An autobiographical essay

Kenneth E. Iverson

Ken and Donald McIntyre worked on what was to be *The Story of APL & J* in 2004 through a series of e-mail messages, with the last e-mail from Ken coming in the morning of Saturday, 2004-10-16. Now Donald McIntyre's health prevents him continuing work on *The Story of APL & J*. The following autobiographical sketch has been extracted from the manuscript, which *Vector* hopes eventually to publish in its entirety. Donald McIntyre emphasises that the text of the sketch is in Ken's own words. *Vector* is grateful for Roger Hui's help with preparing it and for adding the endnotes. *Ed.*

Preamble

My friend Dr Donald McIntyre has a penchant for well-documented historical treatments of topics that interest him. His more important works concern his chosen discipline of geology; notably his discovery and discussion of the lost drawings of James Hutton, and his commemorative works on the occasion of Hutton's bicentennial.

Because of his fruitful use of my APL programming language, and its derivative language J, Donald has asked me many questions concerning their development. I finally suggested (or perhaps agreed to) the writing of a few thoughts on these and related matters.

Because of my other interests in developing and applying J, I have deferred work on these essays, but now realise that the further application of J has already fallen into younger and better hands, such as those of Professor Clifford Reiter in his *Fractals, Visualization, and J* [1].

Likewise, the further development and implementation of J are now in better hands, such as those of Roger Hui, my son Eric and nephew Kirk, and, last but not least, Chris Burke.

Because of my negligence in the keeping of records, I am relieved to realise that I can now in good conscience ignore the provision of carefully-documented references, leaving such matters to McIntyre's expertise. I will also exercise the freedom to explore ideas as they arise, and not restrict myself to fulfilling Donald's expressed desires.

I will begin with a series of topics chosen more or less at random, and will defer the matter of an overall organisation.

Schooling

I started school on 1 April 1926, eight months before my sixth birthday, was promoted to Grade 2 at the end of the school year in June, and was promoted to Grade 4 at the end of the next year.

As I understand it, my father chose the inauspicious day and date as a joke on the elderly and superstitious schoolmaster (Mr McLeod), who reciprocated with his own joke by promoting me to Grade two after just three months. I was, of course, far from ready, but since he was about to retire, I was left as a problem for the new and much younger Mr Norman Bowles.

To my surprise, Mr Bowles put me in neither Grade 1 nor 2, but kept me separate. I recall my eager anticipation of being placed in Grade 2 when (after about a month) I reached their reading lesson. I was disappointed to find that nothing changed, but was rewarded by promotion to Grade 4 at the end of the year, and was ready to leave the school to enter Grade 9 at age 12.

All this was possible in a one-room school (with one teacher and some 30 students). Younger students could overhear the work of their elders, who might also be asked to help them. I also had much help from my older sister Aleda, who herself went on to become a teacher.

The timing was fortunate, because progress soon overtook the rural province, the one-room schools disappeared, and local children were bussed some fifteen miles in snowy Alberta winters.

For Grade 9 I had to go to the nearest village about 14 miles away, and stayed with family friends on a farm only 5 miles from the school. As a 12-year-old I was a social misfit in high school, but completed the year with a good start on Grade-10 subjects.

Although I enjoyed school, I chose to quit at the end of the year to work at home on the farm. In those years of the Great Depression my choice was perhaps a relief to my family. But my reasons were simple: as far as I knew, the only purpose of further schooling was to become a schoolteacher, and that I decidedly did not want. I knew absolutely nothing about universities and the preparation they provided for careers quite unknown to me.

I finally learned about universities from my Air Force mates, many of whom planned to return to university, thanks to government support for servicemen. In fact, one of my buddies threatened to return and beat my brains out if I did not grasp the opportunity.

When I quit school, my younger brother Byron was sent to school in the city of Edmonton, staying with an aunt and uncle. I asked him to borrow science books for me from the city library, and he chose some on radio – sparking my interest.

In 1938 at the age of 17 I enrolled in a correspondence course with De Forest's Training of Chicago, at a cost of $205. Fortunately, my elder brother Elmer chose to do likewise – it was very helpful to have someone to work with.

We desperately wanted radio parts to work with, and persuaded Dad to buy a radio that would run on a re-chargeable automobile battery rather than on expensive 135-volt dry cell batteries – and, more importantly, to let us have the old radio.

We then began to dissect it, carefully making a circuit diagram as we went. I find it interesting to recall our difficulties in identifying parts (such as "condensers" or "capacitors"), having studied them in the abstract, but never having seen one on the hoof.

The correspondence course culminated in two weeks practical training in Chicago, a thrilling time for us, made more so by our uncle Ingmar with whom we stayed – he loved Chicago, and made sure that we saw all we could of it (including giant presses at The Chicago Tribune) in the time available.

Looking for further reading after completing that course, I soon realised that the really interesting books on radio and electricity used calculus. On a visit to Edmonton, I looked for a book on the subject, and came away with a used copy of *Calculus* by Herman W. March and Henry C. Wolff [2].

The long cold Alberta winters sometimes made outdoor farm work impossible, and granted me the leisure to study M&W. My most vivid memory is of the joy in discovering how the beautiful circular functions were finally united in a single family under the exponential.

I think it worth contrasting this joy with the reaction of my youngest brother (Clayton Ray – no less competent than I) to his encounter with the same material in a university course. I was disturbed that he didn't share my joy in it, but dismissed it as "just more formulas".

I was drafted into the army in 1942 and joined the air force in 1943. The Canadian Legion offered correspondence courses to men in military service, and I took eight of them, nearly enough to complete my high school. Three things I remember – the amazing patience of my tutors, the fact that I never met another serviceman who took courses, and the useful discipline of studying a text with no immediate recourse to a teacher.

Years later I was pleased to find that my own work in APL was exploited by a Mr Clementi who, some 40 years ago, chose to use it in a correspondence course in Australia because of its brevity. This brevity made it feasible to enter a student's written submission, run it (perhaps with modifications to make it work, or otherwise improve it), and mail the typed result from the computer terminal.

After my discharge in 1946, I enrolled at Queen's University in Kingston (Math and Physics in the Arts Faculty). My experience as an undergraduate at Queen's University in Kingston was unusual in that about half the class consisted of returning veterans: the faculty was overjoyed to find students so serious about their work, and the non-veterans were resentful of the "unfair competition".

I rather resented the requirement to take two courses quite outside of my main interest; one in English and one in Philosophy. I found both most rewarding, and am left with a firm belief in forcing students to look at some areas that they are confident are of no interest to them.

Physics labs required a lot of time, and taught me two things. The first was that they were designed only to confirm things we "already knew" from theory, and were not "experiments" in the sense of discovering anything. This was a lesson I made use of in later work (at IBM) in designing "computer experiments".

The second was the importance of recording results in pencil, so that they could be fudged to show reasonable results in the required lab reports. A close friend among the younger students learned this the hard way (in a lab designed to measure the difference in the heat capacity of air at constant volume and at constant pressure).

Because of "bad" results he had to repeat the experiment. His new bad results infuriated the professor in charge, who came in on a Saturday to watch the whole process. The upshot was that he re-analysed the experiment, concluding that secondary effects (such as draughts from an open window) would mask the intended effect – in spite of which, students had been reporting "good" results for a full five years.

I also learned something of the difference between detailed meticulous teaching, and good teaching. In physics, Professor Cave was meticulous in copying from his notebook only the main steps in a proof, making brief comments on the algebra or calculus needed to connect them, but leaving students to fill in details for themselves. This we did, to great advantage.

In math, Professor Jeffrey (head of the department) achieved much the same, though inadvertently. Beginning a lecture with an obviously well-planned approach, Professor Jeffrey (head of the Mathematics Department) might soon pace the stage and say "I'll bet we could do it this way". The lecture often ended in some confusion, and our subsequent work to unsnarl it taught us a good deal. I remember the class as the first time I actually saw a mathematician at work.

Jeffrey urged me to go on to graduate school – an idea I accepted because the veteran's benefit would pay, and because I had no idea what else to do. But, when I proposed to stay on with the newly formed graduate program at Queen's, he objected, saying that in my four years I had absorbed their point of view, and should move on.

Acknowledging his point, I declined the advice to move because I was already 30 years old, and had the responsibility of a wife and two children. Jeffrey restored my perspective by remarking that 30 was his age at beginning graduate school, that he and his wife had no children and still did not, but did have two cats.

In the event, on my graduation in 1950 I went to the math department at Harvard, a move I have never regretted. Although I got respectable grades, I found the math department a very cold place, perhaps in recognition of my limited potential for math. Anyway, I got my Masters degree in one year, and then switched to the Department of Engineering and Applied Physics, after being attracted by a course that I took with Professor Howard Aiken – called "Switching Theory" I think.

Although press reports of the new "giant brains" were beginning to appear, I had no idea of the computer work going on at Harvard under Aiken. Although he worked in a different division, I chose to attend his course, and was quickly captivated. It was the first time as a student that I was given the impression that there might be new work to be done.

Although we sometimes had as many as six students to a desk, Aiken squeezed his graduate students into his Laboratory. To us it made an enormous difference to finally be "on the inside", with significant access to faculty, and with good opportunities to serve as Teaching Fellows in courses.

Aiken quickly arranged a thesis topic for me by introducing me to economics Professor Leontief, whose graduating student had used Aiken's computer in his work in Input-Output Analysis. I was expected to extend the work to handle "Capital Goods"; mathematically an extension to systems of differential equations.

I believe that Aiken did a great deal to encourage clear writing by his students, and with relatively little effort on his part. He (deliberately, I expect) developed a reputation for fierce reading of drafts of theses, with the result that we all asked our fellow students to criticise our work before approaching Aiken.

When he read my first draft, I found him both helpful and gentle, often asking if I was sure that this was just the right word, when he knew full well that it was not.

Teaching I

When I received my degree in 1954, Aiken got me appointed to the faculty as an Instructor to help man his new Masters program in Automatic Data Processing. Except for a 6-month “mini-sabbatical” to work for McKinsey, I continued until 1960.

At the time that I graduated in 1954, Aiken was advising that universities should get out of the business of designing and building computers, and should turn their attention to the applications of computers. He argued that companies such as IBM and Remington Rand already recognised a business opportunity, and were in a better position than universities to address such problems as quality control.

Against strong opposition from the Harvard administration, Aiken managed to introduce a Masters program in Automatic Data Processing in 1955; in effect, the first computer science program. I was one of four of his graduating students that Aiken was able to get appointed to the position of Instructor to man his program. A thorough account of these years is included in *Howard Aiken: Portrait of a Computer Pioneer* [3], by the Harvard historian I. Bernard Cohen.

Although Aiken had mapped out a broad program that included economics, business applications, switching theory, operations research, numerical analysis, and computer programming, it was largely left to us green graduate students to flesh out the courses.

I was appalled to find that the mathematical notation on which I had been raised failed to fill the needs of the courses I was assigned, and I began work on extensions to notation that might serve. In particular, I adopted the matrix algebra used in my thesis work, the systematic use of matrices and higher-dimensional arrays (almost) learned in a course in Tensor Analysis rashly taken in my third year at Queen’s, and (eventually) the notion of Operators in the sense introduced by Heaviside in his treatment of Maxwell’s equations.

Harvard had a benign rule of five years to tenure or out, benign because it prevented fierce Bostonophiles from clinging forever in the hope of eventual tenure. It was particularly benign in those postwar days of the establishment of commercial research centres, where anyone from the science faculty of a respectable university could expect to double his salary – as I did when I joined the Research Division of IBM.

Schools of education prepare teachers for elementary schools with a great emphasis on the techniques of teaching. At college level, on the contrary, no preparation for teaching is (or at least was) provided, and graduates, full of their subjects, were unleashed on students. I was fortunate in having as a Teaching Fellow a graduate student who had been born at a podium – Fred Brooks. Moreover, Fred and I were sufficiently close friends that he could, and did, tell me bluntly of my bad teaching techniques in post-mortems held after each lecture. For example, he once said "That was a very interesting point you made about such-and-such – too bad you made it to the blackboard, and could not be heard past the first row."

First test of notation

The first real encouragement of my work in notation came when I took a six-month mini-sabbatical after two years of teaching in Aiken's program, to work for the still-thriving consulting firm McKinsey & Co. This was in the fall of my third year on the faculty.

This occurred because M&Co. had undertaken work that required the use of a computer. Having no computer knowledge themselves, M&Co. turned to the source of all knowledge (Harvard, and eventually Aiken), and Aiken recommended me.

When I arrived in San Francisco, Ted Strong (of M&Co.) explained that his client, Hawaiian Sugar (in a dispute over shipping rates with Matson Lines), had embarked on a thorough analysis of their options – including, for example, the choice of locating refineries in Hawaii or in the US. This required a computer, and Ted had secured the services of Bob Oakford and Dan Fisher of Stanford University, to program the (then rather new) IBM 650.

Ted held weekly meetings with Oakford to communicate his remarkably-detailed knowledge of sugar and shipping. These meetings seemed to go well, but he was unable to get anything understandable about the progress of the programming. I asked Ted to do the same for me, but recorded the steps and decisions in the notation I had been developing for teaching, notation that was eventually implemented as APL.

The encouraging thing was that we were able to converse in accurate detail, Ted knowing nothing of computers, and I knowing nothing of sugar and shipping. Moreover, Oakford and Fisher were able to understand these same programs, just as I was able to understand their 650 programs. In a few months the 650 programs were nearly complete, and we four eagerly anticipated their use – whereupon the project was abruptly terminated.

This was a great disappointment, even though the reason was victory – all this computer stuff had intimidated Matson to agree to a re-negotiation of rates.

IBM Research Center

I left Harvard with offers from IBM Research and Bell Laboratories. Fred Brooks (who had joined IBM somewhat earlier) advised me to choose IBM because computers were their primary concern. I did not join IBM until 1960, at which time I was just finishing up my *A Programming Language* [4], which included a chapter (called *Microprogramming*) on the formal description of the IBM 7090 machine. After I joined the IBM Research Division, Fred advised that I stick to whatever I really wanted to do, because management was so starved for ideas that anything not clearly crazy would find support. In particular, I was allowed to finish and publish *A Programming Language* [4], as well as *Automatic Data Processing* [5] with Fred Brooks as co-author.

I soon met Adin Falkoff, and we worked jointly on the notation for some twenty years. It was Adin who suggested [6] the name *APL* (from the title of my 1962 book). In the 1962 book, I had used APL to describe an IBM computer, and Adin and I (with Ed Sussenguth) used it to produce a formal description [7] of the IBM System/360 computers then under design.

An important consequence of this work (published by John Lawrence in the IBM Systems Journal) was that it attracted important contributors to APL, notably Larry Breed (a Stanford student, who joined us at IBM and chose to undertake the crucial work of implementing it as a programming language), and Donald McIntyre, head of Geology at Pomona College (who had acquired a 360 Model 40 for Pomona, and used the formal description to become more expert than the IBM Systems Engineer assigned to Pomona).

Teaching II

I believed that APL could be used in teaching, and Adin said that to test the point we must take a text used in the State school system, and try to teach the material in it. He further proposed that we invite active high school teachers.

We hired six for the summer, with the plan that two (nuns from a local school, who could provide a classroom in which we supplied a computer [typewriter] terminal) would do the teaching, while the other four (with a two-week head start) would write material.

To our surprise, the two teachers worked at the blackboard in their accustomed manner, except that they used a mixture of APL and conventional notation. Only when they and the class had worked out a program for some matter in the text would they call on some (eager) volunteer to use the terminal. The printed result was then examined; if it did not give the expected result, they returned to the blackboard to refine it.

There were also surprises in the writing. Although the great utility of matrices was recognised (as in a 3-by-2 to represent a triangle), there was a great reluctance to use them because the concept was considered to be too difficult.

Linda Alvord said to introduce the matrix as an outer product – an idea that the rest of us thought outrageous, until Linda pointed out that the kids already knew the idea from familiar addition and multiplication tables.

Finally, it was this interest in teaching that led us to recruit Paul Berry, after seeing his “Pretending to Have (or to Be) a Computer as a Strategy in Teaching” [14] when it appeared in *Harvard Educational Review*.

Rigidity of viewpoint

It was a revelation to see that the outer product (adopted from Tensor Analysis) could appear so simple from a different point of view. However, this was a lesson that had to be learned again and again. A few examples:

- We became so wedded to the power of operators that we made complex and awkward attempts [8] to use them to achieve the effects of `f+g` and `f*g` as used in calculus texts. One such proposal [9] was made by me and Arthur Whitney at an APL conference – probably in Heidelberg.

- On the return flight from the 1988 APL conference in Australia, Gene McDonnell and I worked out what came to be called *fork* and *hook* [10] – realising that the forms `f+g` and `f*g` were not used in APL, and could be introduced without conflict.

- But again it took some time to adopt the more general, and more accessible, notion of a train, and to suppress discussion of forks. It is more accessible because this sense of *train* occurs in English dictionaries: “An orderly succession of related events or thoughts; a sequence”.

- Similar rigidity appeared in our use of the terms *vector*, *matrix*, and *higher-dimensional array* instead of the accurate and familiar *list*, *table*, and *report* [11]. The last provides a completely general term, which may be qualified by *rank-1*, *rank-2*, etc., and even by *rank-0* for a scalar.

- The notion of the dual of a function [12] (or system) is clearly important, because it is invoked in many areas of mathematics. It was not until I realised that the dual of a function `f` was more properly expressed as "the dual of `f` with respect to a second function `g`" that I saw the possibility of using a conjunction `&.` in the present formulation: `f&.g` is `(g inverse)` of `f` of `g`. This has found wide application, and the conjunction is called *under* – from analogies such as "surgery under anaesthetic". The general idea is that many (if not most) procedures (not only in mathematics) must be performed after some preparation `g` and finally the effects of `g` must be undone.

IBM Scientific Center

As manager of IBM's Scientific Centers, Joe Mount invited us to open a centre devoted to APL. We chose Philadelphia, and moved the entire group, remaining for about seven years.

These were very fruitful years, beginning with the organisation of the first APL conference in Binghamton, NY in mid-1969 (dubbed "The March on Armonk" by Garth Foster of Syracuse University, because of the perceived neglect of APL by IBM headquarters). This conference attracted a number of new people, including Donald McIntyre – who had been "sent" by IBM because of Pomona's Model 40 and his expert knowledge of it through studying the APL description of the 360 computers. In spite of a busy schedule, Donald spent a short time with the APL group the same summer.

Teaching III

My daughter Janet attended Swarthmore High School, and recommended Rudy Amann (head of the math department) as an excellent teacher. I therefore approached him with a proposal that we put an APL terminal in his school as a tool for teaching mathematics, suggesting that he first spend the summer with the APL group to assess the matter.

Rudy responded that he could spend only two weeks, which he did. I gave him an office with a terminal (and the calculus text in APL [13] that I had written after our earlier experiment with high school teachers), and invited him to come to me or anyone in the group with questions. Since he never stirred from his office, I despaired, but at the end of the two weeks he announced that he wished to go ahead with the project.

Rudy was pleased with the results, and told me of canvassing those of his students who went on to college, finding that they were pleased with the preparation he had given them. One thing he had done was to use some of the final two "review" weeks to show them the translation from things like `+/` to the sigma notation they would encounter in college.

Rudy continued through a second year, but the inevitable occurred – he was promoted to principal, and had been unable to interest any other teacher in the math department in continuing.

About this time we hired Paul Berry. Paul had degrees in psychology (including a PhD), adding to the rather wide diversity in the early APL crew:

- Chemistry: Adin Falkoff
- Math and Physics: me
- English: Gene McDonnell, Graham Driscoll
- Computer Science: Larry Breed, Roger Moore, Phil Abrams
- Mechanical Engineering: Richard Lathwell
- Secretary: Colleen Conroy (learned APL and promoted to Programmer)
- Other: Ziad Ghandour who, on a trip to his native Lebanon, contributed the Arabic in one of the documents, saying that APL should have originated in Mecca.

I.P. Sharp Associates

I.P. Sharp Associates of Toronto was a pioneer user of APL, establishing it as the base of a time-sharing service widely used in Canada, U.S., and Europe. This came about because Larry Breed had proposed that we hire Roger Moore as a consultant to aid in his implementation of APL. Roger was his classmate at Stanford, and a senior member of I.P. Sharp – he was impressed enough by APL to propose that his firm adopt it.

Having effectively lost control of APL to the Palo Alto Scientific Center (under Horace Flatt), I left IBM and joined Sharp. It was exhilarating to work for a company seriously devoted to APL. Moreover, the seven years to my next retirement (at 67) saw many enhancements to the APL system, largely under the direction of my son Eric.

His group was largely made up of young men who had come to Canada from U.S. to escape service in Vietnam. Their beards and otherwise scruffy appearance led a visiting businessman to say “This place is a zoo.” Rather than resent the implied insult, the group proudly adopted the name, and were known thereafter as “The Zoo”.

My work that proved most important to the later development of J was the publication of my “A Dictionary of APL” [15]. In particular, it proposed a parsing scheme that depended on the “first four elements only” of an execution stack.

Parsing was controlled by a brief four-column table of cases – the first row that agreed with the first four elements of the execution stack determined what action was to be taken.

Roger Hui later told me that during a six-month period off work he had had the leisure to study this paper carefully, and it was this study [16] that led to our collaboration on J after my retirement from Sharp.

Second retirement

In my “A Personal View of APL” [17] I said:

> … the initial motive for developing APL was to provide a tool for writing and teaching. Although APL has been exploited mostly in commercial programming, I continue to believe that its most important use remains to be exploited: as a simple, precise, executable notation for the teaching of a wide range of subjects.

When I retired from paid employment, I turned my attention back to this matter, and soon concluded that the essential tool required was a dialect of APL that:

- is available as “shareware”, and is inexpensive enough to be acquired by students as well as by schools
- Can be printed on standard printers
- Runs on a wide variety of computers
- Provides the simplicity and generality of the latest thinking in APL

The result has been J, first reported in the *APL90 Proceedings* [18].

In *Incunabulum* [19], an appendix to his *An Implementation of J* [20], Roger Hui describes the inception of work on J as follows:

> One summer weekend in 1989, Arthur Whitney visited Ken Iverson at Kiln farm and produced – on one page and in one afternoon – an interpreter fragment on the AT&T 3B1 computer. I studied this interpreter for about a week for its organisation and programming style; and on Sunday, August 27, 1989, at about four o’clock in the afternoon, wrote the first line of code that became the implementation described in this book.

Roger’s acknowledgement in this book was the most extravagant I have ever received: *Ex ungue leonem*.

Details of the development of J can be seen in every release by clicking on the Help menu, and then on Release Highlights. For example, from the J 4.01 Release:

- Use `=:` for global definitions in scripts. Run → Window and Run → File use `load` and definitions made with `=.` are local to `load` and disappear when it finishes.
- The J file suffix has changed from `.j?` to `.ij?` (`.js` to `.ijs`) to avoid javascript conflicts.
- Use `system\extras\migrate\ext.ijs` if you have lots of files.
- system directory contains the other distributed directories (`main\stdlib.js` is now `system\main\stdlib.ijs`).

Chris Burke and my son Eric soon became interested in J, and, among many other things, designed and implemented a user interface (GUI) to the Windows operating system. The aspect of their work that I most appreciate is the Labs and Lab Authoring system – which is, unfortunately, not yet widely used.

Teaching IV

Since J is now free, and uses only ASCII characters, the requirements set forth in my "A Personal View of APL" have been fully met. Moreover, a few have used it seriously at college level (notably, Clifford A. Reiter of Lafayette College [23]), but interest in such work spreads slowly.

But my attempts at interesting schools at lower levels have had no result – if it is not in the approved curriculum, it doesn't exist. However, I have gained important teaching experience by using J in a workshop called "Exploring Math", presented to classes of retired folk drawn from a wide variety of non-mathematical backgrounds, including teachers at all levels, lawyers, medical doctors, and psychiatrists.

This workshop suffered from the fact that few had easy access to a computer, a situation that has now changed radically for the better. More importantly, it suffered from the fact that I drastically over-estimated the speed at which they could overcome their fear of mathematics and assimilate ideas foreign to their experience. As a result, the class soon dropped to three, but it reached the point where the work was largely directed by their interests and resulting questions – which we explored at length using J.

For this workshop, I wrote *Math for the Layman* [21], basing it on the idea of mathematics as a language, as expressed in Lancelot Hogben's still-popular *Mathematics for the Millions* [22]:

The view which we shall explore is that mathematics is the language of size, shape and order and that is an essential part of the equipment of an intelligent citizen to understand this language. If the rules of Mathematics are the rules of grammar, there is no stupidity involved when we fail to see that a mathematical truth is obvious. The rules of ordinary grammar are not obvious. They have to be learned. They are not eternal truths. They are conveniences without whose aid truths about the sorts of things in the world cannot be communicated from one person to another.

References

[1] *Fractals, Visualization, and J*, Cliff Reiter, 3rd edn, 2007, ISBN 978-1-4303-1980-1, Lulu.com http://www.lulu.com/content/635966

[2] *Calculus*, H.W. March & H.C. Wolff, McGraw-Hill, 1917

[3] *Howard Aiken: Portrait of a Computer Pioneer*, I. Bernard Cohen, MIT Press, 2000, ISBN 0262531798

[4] *A Programming Language*, Kenneth E. Iverson, Wiley, 1962

[5] *Automatic Data Processing*, Fred Brooks & Kenneth E. Iverson, Wiley, 1963

[6] *A Source Book in APL*, http://www.jsoftware.com/jwiki/Doc/A_Source_Book_in_APL#origins_of_APL

[7] "A Formal Description of SYSTEM/360", A.D. Falkoff, K.E. Iverson & E.H. Sussenguth, *IBM Systems Journal*, 3, N°3, 1964 http://www.research.ibm.com/journal/sj/032/falkoff.pdf

[8] *Remembering Ken Iverson*, Roger Hui, 2004, http://keiapl.org/rhui/remember.htm#fork0

[9] "Practical uses of a model of APL", Kenneth E. Iverson & Arthur T. Whitney, *ACM SIGAPL Quote-Quad*, 13, N°1, September 1982, ISSN 0163-6006, http://portal.acm.org/citation.cfm?xml:id=390006.802236

[10] "Phrasal forms", Eugene E. McDonnell & Kenneth E. Iverson, in *Conference Proceedings on APL as a Tool of Thought*, 1989, ISBN 0-89791-327-2, http://portal.acm.org/citation.cfm?xml:id=75172

[11] *Nouns*, in *The J Dictionary*, http://www.jsoftware.com/help/dictionary/dicta.htm

[12] *Ken Iverson Quotations and Anecdotes*, Roger Hui, http://keiapl.org/anec/#under0

[13] *Elementary Functions: An algorithmic treatment*, Kenneth E. Iverson, 1966, Science Research Associates, http://www.jsoftware.com/jwiki/Doc/Elementary_Functions_An_Algorithmic_Treatment#Preface

[14] "Pretending to Have (or to Be) a Computer as a Strategy in Teaching", Paul Berry, in *Harvard Educational Review*, 34, 1964, pp.383-401

[15] "A Dictionary of APL", Kenneth E. Iverson, in *ACM SIGAPL Quote-Quad*, 18, N°1, September 1987, ISSN 0163-6006, http://portal.acm.org/citation.cfm?xml:id=36983.36984

[16] Roger Hui, *op. cit.*, keiapl.org http://keiapl.org/rhui/remember.htm#parser

[17] "A personal view of APL", Kenneth E. Iverson, *IBM Systems Journal*, 30, N°4, 1991 http://www.research.ibm.com/journal/sj/304/ibmsj3004O.pdf

[18] "APL\?", Roger K.W. Hui, Kenneth E. Iverson, Eugene E. McDonnell & Arthur T. Whitney, in *Conference Proceedings on APL 90: for the future*, 1990, pp.192-200, ISBN 0-89791-371-X, http://portal.acm.org/citation.cfm?doxml:id=97808.97845

[19] *Incunabulum*, Roger Hui, 1992, http://www.jsoftware.com/jwiki/Essays/Incunabulum

[20] *An Implementation of J*, Roger Hui, 1992, http://www.jsoftware.com/jwiki/Doc/An_Implementation_of_J

[21] *Math for the Layman*, Kenneth E. Iverson, http://www.jsoftware.com/books/pdf/mftl.zip

[22] *Mathematics for the Million*, Lancelot Hogben, W.W. Norton, 1993

[23] Reiter, *op. cit.*