# THE USE OF APL IN TEACHING

Dr. K. E. Iverson

I.B.M. Scientific Centre,
P.O. Box 218,
Yorktown Heights,
New York, 10598, U.S.A.

Abstract:

A new approach to teaching algebra using the formal notation of A.P.L. is discussed. Students are introduced to ideas by means of examples, and are encouraged to investigate by themselves.

<u>Rapporteurs</u>:

Mr. T. Anderson
Mr. J. C. Knight

Dr. Iverson began by saying that he intended to illustrate three main points.

1. An approach to introducing notation.
2. Some characteristics of the APL language.
3. A style of use of computers in teaching.

An approach to introducing notation

The use of computers in teaching appears to fall into two distinct categories.

1. CAI which is generally programmed instruction, e.g. Coursewriter, and may be regarded as drill.
2. Unrestricted use, where a student considers the computer as a tool to help solve his problems.

Teaching courses using computers generally degenerate to teaching programming or problem solving, losing sight of their original curriculum because of the difficulty of programming languages. It is interesting to note that there are no courses in mathematical notation, this is merely introduced as needed in conjunction with the concept being discussed.

This seems to be the ideal way of teaching with computers. We introduce some topic, and a well-designed language would not be an obstruction.

A careful use is to be made of a formal notation which can be run on the computer, an advantage which ordinary mathematical notation does not have.

Characteristics of the APL language

There are three initial concepts to establish. Thereafter, the language can be introduced as needed. These concepts are straightforward and they are:

1. Writing an expression.

```
      (3+4)×(5+6)
77
```

2. Assigning a name to the result of an expression.

```
      Y←3+4
```

$Y$ may now be used in any expression.

```
      Y×Y
49
```

3. Assigning a name to an expression

```
        ∇Z←F X
[1]     Z←(X-3)×(X-5)∇
```

*F* may now be used to evaluate the above expression for any value of *X*

```
        F 7
8
```

<u>A style of use of computers in teaching</u>

The behaviour of *F* can be explored for a set of values 1,2,3, etc.:

```
        F 1 2 3 4 5 6 7
8  3  0  ¯1  0  3  8
```

This shows the mapping concept of a function and introduces the student to the concept of vectors, which may also be given names.

```
        S←1 2 3 4 5 6 7
        F S
8  3  0  ¯1  0  3  8
```

Tables are a useful aid to the exploration of elementary operators. Thus we have

```
        M←S∘.-S
        M
0  ¯1  ¯2  ¯3  ¯4  ¯5  ¯6
1   0  ¯1  ¯2  ¯3  ¯4  ¯5
2   1   0  ¯1  ¯2  ¯3  ¯4
3   2   1   0  ¯1  ¯2  ¯3
4   3   2   1   0  ¯1  ¯2
5   4   3   2   1   0  ¯1
6   5   4   3   2   1   0
```

a subtraction table which has been given a name. Such a table may be used to demonstrate the concept of commutativity by comparing its transpose with itself.

```
        M=⍉M

1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

The table shows that the operation of subtraction does not commute.

N.B. The use of the integers 0 and 1 for logical results enables their use in further arithmetic operations.

APL enables commutativity to be investigated easily for many other less familiar operators, e.g. ⌈ ⌊

```
⎕←Q←S∘.⌈S

1  2  3  4  5  6  7
2  2  3  4  5  6  7
3  3  3  4  5  6  7
4  4  4  4  5  6  7
5  5  5  5  5  6  7
6  6  6  6  6  6  7
7  7  7  7  7  7  7

Q=⍉Q

1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
```

Similarly set closure may be demonstrated (not proven) for various operators, using the set membership operator.

```
Q∊S

1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
```

It is easily seen that multiplication is not closed for finite sets.

```
(S∘.×S)∊S
1 1 1 1 1 1 1
1 1 1 0 0 0 0
1 1 0 0 0 0 0
1 0 0 0 0 0 0
1 0 0 0 0 0 0
1 0 0 0 0 0 0
1 0 0 0 0 0 0
```

The above has dealt with expression evaluation. Equations and their roots can also be tackled with the APL notation and approach.

Using the function $F$ previously defined and the vector $S$ it is possible to look for the zeros of $F$ ,

```
0=F S
0  0  1  0  1  0  0
```

or indeed, more general values of $F$ may be explored.

```
          R←8 7 6 5 4 3 2 1 0 ¯1
          R∘.=F S

 1 0 0 0 0 0 1
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 0 1 0 0 0 1 0
 0 0 0 0 0 0 0
 0 0 0 0 0 0 0
 0 0 1 0 1 0 0
 0 0 0 1 0 0 0
```

This table gives a rough graph, in fact an incidence matrix. A more pleasing graph is obtained by using the character handling facilities of the notation.

```
          ' *'[1+R∘.=F S]
 *     *



  *   *



   * *
    *
```

Any useful technique such as this may be defined as a function.

```
          ∇Z←GR M
 [1]      Z←' *'[1+M]∇
```

The study of algebraic relations tends to be dull in conventional texts but by using arrays students can see relations as patterns. This is an important and useful teaching aid.

```
          S∘.≥S

 1 0 0 0 0 0 0
 1 1 0 0 0 0 0
 1 1 1 0 0 0 0
 1 1 1 1 0 0 0
 1 1 1 1 1 0 0
 1 1 1 1 1 1 0
 1 1 1 1 1 1 1
```

The execution of a function which uses iteration to compute its result can be followed by means of the APL trace facility.

```
        ∇Z←X G Y
[1]     Z←X
[2]     X←X|Y
[3]     Y←Z
[4]     →X≠0∇

        TΔG←2 3

        48 G 78
G[2]    30
G[3]    48
G[2]    18
G[3]    30
G[2]    12
G[3]    18
G[2]    6
G[3]    12
G[2]    0
G[3]    6
6
```

N.B. Branching is introduced with the minimum of new notation and uses the arithmetic capabilities already specified. Evaluation of an APL expression proceeds from right to left. This is equivalent to the structuring of sentences in English and results in the expressions being easy to read from left to right — compare mathematicians f(g(h(x))). Also we gain the useful features $-/X$, which is the alternating sum, and $\div/X$, which is the alternating product.

<u>Professor Page</u> raised the question of execution times and efficiency of programs and asked Dr. Iverson to comment.

<u>Dr. Iverson</u> suggested that efficiency at the level of individual instructions has been overstressed, and that overall efficiency is of greater importance. APL facilitates obtaining overall efficiency by its communicability and compactness.

<u>Professor Page</u> recalled Professor Knuth's comment that in some programs just a small percentage of the code is executed for a very large amount of the time and hence needs to be efficient.

Dr. Iverson agreed and said that once the algorithm and system have been developed the parts which have to be efficient can be reprogrammed by someone with expert knowledge. APL enables these critical areas to be determined easily.

<u>Professor Wirth</u> pointed out that if one knows exactly how a machine works and represents its data then frequently one completely re-designs an algorithm to take account of this.

<u>Dr. Iverson</u> agreed but said that in general this level of detail is not necessary. When it is, then of course it must be considered.

Professor Perlis commented that neat looking one line APL programs using the power of the notation may not use the best method of computation. When the best method is used the APL becomes inelegant. He cited an example.

<u>Dr. Iverson</u> disagreed and claimed that the example could have been written efficiently and elegantly using different APL features.

An example of APL use is in finding the depth of nesting in a parenthesised expression. This may be done as follows:

```
      E←'((3+X)×(Y-Q))*((Y≥Q)-1)'
        12222112222100122221110

      '('=E
1  1  0  0  0  0  0  1  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0

      ')'=E
0  0  0  0  0  1  0  0  0  0  0  1  1  0  0  0  0  0  0  1  0  0  1

      '()'∘.=E
 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 1

      1 ¯1+.×'()'∘.=E
1  1  0  0  0 ¯1  0  1  0  0  0 ¯1 ¯1  0  1  1  0  0  0 ¯1  0  0 ¯1

      +\1 ¯1+.×'()'∘.=E
1  2  2  2  2  1  1  2  2  2  2  1  0  0  1  2  2  2  2  1  1  1  0
```

and this expression can be defined as a function.

<u>Professor Randell</u> asked whether the young students being taught by this system would display such ability in solving this nesting problem.

<u>Dr. Iverson</u> said that it was his experience that they did and have no difficulty with array manipulation when it is introduced this way.

Further examples were given covering Polish notation, Statistics, Geometry and the method of false position for solving equations.

Returning to the teaching aspect, when children are taught natural language this is done in the order — meaning, correct structure and finally equivalence, i.e. considerations of style. Unfortunately, this last topic is dealt with first when teaching algebra and programming languages. It would be best left until a familiarity with the features of the language has been gained and should be done at an elementary level such as observing patterns.

In line with this the concept of the power function may be introduced numerically, rather than by its formal definition.

```
      S←⍳7
      S
1  2  3  4  5  6  7
      2*S
2  4  8  16  32  64  128
```

and the student can see how this extends to the right

```
      2*S,8 9 10
2  4  8  16  32  64  128  256  512  1024
```

and by the same rule, to the left

```
      2*¯2 ¯1 0,S,8 9 10
0.25  0.5  1  2  4  8  16  32  64  128  256  512  1024
```

Using the previously described graphing capability it is possible to introduce the notion of linear equations and their solutions.

To construct methods for determining prime numbers their properties can be investigated. Clearly the property of being prime has something to do with divisibility, so a table of remainders is calculated.

```
      S←⍳12
      S∘.|S
 0  0  0  0  0  0  0  0  0  0  0  0
 1  0  1  0  1  0  1  0  1  0  1  0
 1  2  0  1  2  0  1  2  0  1  2  0
 1  2  3  0  1  2  3  0  1  2  3  0
 1  2  3  4  0  1  2  3  4  0  1  2
 1  2  3  4  5  0  1  2  3  4  5  0
 1  2  3  4  5  6  0  1  2  3  4  5
 1  2  3  4  5  6  7  0  1  2  3  4
 1  2  3  4  5  6  7  8  0  1  2  3
 1  2  3  4  5  6  7  8  9  0  1  2
 1  2  3  4  5  6  7  8  9 10  0  1
 1  2  3  4  5  6  7  8  9 10 11  0
```

The real concern is with divisibility, i.e. a remainder of zero.

```
        0=S∘.|S

  1 1 1 1 1 1 1 1 1 1 1 1
  0 1 0 1 0 1 0 1 0 1 0 1
  0 0 1 0 0 1 0 0 1 0 0 1
  0 0 0 1 0 0 0 1 0 0 0 1
  0 0 0 0 1 0 0 0 0 1 0 0
  0 0 0 0 0 1 0 0 0 0 0 1
  0 0 0 0 0 0 1 0 0 0 0 0
  0 0 0 0 0 0 0 1 0 0 0 0
  0 0 0 0 0 0 0 0 1 0 0 0
  0 0 0 0 0 0 0 0 0 1 0 0
  0 0 0 0 0 0 0 0 0 0 1 0
  0 0 0 0 0 0 0 0 0 0 0 1
```

The columns of this table indicate which numbers divide the column number. A prime is recognised as having exactly two distinct divisors and this is determined by equating the column sums to two.

```
        2=+/[1]0=S∘.|S
0  1  1  0  1  0  1  0  0  0  1  0
```

Compressing this over S gives the primes in S

```
        (2=+/[1]0=S∘.|S)/S
2  3  5  7  11
```

Another way of looking at primes is that they have something to do with multiplication. Therefore, construct a multiplication table

```
        S←1+⍳11
        S∘.×S

  4    6    8   10   12   14   16   18   20   22   24
  6    9   12   15   18   21   24   27   30   33   36
  8   12   16   20   24   28   32   36   40   44   48
 10   15   20   25   30   35   40   45   50   55   60
 12   18   24   30   36   42   48   54   60   66   72
 14   21   28   35   42   49   56   63   70   77   84
 16   24   32   40   48   56   64   72   80   88   96
 18   27   36   45   54   63   72   81   90   99  108
 20   30   40   50   60   70   80   90  100  110  120
 22   33   44   55   66   77   88   99  110  121  132
 24   36   48   60   72   84   96  108  120  132  144
```

The primes are those numbers which do not occur in this table

```
        (~S∊S∘.×S)/S
2  3  5  7  11
```

The sieve of Eratosthenes can be programmed very efficiently (both in space and execution time) as an iterative APL program.

Similarly, polynomials may be considered and their multiplication and differentiation studied. This can be extended quite easily to the consideration of infinite series — exponential, logarithmic, etc.

In conclusion, it can be said that children find this approach helpful since they can be given concrete examples of the notions they are being taught. Additionally, they can experiment and investigate for themselves.