its opening paragraph was:

> "Early in the design stage of the VLA, it was realized that an array of several tens of antennas connected to several hundreds of receivers would present problems in control and display far beyond those encountered in any radio astronomy system presently in operation. The most immediate solution to these problems is to have a digital computer perform the detailed functions, receiving from the operator only a generalized description of the task it is to perform. Once one conceives of using a digital computer for control and monitoring of the antennas and receivers, it is very natural to conceive of extending its duties to the manipulation, control, and display of the data of the array as well. Indeed, the computation problems in data manipulation are very quickly seen to be of much greater magnitude than those in monitor and control."

The Proposal goes on to say

> "After the completion of the observation, the computer will sort the observed data points onto the u-v plane, calculate and apply various calibration corrections, combine observations, apply a weighting specified by the observer, perform the Fourier inversion, and output a map of the region of sky under study. This will be done asynchronously with the computations necessary for observations, but at a rate such that the computations will not fall behind. No backlog should be allowed to form."

After a detailed analysis, it was concluded that a computer of 2 million floating point operations per second (MFLOPs) would suffice. The total cost would be $3.4 million for computer hardware including communication to the telescopes and $150 thousand for the software. More insightful perhaps was the remark

> "In either case, the output map would be recorded on magnetic tape for further computer processing, though it is not immediately anticipated that this special processing should be programmed on the VLA computer system. This is probably more suitably done on a large, general purpose computer."

This last sentence has occupied a lot of us ever since. It was soon decided that we should move the asynchronous portion of the computing to a computer system fairly isolated from the real-time machines. A DEC-10 was purchased for the purpose and over the years several software systems developed to run on it. These are the subject of other papers and will not be described here other than to remark that the study and design documents for these systems helped to refine the lists of operations which we hoped to do on the *uv* data and images produced by the VLA. It was both amusing and frightening to watch as the "required" computer power grew with every review of the data processing needs.

In May 1976, three months before VLA antenna number 6 was supposed to be delivered, a VLA Advisory Committee meeting was held. In preparation for the meeting, memos were written by Bob Hjellming and Bob Burns. Hjellming's memo was mostly concerned with immediate problems but asked whether one-fourth of the asynchronous group should devote its time to solving imaging questions related to bandwidth and time smearing and to curvature of the celestial sphere (problems not really solved to date) and whether "the basic assumption that all calibration should be done on-site" was correct. He did propose an output *uv*-data format not unlike the Export format to be described below. Burns was more concerned with questions related to off-site data processing, in particular a plan to use the Charlottesville IBM 360/65 to begin an "interim post-processing development." Showing considerable foresight, Burns asked questions about computer independence, mini-computers, and the use of large computational centers. In July 1976, Burns wrote a lengthy memo, later called VLA Computer Memorandum 139, entitled "VLA Post-Processing: An Initial Discussion and Proposal." He defined post-processing to include:

1. "Additional editing and calibrating, if required,
2. *uv*-plane data display,
3. further map synthesis,
4. map correction,
5. map analysis and interpretation, and
6. map display for user and for publication."

After an analysis of available facilities, current practise and estimates of the software needs for VLA processing, he concluded "The scientific output of the VLA will be diminished if the NRAO does not provide adequate facilities for all stages of the data manipulation and analysis." He discussed machine independence again and described a plan to develop software both on the existing IBM and on a mini-computer which would be equipped in time with interactive display devices and an array processor.

So we talked about it for another couple of years. In March 1977, Dave Heeschen, NRAO's Director, formed an in-house scientific committee chaired by Mort Roberts to investigate a number of questions related to off-site processing both at some NRAO facility and at the users' home institutions. The report of this committee appeared in October 1977 and supported the concept of a central large computer with several mini-computer systems of which some would be at non-NRAO facilities. They deduced that the software would have to be provided by NRAO and urged immediate commencement of a project to produce a post-processing system. In November 1978, Mort Roberts, by then the NRAO Director, asked Dave Shaffer to chair a committee "to propose a unified approach to VLA post-processing." Judging by private notes of mine and remarks on drafts by Shaffer, the discussion had become considerably more acrimonious, dividing primarily on Socorro versus Charlottesville lines. A report was finally issued in February 1979, that basically defined the initial *AIPS* project, concluding

> "Our principal recommendations are thus: A Development Group in Charlottesville; a complete off-line system for New Mexico, to be ready and delivered in 1980; sufficient on-line capability for the VLA site; and at least one, preferably two, systems for Charlottesville. Additional systems depend on user demand."

3 Charlottesville Software Packages and Critical Steps

The last half of the 1970s did not consist entirely of committee meetings and reports, even if it sometimes seemed that way. A number of much more practical developments that eventually led to *AIPS* also took place. The first important step was the development of two formats for the export of data from the VLA. In September 1975, Bill Randolph wrote VLA Computer Memorandum 126 detailing the "Data Format from Synchronous System." This synchronous format was a clever, scalable format with a directory at the beginning giving individual length and address pointers to the areas containing data on sub-arrays (and sources), antennas, bad correlators, and the actual visibilities. This structure allowed more information to be appended to each area without doing major violence to existing software, allowing the format to go through numerous revision numbers during its lifetime. (Revisions 2 and 3 were described by Randolph in an addendum to Memo 126 in July 1976; by April 1982 a re-written memo detailed revision numbers up to 7.) This format was as nightmarish in its binary form as it was clever in its logical form. The data values were in ModComp integer, floating, and extended precision binary forms, but each 32 bits worth of data were expanded to 40 bits on tape with 36 bits intended for the DEC-10 asynchronous computer and 4 bits always ignored. Despite these choices, which were intended to make convenient the reading of these data by the DEC-10, the program that filled the DEC-10 data base was still known by the name "blood sucker" for what it did to everything else attempting to run on that eventually overloaded machine.

The Export format was intended to deliver calibrated and edited data from the DEC-10 on magnetic tape. The meeting to design this format, held at the VLA in September 1976, became somewhat acrimonious since many of the VLA personnel assigned to the problem believed deeply that, and I quote, "no one outside the

VLA will ever be able to read or make sense of these data." Fortunately, Barry Clark, if my memory serves correctly, stepped into the discussion and persuaded us to develop a reasonable format anyway. I wrote a memo to Clark dated November 1976 summarizing the format we had designed. A later undated memo by Dave Ehnebuske and Jerry Hudson described an improved Export format that *AIPS* can still read. The tapes were written in IBM VSB (variable, spanned, blocked) format with the unusual specification that each logical record ended with 4 16-bit words which would describe the next logical record. The tape began with format definition records, but the practical problems of negotiating changes meant that we never used this invitation to upgrading the format over time. Fortunately, the logical records and the very structured form in which they occurred were well designed and did not require serious modification. Bytes were in standard industry order and tape form and all data were in integer with decimal points at defined locations within the words. These attributes made the data easy to read on a wide variety of computers but must have been hard work for the DEC-10 to write. The Export format was supposed to be in production, according to some of the 1976 memos, by Spring 1977. However, the memo I wrote to Barry Clark indicating that I had received a usable tape was dated December 1977 and I wrote a number of other memos to Clark through most of 1979 indicating residual problems with the Export tapes written by the DEC-10.

Spurred, I suspect, by the "no one outside the VLA" remark, Fred Schwab and I decided to write a package of VLA data reduction programs to run on the IBM 360/65 in Charlottesville. We had both worked on the Green Bank Interferometer software, so we simply converted all of that package to work with a VLA-appropriate format similar to the one used for Green Bank data. It took us four weeks of furious effort to get the package going, although rewrites to allow for more antennas and to add new capabilities occupied us for some time thereafter. By March 1977[3] this package was able to read synchronous system data tapes and perform a variety of operations on the data. It had an elaborate algorithm to find and flag bad data, which were all too common in the early days before sophisticated on-line flagging. It also had the Green Bank package's clever routine to find antenna locations and a full suite of routines to correct data for changes to source and antenna positions and for various atmospheric and elevation effects. It did the standard gain and polarization solutions and applied them to the data. It had averaging, sorting, and model fitting and subtracting programs as well as several printer displays for the *uv* data. It also was able to map the data, do a standard Högbom Clean, and display the images on printers, CalComp plotters, and the Dicomed film recorder. It even had a Users Guide. The programs were in PL/1 and were run in sequences managed by IBM Job Control Language, `PARMS` and `INCLUDE/EXCLUDE` cards with data normally read from and written to tape.

By September 1977[4], a second package of IBM PL/1 programs had been created based on the Export format. This "`DEC`" package had *uv*-plane capabilities similar to the "`VLA`" package and used the same map Cleaning and display programs and program control logic. Development of this package continued through all of 1978 and 1979. It was slowed by problems with the Export data and by my devoting much of my time to a direct fore-runner of *AIPS*. Although powerful, both of the IBM packages suffered from the inherent defects of batch systems. As stated by Burns and Greisen,[3]

> "However, we feel that the long waits required in batch mode for the results of each sub-operation lengthen the data reduction process enormously and cause the astronomer to lose his concentration and to take shortcuts in the processing. The latter effects can degrade the final results. The combination of real-time displays with a responsive computer system, similar to that used for single-dish processing at NRAO, would allow the astronomer to process his data more rapidly, to maintain his concentration on those data, to check fully the results of his data manipulation, and to discover more easily unexpected problems or results which may be present in his data."

While waiting for management to give us lots of expensive hardware, I began an ill-advised software system eventually called `NIPS`.[5] The code that was generated was all in ModComp assembly language, although

[3] Burns, W. R. and Greisen, E. W., VLA Computer Memorandum 140, "VLA Post-Processing: Phase I," March, 1977.
[4] Greisen, E. W., VLA Computer Memorandum 141, "VLA Post-Processing: Phase I Continued," September, 1977.
[5] Greisen, E. W., VLA Computer Memorandum 144, "Post-Processing — Phase I: Technical Memorandum: The Beginnings of `NIPS`", March 1978.

sections in Fortran were planned, and the program structure was geared to use every arcane bit of the ModComp architecture of the day. I learned a lot by doing — and abandoning — this system. It was a heroic attempt to make a small computer do more than it really could, but it completely ignored the wisdom of writing machine-independent code and systems in which more than one programmer could participate.

The last critical pre-*AIPS* event was the development of the FITS format. Don Wells has written a lovely history of the event which may be found at the FITS World-Wide Web site.[6] In December 1976, Ron Harten (then at the Netherlands Foundation for Radio Astronomy) and Don Wells (then at the Kitt Peak National Observatory) began a discussion of data interchange formats. They exchanged test data in several forms over the next two years. The National Science Foundation organized a meeting in Tucson in January 1979 whose primary purpose was to make image processing capability more widely available in the U.S. astronomy community. This led to a task force on image interchange formats and a meeting at the VLA organized by Bob Burns 26–29 March 1979. Prior to the meeting, Wells sent around documents describing his efforts with Harten which produced the following remarks from Barry Clark:

> "A single physical block size of either 1440 or 2880 Bytes sounds to me like a reasonable record length. Shorter is inefficient use of tape, longer will encounter buffer problems in very small systems. I suggest the header information should correspond to some reasonable standard, with keywords being the main definition effort. I suggest as a standard for the header that keywords be limited to six characters, be followed by `'='`, `' = '`, `'= '` or `' ='` and then by a single value. A string of blanks would be equivalent to a single blank. Values would be in the form for Fortran 77 list directed I/O."

Clark included a page titled "Suggested List of Keywords," five of which made it into the final Basic FITS Agreement: `BSCALE`, `BUNIT`, `OBJECT`, `HISTOR` and `COMENT`, although the last two were re-spelled when it was decided to use 8-character keywords. This meeting was remarkable in that it generated a consensus on a very general format, including a flexible way to describe multi-dimensional images initially suggested by Harten. This format, *without change* other than the addition of further forms to follow the Basic FITS images, is an international standard to this day.[7] The importance to the insides and outsides of *AIPS* of the general FITS way of looking at data cannot be overstated. The Charlottesville "`DEC`" package produced the first FITS tape in April 1979, a tape was returned by KPNO in September, and the VLA asynchronous system (DEC-10) produced its first usable FITS image tape in November 1979.

4 Things Start to Get Serious

Most of the period April 1978 through June 1979 was spent in innumerable design discussions, trying to figure out what we should do and how and where we should do it. In the Fall of 1978, I visited the VLA to take a close look at the IMPS package developed by Jim Torson with help from Al Braun. A year later, I visited Groningen with Ed Fomalont to get a good look at their GIPSY package. Both of these were ultimately rejected, perhaps because we wanted to develop and/or have full control of our own software. It is true that both systems were not coded for portability and were otherwise tied to their local hardware; IMPS in particular was heavily committed to a particular, very nice graphics device.

The issue was beginning to get serious not only because the VLA was actually producing a lot of data, but also because money began to become available for additional computing hardware. Dave Heeschen finally (in our view) responded to our memos (and undoubtedly the input from a great many other people) to offer Bob Burns about $300,000 to buy peripherals for the ModComp in Charlottesville to begin a post-processing project. Burns admitted to me that he was so tired of asking for the money that he almost turned it down; fortunately, he did not. Let me remind the reader what things cost in 1977:[4] an 84-Mbyte, 3330-type disc

[6] in particular `http://www.cv.nrao.edu/fits/documents/overviews/history.news`.

[7] Wells, D. C., Greisen, E. W., and Harten, R. H., 1981, "FITS: a flexible image transport system," Astronomy and Astrophysics Supplement Series, 44, 363-370. Further references and description may be found at WWW `http://www.cv.nrao.edu/fits`.

with one port and a controller was \$32000, 64 Kilobytes of ModComp memory was \$17500, a Floating Point Systems array processor was around \$130000 with 64 kilo-words of fast memory, and an IIS Model 70 TV display cost roughly that amount as well. I remember that IIS memory was \$1100 per 512×512 bit plane, or roughly \$40000 just for the basic memory in the displays we eventually acquired. And these are 1977 dollars. The purchasing process for an array processor (AP) and image display was begun in the first quarter of 1978; the AP was installed on the ModComp early in 1979 and the image display in the Spring of 1979.

Digital Equipment Corporation announced a new, eventually revolutionary mini-computer called the VAX 11/780 in October 1977. About a year later, Tom Cram and I walked into Bob Burns' office and jokingly suggested that, if he really wanted us to write machine independent code, then he should buy us one of the VAXes to go with the ModComp. We took it to be a joke because the total cost with peripherals would be around half a million dollars, but the joke was on us. Burns pulled some of his magic and a VAX known as `VAX1` was purchased for the VLA but delivered to Charlottesville in the fourth quarter of 1979, with the array processor and image display arriving a few months later. This new computer had real software development tools, a virtual memory operating system (VMS), and dynamic disk file creation and other modern concepts. The ModComp was an excellent real-time computer, but its text editors were simple minded and its debugger non-existent. Furthermore, its 128-Kilobyte memory limit meant that programs had to be heavily overlayed and hence that a simple link edit could consume 0.5 to several hours. The ModComp, however, kept us honest in our coding and was largely responsible for the high degree of machine independence eventually achieved by *AIPS*.

Another matter that made things feel serious was the attitude of the National Science Foundation. To quote from a letter signed by William E. Howard, III, Director of the Division of Astronomical Sciences and dated June 5, 1980:

> "... I should reiterate that the VLA data processing problem is one that NRAO must solve within its own budget, that the NRAO must spend its own funds on a computer system that promises to meet visitor needs adequately and that we expect that the data reduction needs of all VLA users, visitors as well as staff, should be met to the same degree and to the same extent that such needs have been met at all the other NRAO telescopes in the past. ... If it appears that visitors must spend unacceptably and unreasonably long periods of time at NRAO reducing and thinking about their maps, the NRAO must place a higher priority on their computer expenditures for VLA reduction at the expense of other programs. ... We expect NRAO to solve the VLA data processing problem, not the NSF."

5 Starting on *VPOPS*, *RANCID*, and *AIPS*

The ModComp computer became a very busy place in the middle of 1979. Fred Schwab developed a stand-alone program named `SCAL` to do self-calibration of VLA data. It took input in the form of time-baseline ordered Export format *uv* data on magnetic tape and source model data on punched cards. The model data could be produced by a special IBM program run on the components map generated by one of the IBM map Cleaning programs. Schwab's July 1979 user instructions are very nostalgic in that they even describe all the devices that need to powered up and which panel switches to press to start the computer. `SCAL` had a very nice option: it used the front panel sense switches to allow the user to turn on and off displays of the solutions. Parameters were provided to `SCAL` with a question-and-answer session at its beginning.

At the same time, I began coding routines to drive the image display and Tom Cram began in earnest to translate the `POPS` code used by single-dish programs into a program called `VPOPS`. This program was intended to perform quick operations to access the users data catalog and interactive devices and to start separate programs called tasks to perform longer operations. The `POPS` code developed in the 1970s by Jerry Hudson and put to practical use by Tom Cram remains the heart of the `AIPS` program today, although we have added functionality to the `POPS` language and code over the years.

The Post-Processing Group in 1979 consisted of Ed Fomalont (scientific direction), Bob Burns (technical management), Tom Cram (systems, `POPS`), David Brown (system support), Fred Schwab (algorithms), and myself (whatever). In September 1979, Tom Cram left NRAO for KPNO and in January 1980 Bill Cotton and Walter Jaffe joined the Post-Processing contingent. There was also a Post-Processing Committee chaired by Carl Bignell which began meeting in April 1979 and continued to meet until it was dissolved at its own recommendation in April 1981. This committee was onerous for two main reasons. The first was the obvious problems of significant preparation plus a week's travel between Socorro and Charlottesville for a significant number of people every couple of months. Less tangible, but more serious, was the somewhat natural state of tension between the two groups, both the tensions between VLA personnel and "outsiders" and between anxious users and overwhelmed programmers. The report of the January 1980 meeting of the Committee, which I cannot find, must have been particularly "interesting." It provoked three significant reactions. The first was a contract let to Jerry Hudson, who no longer worked for NRAO, to study the use of `POPS` for post processing. His report of March 1980 concluded that `POPS` was in need of improvement but was adequate for a "first-generation post-processing system." He also approved of the "task-shedding" scheme. The second reaction was a well-prepared and lengthy report by Ed Fomalont to the Committee for its April meeting outlining major changes including the virtual operating system interface ("`Z`" routines), working file management and I/O, progress on the Clark Clean for the array processor, and a new internal header format. The third reaction was our decision to rename the package to something less likely to be used in every sentence as a rallying point for opposition. That new name was *RANCID*, which stood for Radio Astronomy Numerical Computation Imaging Device, and it achieved its desired effect. For the next year people did not take us so seriously and allowed us to get on with the early design and coding phases with only a reasonable level of political interference.

Finally, of course, the NRAO Director had to go to the Visiting Committee to explain why he had spent around one million dollars on *RANCID* software. This was distasteful to him and he "requested" a new name for the project. On March 31, 1981, the name *AIPS* (Astronomical Image Processing System) was chosen. It beat out `AIDA` (Astronomical Image Display and Analysis), `MIRAS` (Map Imaging Reduction and Analysis System), `DIANA` (Data Imaging and Numerical Analysis), `MADRE` (Map Analysis and Data REduction), `IRIS` (Image Reduction and Improvement System), and a variety of others that were either already taken or unprintable even in those politically incorrect days. I guess that, if the first of these had won, then we would all be knowledgeable today about opera rather than primatology. The effect of the other names is hard to estimate.

In early 1980, a committee of users engaged in a remarkable exercise of attempting to design an internal header for *VPOPS*. I have no papers left from that committee, but remember the results as being rather worse than a cynic would normally expect. Fortunately, Don Wells visited me on April 11, 1980 in Charlottesville and together we explored the idea of using a FITS-like header as an internal one. We realized quickly that a binary data structure based on FITS principles could be used to describe both images and *uv* data and up to 20 "extension" files in fewer than 256 integer words (512 bytes in those days). By the end of that month, the structure we defined that day was adopted as the internal header along with a scheme of computed, mnemonically-named pointers to each of the components which is still in use today.

6 The Joys of Coding and Using Early *AIPS*

The creation of truly portable code was difficult in 1980. Fortran 66 was not a standard language in modern terms. There was no official ANSI standard and word lengths and the relationships between word lengths were not defined. There were no character variables and there were essentially no standard input/output methods. Computers in those days were also very variable. DEC's PDP 11s were 16-bit byte addressing computers which limited a program to 64 kilobytes in length. ModComps allowed programs twice as long, while DEC VAXes allowed any length due to its new Virtual Addressing eXtension. There were 24-bit, 36-bit, and 60-bit computers in widespread use as well as several character encoding schemes. We were told by various committees that we had to write code to run on the PDP 11s, which were owned widely in the astronomical community at that time, but, even with the ModComp to help, we never managed to squeeze our programs down enough to try that port. We did set ourselves the goal to run on all of the larger machines we could find and, over the years, had some success in that.

In April 1980, Frank Ghigo took a copy of *RANCID* to install on the University of Minnesota's main Cyber 74 computer. Our ModComp and VAX had 16-bit integers, 32-bit floating-point, and ASCII characters. His computer had 60-bit integers and floating-point and a 6-bit character. When he wrote us in September 1980[8], he had gotten about 70 subroutines to run, which was about enough to get the `POPS` processor to run without most of the current verbs. He was very complimentary about the general portability of our efforts — which were good for the time — although he did write that "initial attempts to decipher `MSGWRT` and its attendant Z-routines led me to suspect the work of a madman." His suggestions for system-wide parameters, better handling of characters and equivalences, and the like caused me to conduct an initially surreptitious re-write of all of *RANCID* as the rest of the group continued to generate new code.

The result of all the attempts to achieve portability and efficiency was an *AIPS* coding style that was ponderous and demanding. Although we hoped to allow astronomers to code in *AIPS*, the complexity of things was too high a barrier for all but the most determined. Among its attributes were:

- All integers were explicitly 16-bit integers, capable of counting only between −32768 and 32767. Integers to count over a wider range consisted of 2 standard integers and all operations on them were performed by subroutines, especially the infamous `ZMATH4`.
- No numeric constants were allowed in call sequences since compilers did not agree on the type that numbers were assigned. Thus, to send a constant to a subroutine, the programmer had to declare and initialize a variable. Our habit was to use obvious names and `DATA` statements such as `DATA N2 /2/`.
- Character strings were stored as 2 characters per integer, 4 characters per float, or as packed strings (as many as would fit). The latter were required for `ENCODE` and `DECODE` of numeric variables, while the former were used for output and string handling. Numerous subroutines were created to access characters and to switch between the forms.
- No assumptions were allowed regarding variable lengths other than certain minimums and that a floating variable held an integer number of reals, etc. System-wide parameters were available in a Fortran `COMMON` to compute pointers into data structures, which were `EQUIVALENCE`d arrays of integers, floats, and doubles.
- Useful statements like `WHILE` and `IF` with `THEN` and `ELSE` were not allowed because some compilers were strict in their Fortran 66 standards. We even had to require all variable declaration statements to precede all `COMMON` statements which preceded all `DATA` statements.
- Because tasks were required to fit in the ModComp address space (or preferably half that), all tasks had to be constructed so they could be overlayed. This meant that `MAIN` routines were themselves short and used to declare all `COMMON`s and to call a sequence of functions (*i.e.*, initialize, do the operation, write history, end).
- System services such as file access, printing, I/O to a terminal, and the like were available only through a virtual operating system interface as presented by the call sequences of a significant number of "Z" routines. Over time, virtual device interfaces to television displays ("`Y`" routines) and array processors ("`Q`" routines) were also developed. This allowed us to have multiple versions for multiple operating systems and devices, but restricted programmer access to basic services.
- All images cataloged on disk were in 16-bit scaled integer form. Normally, this required that an image be computed in floating-point form, written to a scratch file, and then re-read to scale and write to the cataloged file.

[8] Ghigo, F., *AIPS* Memo No. 3, "Adapting *RANCID* to the U. Minn CDC CYBER 74," September 1980.

- For efficiency, computation was expected to overlap disk operations. The code to do this was non-intuitive, requiring, for example, "writes" of a row *before* the data were filled into the row. Buffer pointers, returned by elaborate subroutines, were required at all times.

- Coding was done in one room located between the ModComp and VAX rooms on a variety of simple terminals. Code management was handled by asking around to see if anyone else was working on a particular routine. Debuggers only worked on the VAX, but the ModComp was able to turn up a large number of the bugs. Link edits on the ModComp took hours in many cases since each leaf in an overlay tree had to be link edited separately. There were frequently 3–6 programmers and several others all trying to use one VAX, a VAX that only became a 3-Megabyte computer in the second quarter of 1981.

Things were not much easier for users. The first VAX was shipped to the VLA site in the last quarter of 1980 as planned, along with the FPS array processor, IIS image display, and *RANCID* already on disk. (A replacement VAX was delivered to Charlottesville early in 1981.) A third VAX was turned over to *AIPS* use at the VLA in the middle of 1981, but it was not equipped with the advanced peripherals until early in 1982. To give priority to users assigned priority, a "roller" was added to all array-processor programs. It would periodically roll all of the users' data out of the AP to a disk file and then (after August 1983) check for higher-priority (lower *AIPS* number) AP tasks waiting for the device. If one was found, the task would suspend itself for a time and then check again. Machines became so crowded, however, that the roller action occasionally would fail because a lower priority task could not get the time to roll itself out of the way!

In August 1982, Tim Cornwell wrote an illuminating summary of the rules for post-processing use at the VLA. Users and projects could only use one of the two VAXes and could sign up for time with Ina Cole up to two weeks in advance for a maximum of 8 hours per week. Ina had rules by which she would reduce the time she had assigned to users if others with greater priority requested time, and Ina controlled only half of the available user 1 and user 2 time on each computer. Sign-up sheets for the other half would appear on Monday afternoon for Tuesday and Wednesday, Wednesday afternoon for Thursday and Friday, and Friday afternoon for Saturday through Monday. The maximum time that any group of users could have on each of `AIPS1` and `AIPS2` was 2 hours between 8 am and 5 pm weekdays, 4 contiguous hours between 8 am and midnight any day, and 6 hours between midnight and 8am. Total times were limited to 6 hours in any day and 20 hours in any week. Data were deleted from disk if untouched for 14 days or if the users left the Site.

Why would users put up with such conditions and be willing to work 24 hours a day and to fight each other for resources? Some of the visitors to the VLA did have VAXes at their home institutions and the fraction that did have such facilities grew rapidly with time. However, very few of them had access to the expensive display and, more importantly, array-processor peripherals that NRAO owned. Many of the early users did not particularly like *AIPS*, but the algorithms found in `UVMAP` (optimal *uv*-data gridding and FFT), `APCLN` (image-plane Clean using the Clark algorithm), and `ASCAL` (self-calibration of *uv* data using a Clean-component model), all done with the speed made possible by an array processor, made *AIPS* irresistible. Barry Clark in particular deserves our thanks for not only inventing an efficient algorithm to implement Högbom's Clean technique,[9] but for coding the inner portions of the algorithm in FPS microcode and for allowing Bill Cotton to install that code directly into `APCLN`. Fred Schwab[10] ported his stand-alone self-calibration code with numerous enhancements to *AIPS*. For a while, it retained the nice interactivity of the ModComp control panel, although VAX implementations could not support the option.

7 Progress into a more Modern Era

The first *AIPSLetter* appeared November 1, 1981 and this newsletter has been published more or less regularly ever since. We just sent out Volume 18, Number 1! The first three issues employed special text plotting software, but, with the May 15, 1982 edition, *AIPS* became one of the first astronomical users of

[9] Clark, B. G., VLA Computer Memorandum No. 152, "An Implementation of Clean," December 1979.
[10] Schwab, F., VLA Scientific Memo NO. 136. "Robust Solution for Antenna Gains," September 1981.

Donald Knuth's typesetting software known as TeX. For many years, the *AIPSLetter* contained a typeset copy of the full `CHANGE.DOC` file so that users and programmers could review all the changes before deciding whether they required an updated release of AIPS. This file is now readily available off the World-Wide Web[11] which saves us the enormous labor of typesetting that text and the user the lesser labor of ignoring it. The AIPS Memo Series began in May 1983 with a lot of earlier memos plus one that is still used on coordinate representations.[12] The AIPS *CookBook* was initially written by Alan Bridle from notes he made while trying to figure out how to use RANCID. It was first offered to the public in November 1981 and appeared in TeX form, edited by members of the AIPS group, in September 1983. In March 1984, the old programmer documentation files were re-written and very greatly improved by Bill Cotton. They were published under the title *Going AIPS*. Over the years, the *CookBook* has undergone numerous revisions and is still quite current. *Going AIPS* also was revised a couple of times, but has languished since 1990. Current *CookBook* chapters and most of the AIPS Memo series are available via the World-Wide Web and are distributed with every release of AIPS.

A nostalgic article about AIPS would not be complete without reference to the fun we — and others — have had with the name. The first *AIPSLetter* to have an image of an ape appeared in January 1983. That image required special arrangements to be made for enough disk space (say 10 Megabytes) and took literally hours to compute and print using home-brewed dithering software and a dot matrix printer. It was so expensive that we did not do another until April 1985. The third image to be used to fill the mailing sheet then appeared with the April 1986 *AIPSLetter*. These three appear in a montage labeled Figure 1. The title page of the September 1983 *CookBook* was also decorated with a grey-scale ape. This ape got further publicity in a popular article on the VLA, reproduced in part as Figure 2. The outside covers of the *CookBook* and *Going AIPS* gave NRAO's graphic artist Pat Smiley an opportunity to display her talents, reproduced in black and white as Figure 3. And, of course, the *CookBook* would not be complete without proper recipes such as the earliest ones reproduced as Figure 4.

The first *AIPSLetter* listed the AIPS group (with a *) and supporting cast as:

Al Braun	VLA		DEC/NET and systems work
David Brown	CV	*	VAX/ModComp systems, AIPS on the IBM
Bob Burns	CV		Overall NRAO computer capability
Tim Cornwell	VLA		VLA VAX manager/friend
Bill Cotton	CV	*	U-V software, liaison with VLBI
Ron Ekers	VLA		Overall AIPS priorities
Gary Fickling	CV	*	VAX system, installation, general software
Ed Fomalont	CV		AIPS Project Manager, AIPS priorities
Eric Greisen	CV	*	Software manager
Kerry Hilldrup	CV	*	IBM and general user support
Arnold Rots	VLA		VLA/AIPS spectral-line coordinator
Fred Schwab	CV	*	Applied mathematics
Don Wells	CV	*	Measuring engine, liaison with optical

By this time, Walter Jaffe, who had contributed substantially to the early design, had already left for a year in Holland. In preparation for that trip he suggested and helped to code the pseudo-array processor, a software emulation of the hardware for those who could not afford the real thing. That emulation is now the only "AP" anyone has. A number of the people listed above still work for NRAO, but I am the sole survivor still — or again — working in the AIPS group. A longer list of AIPS participants prepared in 1988 is given in Figure 5.

The first *AIPSLetter* listed 23 institutions that had received AIPS tapes. By May 1983, a similar list included 50 sites outside of NRAO. The July 1982 *AIPSLetter* had a variety of interesting quotes from outside users on the costs of running and keeping up with AIPS. Among them were:

> "At present our AIPS is at a standstill because we have, for the moment, run out of money. This is a result of both the high charges made by the U of M computer center and the considerable

[11] `http://www.cv.nrao.edu/aips.`

[12] Greisen, E. W., AIPS Memo No. 27, "Non-Linear Coordinate Systems in AIPS," November, 1983.

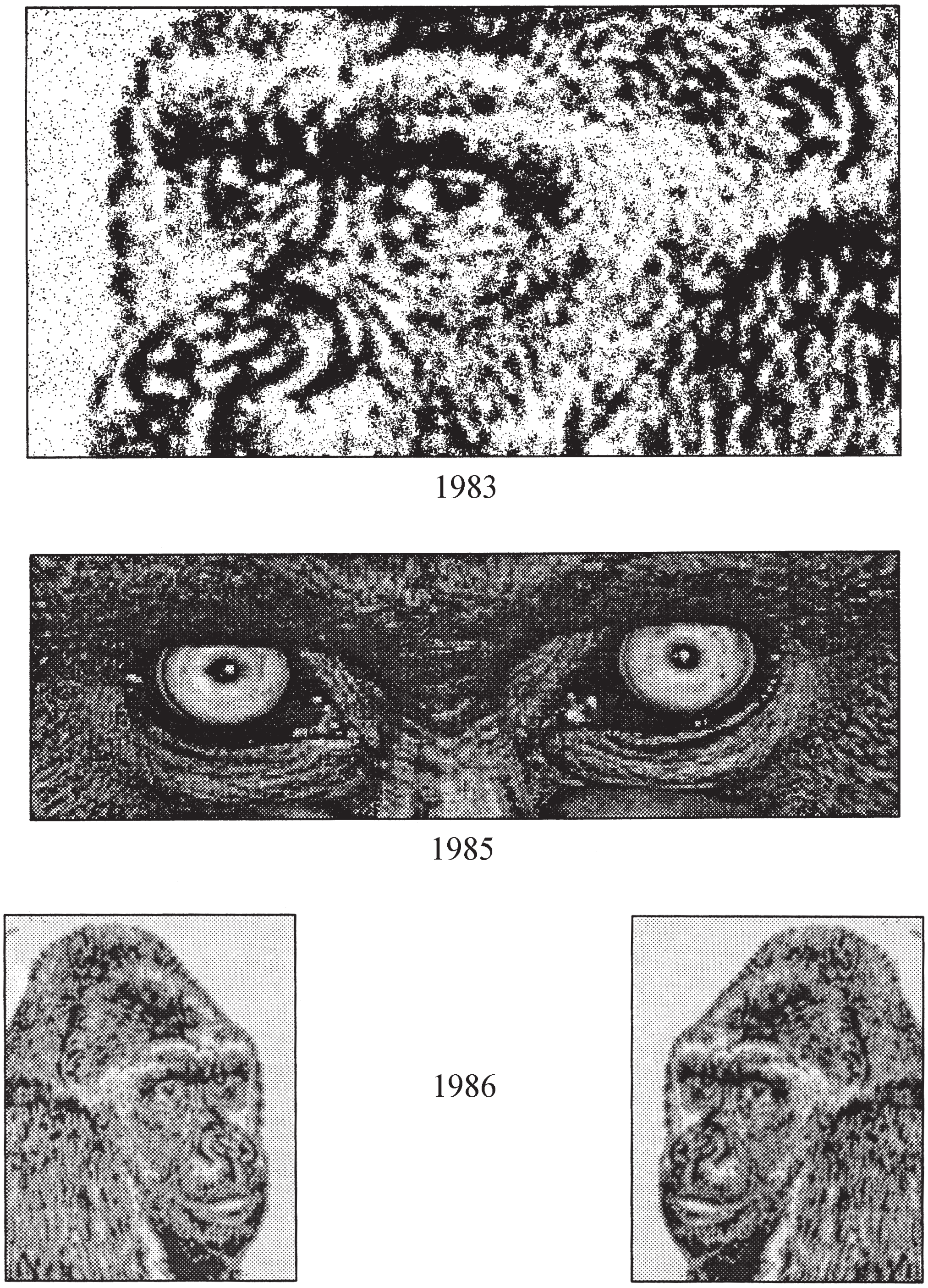


Figure 1: Mailing-page artwork for early *AIPSLetters*.

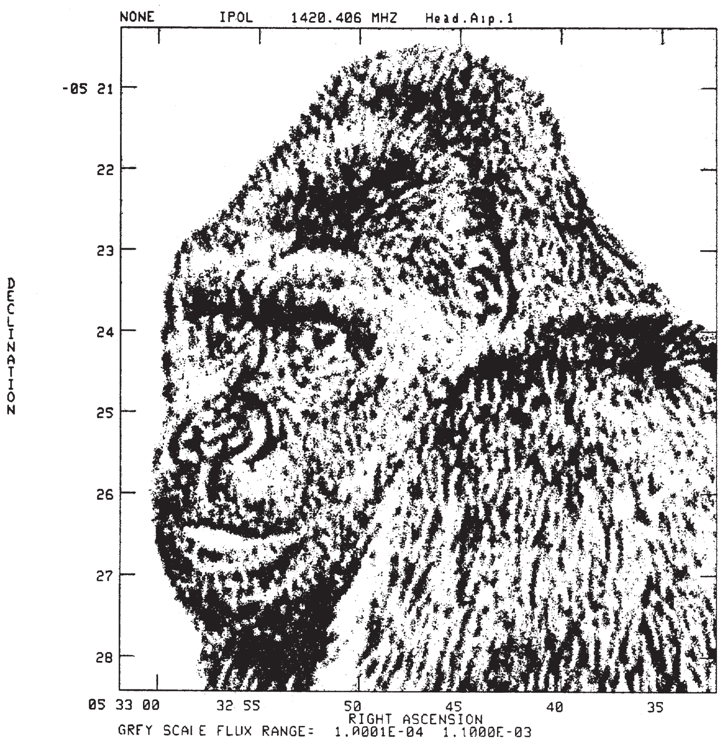


Fig. 11. — « Gorille radioélectrique »(!)

Cette figure illustre à la fois le résultat d'une des formes de traitement d'images VLA, et... le programme utilisé pour y parvenir : « AIPS » (*apes* = singes en anglais).

(Document NRAO)

Pour traiter l'image, c'est-à-dire la nettoyer, mais aussi la modifier à volonté en fonction de certains objectifs, en extraire des chiffres comme des flux ou des positions, etc., un programme spécial, AIPS (Astronomical Image Processing System) a été mis au point. (Ce qui donne lieu à d'amusants jeux de mots, AIPS se prononçant « *apes* », c'est-à-dire « singes » en anglais : la salle de traitement d'images du NRAO à Charlottesville, en Virginie, est communément appelée « AIPS Cage », la « cage aux singes » !) (fig. 11). Avec l'ordinateur approprié, AIPS permet de traiter une image n'importe où; de fait, plusieurs observatoires (dont l'Institut de Radioastronomie Millimétrique à Grenoble et l'Institut d'Astrophysique de Paris) en disposent. Déjà, les observations elles-mêmes peuvent se faire « in absentia »; le temps n'est plus loin où une observation au VLA pourra se faire pratiquement sans quitter son bureau habituel!

Figure 2: Page from Montmerle, T., 1985, "Le Plus Grande Radiotelescope du Monde: Le 'Very Large Array'," l'Astronomie, 99, 487.

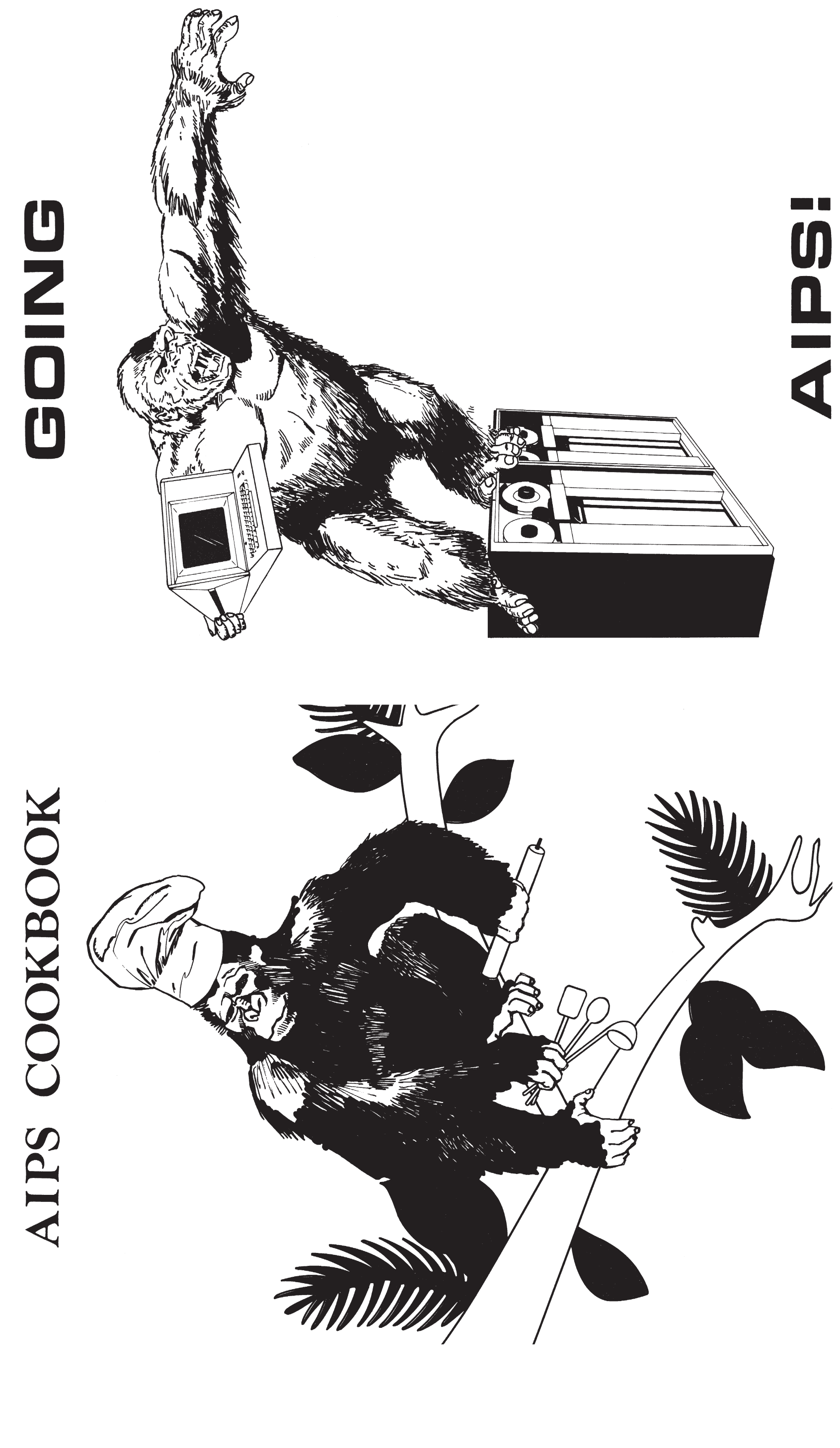


Figure 3: Cover illustrations for the *CookBook* and *Going AIPS*.

2.6. Banana daiquiri

1. Combine in an electric blender: 2 oz. `light rum`, 0.5 oz. `banana liqueur`, 0.5 oz. `lime juice`, 1/2 small `banana` peeled and coarsely chopped, and 1/2 cup crushed `ice`.
2. Blend at high speed until smooth.
3. Pour into large saucer champagne (or similar) glass. Serves one.

4.7. Hot banana soufflé

1. Preheat oven to 375deg.
2. Select a 6-cup soufflé dish or other mold and grease it liberally with 1 tablespoon `butter`.
3. Place 6 `eggs`, 1/2 cup `cream`, juice of 1/2 `lemon`, 1 tablespoon `kirsch`, and 1/4 cup `sugar` in blender. Blend until the batter is smooth.
4. Peel 2 large `bananas`, removing any fibers and break into chunks. With blender running, add the chunks one at a time.
5. Break 11 ounces `cream cheese` into chunks and add them to the blender.
6. When all the ingredients are thoroughly mixed, run the blender at high speed for a few seconds.
7. Pour batter into prepared dish and place it in the hot oven. Bake 45–50 minutes until the top is lightly browned and puffy. You may quit when the center is still a bit soft or continue baking until the center is firm.
8. Serve at once. A whipped cream flavored with Grand Marnier makes a nice topping.

5.4. Bananes rôties

1. Preheat oven to 375 deg.
2. Place 6 (peeled) `bananas` in a baking dish.
3. Sprinkle bananas with juice of 1/2 `lemon`.
4. Pour 2 tablespoons melted `butter` and 2 tablespoons `dark rum` over the bananas. Sprinkle with 2 tablespoons `brown sugar`.
5. Place in oven for 10 minutes.
6. Pour on 2 more tablespoons `melted butter` and 2 more tablespoons `dark rum` and bake for 5 minutes more.
7. Serve at once, spooning some sauce over each banana.

8.4. Golden mousse

1. Combine 1 cup mashed ripe `bananas`, 2 tablespoons `orange juice`, 1/4 cup shredded `coconut`, 3 tablespoons `brown sugar`, a few grains `salt`, and 1/8 teaspoon grated `orange rind`.
2. Whip until stiff 1 cup `heavy cream`.
3. Fold whipped cream into fruit mixture and turn into freezing tray. Freeze rapidly without stirring until firm.

Figure 4: Chapter ending recipes from the September 1983 *CookBook*.

	current *AIPS* group
Ernie Allen	tape and documentation distribution
Bill Cotton	calibration and imaging software, VLB
Phil Diamond	spectral-line software, VLB
Eric Greisen	project design and management, general applications
Kerry Hilldrup	UNIX and Cray systems, Z routines
Nancy Wiener	Gripes, documentation, general assistance
	former *AIPS* group
David Brown	VMS and ModComp systems
Tom Cram	initial design discussions
Gary Fickling	VMS systems, applications software
Ed Fomalont	scientific advisor, applications software
Walter Jaffe	applications and basic software
Thad Polk	geometric corrections software
Gustaf van Morsel	spectral-line analysis software
Don Wells	management and software design advisor
	advisors
Alan Bridle	scientific friend and advisor
Bob Burns	management advisor, Head NRAO Computer Division
Ron Ekers	scientific and management advisor
	software assistance
John Benson	VLB software
Stuart Button	early general applications
Tim Cornwell	mosaicing and maximum entropy tasks
Bob Duquet	super-computer port
David Garrett	preliminary UNIX implementation
Brian Glendenning	SUN image display routines
Jerry Hudson	*POPS* language
Neil Killeen	image analysis tasks
Pat Moore	VLA *AIPS* manager
Arnold Rots	TV display applications
Fred Schwab	self-calibration and other mathematical tasks

Figure 5: *AIPS* participants list circa 1988 from *AIPS* Memo No. 61.

> demands placed by AIPS on any system. To give a few examples, the cost for storing the executable AIPS modules, HELPs, and INPUTS files and a catalog of 15 maps on the disk is in excess of $100 per week. The test runs of APCLN cost about $25 each... The difficulty of running AIPS under these conditions only serves to underscore the need for a dedicated Astronomy Department computer, a point we have of course been making to NSF for years."[13]
>
> "I had hoped that updates would be possible through phone links, but the rate at which code is being modified makes this impractical. In the time from 31 October 1981 to 1 January 1982 more than 5000 blocks of code were modified. Even with 1200 baud line this represents about 6 hours to transmit. At regular long distance rates this is about $200. The link is run by a routine similar to VAXNET and is not totally free from parity errors and dropped characters. The error rate transmitting that much code could be a problem. Clearly tape transport is the most economical way to do a full update. I have used the link to get specific tasks for which we wanted an update as quickly as possible."[14]

AIPS sites were surveyed nearly every year from 1985 through 1990 to determine how AIPS was used. We have not continued the survey since then because desktop workstations are not amenable to measurements of the "fraction of time devoted to AIPS processing" that formed the basis of the earlier surveys' results. In the 1985 to 1990 period, the number of active AIPS computers rose from 54 to 345.[15] More importantly, the total computing power running AIPS full time went from 9.1 to 164.7 in units of VAX 11/780s with array processor. The fraction of that power outside the NRAO went from 51% to 86%. These numbers are a clear measure of the success of the software portability strategy. (Modern users may wish to note that a VAX 11/780 with array processor was very approximately 0.3 AIPSMarks and that modern PCs have been measured to have performances around 10 AIPSMarks, close to a Cray X-MP/4 which used to cost in excess of $10 million.)[16]

The AIPS group, primarily because it needed all the computing power it could get, kept abreast of developments in computer hardware and attempted to make AIPS available to run on it. This began with the VAX at the beginning of 1980. A port of AIPS to IBM mainframes under OS was begun late in 1981, helped us find numerous problems in the code, and was even partly successful by July 1982. That was abandoned in September in favor of a port to a Unix operating system provided by Amdahl for IBM mainframes. AIPS' dependency on correct Fortran compilers was soon apparent, as were layers upon layers of bugs in Amdahl's compiler. These were not solved, and the port declared successful, until June 1984. David Garrett of the University of Texas, beginning in May 1982, had also ported AIPS to run under Unix but on a VAX 11/780. These were but two of many flavors of Unix which forced us to have many versions of some of the Z routines. As a result, it took another year for the Unix versions to be merged, to be regarded as reasonably "standard," and to be shipped to a variety of sites. The surveys mentioned above measured the change from dependency on VMS in 1985 to a preponderance of Unix systems, at least as measured by computing power in 1990. We also went after "big iron" as the super-computers were known. These vector computers were interesting because the pseudo-AP (array processor emulation) routines, which had been in AIPS since 1981, were readily adapted to, and highly vectorized by, big-iron compilers.[17] A port to the Cray-1 of the Minnesota Supercomputer Institute was in progress by September 1984, worked on in part by Bob Garwood who is now at NRAO. Time on the Cray X-MP at Digital Productions in Los Angeles was made available to the NRAO under an NSF supercomputer initiative. Bob Duquet and Kerry Hilldrup began work on this project in early 1985 and, by the October 1985 *AIPSLetter*, it was considered functional. The cpu times achieved were 15–50 times better than a VAX 11/780 with array processor, but the *real* times taken to run the programs were distressingly long (about the same as the VAX plus AP). Fortunately, in

[13] Frank Ghigo, University of Minnesota.
[14] Stuart Button, University of Toronto.
[15] Bridle, A. and Wiener, N., AIPS Memo No. 59, "The 1988 AIPS Site Survey," March, 1989 and Bridle, A. and Nance, J.., AIPS Memo No. 70, "The 1990 AIPS Site Survey," April, 1991.
[16] Greisen, E. W. AIPS Memo No. 85, "DDT Revised and AIPSMark Measurements," February 1994.
[17] Wells, D. C., Cotton, W. D., AIPS Memo No. 33, "Gridding Synthesis Data on Vector Machines," January 1985. See also Wells, D. C., AIPS Memo No. 47, "Installing NRAO's AIPS on vector Computers," June, 1985.

1985, both Convex and Alliant Computer Corporations announced vector or vector/parallel computers that were a lot cheaper to buy and to operate than Crays. We tested these computers in 1985[18] and bought one for Charlottesville's Christmas 1985 to replace the IBM. A second Convex C-1 was obtained for the VLA in January 1987. These were very powerful computers, but they were still a central, shared machine with all the attendant troubles related to inadequate disk space, sign-up sheets, and the like. The breakthrough to the modern era of computers on everyone's desk began when *AIPS* was ported to a Sun-3 in Princeton in October 1986. We found that our user community was reluctant to trust this port, so Don Wells persuaded Sun to loan me a Sun 3/110 for my desk. This was put to work late in 1987, developing the final parts of the Sun Screen Server implementation of an *AIPS* TV display written by Brian Glendenning, then of the University of Toronto. By the time of the code overhaul (see below), the Sun workstation was regarded as the best platform on which to do the initial debugging of a full code re-write.

All of the computer testing and evaluation done from 1985 to the present at NRAO has depended on a certification and benchmarking suite developed initially by Don Wells. This suite is implemented in *AIPS* procedures written in `RUN` files to execute a sequence of *AIPS* tasks on standard data sets, comparing the results with previously computed answers. This suite, called `DDT`,[19] was first described in 1985[20] and has been the subject (or tool) of numerous Memos thereafter.[21] `DDT` enables us to insure that the principal tasks run correctly on new computers and new versions of *AIPS* and to measure the performance of a computer as a typical *AIPS* user would see it.

The *AIPS* group also tried to stay abreast of developments in computer networking. We actually had a computer network established between our ModComp Classic and VAX 11/780 in Charlottesville. It was used to copy text files back and forth and was faster than magnetic tape, but only when the ModComp would not get tired of waiting for the VAX's slow operating system. There were a number of experiments with DECNET and, eventually, a link was established between Charlottesville and the VLA using a leased phone line. At that point we were finally able to keep the code in New Mexico current with the code in Charlottesville. Electronic mail has become important in the project. The first e-mail address announced for the group in October 1985 was `nancy%cvax%deimoscaltech.bitnet` or `cit-hamlet.arpa`. This used a dedicated phone line from CalTech to NRAO in Tucson which was rented to support the work at Digital Productions. We finally wrote an article in the *AIPSLetter* in July 1986 describing our e-mail connectivity through four different networks. In June 1987, we announced "exploding bananas," an e-mail forwarding system that would allow subscribers to receive all e-mail discussions of topics affecting *AIPS*. By now, most of our users receive their *AIPSLetter* and even their copies of the *AIPS* source code and binaries directly through the Internet and World-Wide Web.

Snapshots of the *AIPS* code were given to users whenever they asked until September 1982. At that time, we introduced the concept of frozen releases named, *e.g.*, `15OCT82`, which would be shipped to non-NRAO sites. Releases were done every 2 months until July 1984 when the schedule became every 3 months. Beginning in April 1985, we introduced the concept of `OLD` (shipped), `NEW` (bug fixes only), and `TST` (active development) versions, with a "midnight job" that kept the VLA copies of *AIPS* current with those in Charlottesville. The *AIPS* code remained in the complicated Fortran 66 dialect discussed above until a "code overhaul." This overhaul was announced in April 1987, begun in July 1988 (after a serious reorganization of the `Z` routines), and released as the `15OCT89` version. The overhaul was begun with a powerful text transformation program written by Bill Cotton and driven by a long symbolic list of the transformations desired. Unfortunately, the output of this program still required manual intervention to convert the code to our new standards of function and legibility. The result, however, was code in ANSI-standard Fortran 77 supported by an *AIPS* pre-processor program to allow `INCLUDE` files and `HOLLERITH` variables which are not supported by all compilers.

---

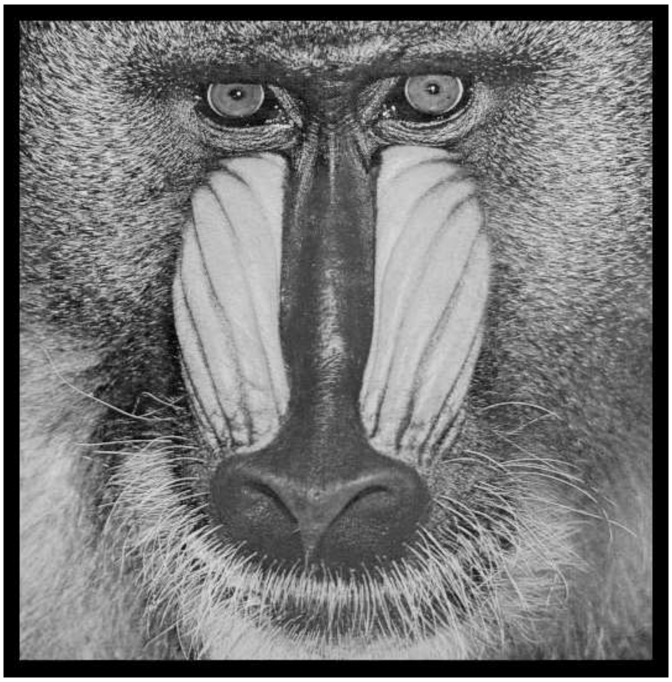
[18] Hilldrup, K. C., Wells, D. C., Cotton, W. D., *AIPS* Memo No. 38, "Certification and Benchmarking of *AIPS* on the Convex C-1 and Alliant FX/8," November 1985.

[19] A typical *AIPS* play on words referring to the "dirty dozen" *AIPS* tasks used, the bug killing aspects of the insecticide, and the macho endurance of the characters in the movie by that name.

[20] Wells, D. C., Fickling, G. A., Cotton, W. D., *AIPS* Memo No. 36, "Certification and Benchmarking of *AIPS* on the VAX-8600," June 1985.

[21] See in particular Langston, G., Murphy, P., Schlemmer, D., *AIPS* Memo No. 73, "*AIPS* DDT History," May, 1991.

AIPS was originally conceived as a "map-processing" package to read in *uv* data only for the purpose of doing the gridding and Fourier transformation. The original code handled continuum images and read *uv* data from Export-format tapes directly into the imaging software, primarily `APMAP`. However, by the middle of 1981, the desire to do self-calibration on VLA data and to begin to do some processing of VLBI data caused the project to develop a format and input/output routines for *uv* data on disk and a variety of tasks to handle these data. Tasks such as `UVLOD`, `UVFLG`, `UVMAP`, and `UVSUB` date from these early days. At the same time, the map input/output routines were revised to handle multi-dimensional images and tasks like `MCUBE` and `TRANS` were written. VLBI applications began appearing in March 1982 with a full global fringe-fitting task (`VBFIT`) in May 1982.[22] User competition for scarce resources led to accounting and `TIMDEST` (automatic deletion of old files) in January 1982, lock files for tape and display devices in May 1982, a roller for array processor tasks in September 1982, a full queueing algorithm for these tasks in September 1983, and task `NOBAT` in March 1984. This last task did not actually use the AP, but it allowed a higher-priority user to block lower-priority users from the device. *AIPS*' high standards for handling celestial and other coordinates began in May 1983. The first openly interactive task, `XGAUS`, appeared in September 1983. The multi-field, multi-channel, *uv*-data-based imaging and Cleaning task `MX` first appeared in November 1983. This "battery-powered Clean" algorithm was developed by Bill Cotton and Fred Schwab as an enhancement of the Clark Clean algorithm. The Fourier transform of the Clean-component model is subtracted from the *uv* data and the residual data are re-gridded and transformed. In this way, problems related to gridding and aliasing are sharply reduced and a larger field of view may be used. A calibration package for *AIPS* was begun in July 1984, but did not appear until the `15JAN87` release. The package was limited initially to continuum calibrations; polarization calibration appeared in `15OCT87` and spectral-line bandpass calibration arrived with `15JAN88`. The task `TVFLG` appeared in October 1987 to allow users to edit data interactively with the TV display. The calibration package did not get much use outside of Charlottesville until after the code overhaul and the decomissioning of the VLA's DEC-10. At that point, *AIPS* development and politics definitely became "interesting," but that is the subject of another manuscript, perhaps one entitled "*AIPS* in its Later Years or Freon is in Short Supply."



Title page illustration for the `15APR98` *CookBook*.

[22] Schwab, F. R., VLBA Memo No. 82, "Global Fringe Search Techniques for VLBI," April 1982.