Vector thinking in APL, R and Mathematica

A. I. McLeod

University of Western Ontario

Abstract

Keywords: APL, Mathematica, R.

APL was first developed by Kenneth E. Iverson (http://en.wikipedia.org/wiki/Kenneth_ E._Iverson), in a Ph.D. Thesis at Harvard and in subsequent graduate courses he taught there, as an executable mathematical notation (Iverson 1980) and this development was eventually recognized with a Turing Award (http://en.wikipedia.org/wiki/Turing_Award). The material from Iverson's thesis and lectures was later published in book form (Iverson 1962) and then after Iverson joined IBM it was implemented on IBM/360 computers and this implementation is the subject of the first widely available manual on APL (Pakin 1968). A survey of APL and its extensions to APL2 and J is given in (http://en.wikipedia.org/ wiki/APL_programming_language).

There are many similarities between APL and the S family of languages. At the most fundamental conceptual level these languages emphasize the elegance of vector thinking and the vectorization of computations (Chambers 2008, Ch. 6.4). For example, one method of generating 1000 NID(0, 1) random variables would be to generate a matrix of dimension 12×1000 of uniform (0, 1) random numbers and then sum the columns and subtract 6. In R we use apply but in proglangAPL the reduction operator / plays the same role. Similarly in Mathematica, in this situation, we would use Map or /@.

Specifically in R,

```
R> X <- apply(matrix(runif(12*1000), nrow=1000), MARGIN=1, sum) - 6
```

whereas in APL,

 $X \leftarrow \ ^-6 \ ++/ \ (10 \star \ ^-6) \ \times \ 1000 \ 12 \ \rho \ ? \ 10000 \ \rho 10 \ \star 6$

and in Mathematica,

```
X = Total /@ Partition[Array[Random[] &, 1000*12], 12] - 6;
```

Vectorization of computations combined with an interactive programming environment, pioneered in APL, has proved to be a very effective way of getting results fast. In fact, Wolfram (2003, p. 1408) lists nine similar vector-thinking type functions in APL and their equivalents in Mathematica. Roughly speaking, for numerical computation, all nine of these vectorization functions have their builtin approximate equivalents in S as well as shown in Table 2.

APL	Mathematica	S
catenate	Join	c, cbind, rbind
compress	Select	Index vectors [*]
grade	Ordering	order
grade	Sort	sort
iota operator	Range	1:n
ravel	Flatten	as.vector
reduce	Map	apply
reshape	Partition	dim
shape	Dimensions	dim

Some basic vectorization functions in APL, Mathematica and S.

* Index vectors, see ?, Sections 2.7

The R workspace management package **rwm** contains functions loadws(...) savews(...) and clearws(...) that are loosely inspired by the *APL* commands)LOAD,)SAVE and)CLEAR (Polvika and Pakin 1975, Ch. 8), (Anscombe 1981, p. 103), (Grenander 1982, pp. 151–152), (Stiers, Goovaerts, and de Kerf 1983, Ch. 9).

Today there are many exciting developments in the use of APL for software re-engineering (Askoolum 2006). A new .Net application of APL has been implemented that is useful for internet applications utilizing Microsoft Office products - see,

http://www.aplnext.com/visualapl/getready/default.aspx

An overview of APL with links to many other current implementations is available from *Wikipedia*,

http://en.wikipedia.org/wiki/APL_%28programming_language%29

References

- Anscombe FJ (1981). Computing in Statistical Science through APL. Springer-Verlag. ISBN 3540905499.
- Askoolum A (2006). System Building with APL + WIN. Wiley, New York. ISBN 978-0470030202.
- Chambers JM (2008). Software for Data Analysis: Programming with R. Statistics and Computing. Springer-Verlag.
- Grenander U (1982). Mathematical Experiments on the Computer, volume 105 of Pure and Applied Mathematics. Academic Press, New York. ISBN 0-12-301750-5.

Iverson KE (1962). A Programming Language. Wiley, New York. ISBN 0-471430-14-5.

- Iverson KE (1980). "Notation as a Tool of Thought." Communications of the ACM, 23(8). 1979 Turing Award Lecture.
- Pakin S (1968). APL/360 Reference Manual. Scientific Research Associates Inc.
- Peell HA (1987). "An APL idiom inventory." SIGAPL APL Quote Quad, 17(4), 362–368. ISSN 0163-6006. doi:http://doi.acm.org/10.1145/384282.28360.

A. I. McLeod

- Polvika RP, Pakin S (1975). APL: The Language and its Usage. Prentice-Hall, New Jersey. ISBN 0-13-038885-8.
- Stiers D, Goovaerts MJ, de Kerf J (1983). APL: The Language and its Actuarial Applications, volume 2 of Insurance Series. Elsevier, Amsterdam. ISBN 0 444 70155 9.
- Thomson N (1989). APL Programs for the Mathematics Classroom. Springer-Verlag. ISBN 0-387-97002-9.
- Wolfram S (2003). *The Mathematica Book*. Wolfram Media, 5th edition. URL http://documents.wolfram.com/mathematica/Mathematica_V5_Book.zip.

Affiliation:

A.I. McLeod University of Western Ontario E-mail: aimcleod@uwo.ca