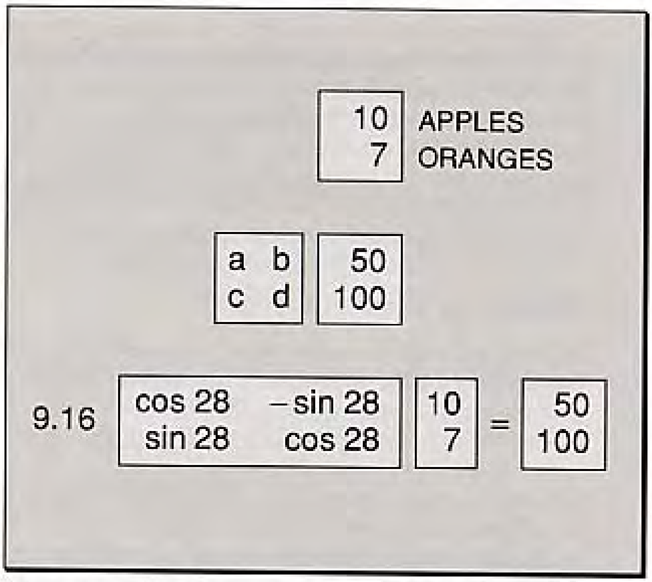
form . . . emerges . . . in a glorified shape—as an organism composed of discrete parts, but having an essential and undivisible unity as a whole of its own. . . . The conception of multiple quantity rises upon the field of vision. . . . [Matrix] dropped its provisional mantle, its aspect as a mere schema, and stood revealed as bona fide multiple quantity subject to all the affections and lending itself to all the operations of ordinary numerical quantity."

"This revolution," he continued, "was effected by a forcible injection into the subject of the concept of addition; that is, by choosing to regard matrices as susceptible to being added to one another; a notion as it seems to me, quite foreign to the idea of substitution, the nidus in which that of multiple quantity was laid, hatched and reared. This step was, as far as I know, first made by Cayley . . . in his [immortal] Memoir on Matrices [1858], wherein he may be said to have laid the foundation-stone of the science of multiple quantity. That memoir indeed (it seems to me) may in truth be affirmed to have ushered in the reign of Algebra the 2nd; just as Algebra the 1st . . . took its rise in Harriot's Artis Analyticae Praxis, published in 1631, . . . exactly 250 years before I gave the first course of lectures ever delivered on Multinomial Quantity, in 1881, at the Johns Hopkins University."[18] References 112 to 115 add some additional information about Cayley.

If Sylvester were here today, what pleasure would he find in Iverson's notation, implemented even on our personal computers as an interactive language—this notation that encourages, and as it were expects, us to think in terms of arrays or multiple quantities, manipulating them as entities in the spirit of Sylvester's exhortations! That eloquent mathematician would be even more moved, I am sure, by boxed arrays (arrays of arrays), and array processors, which are APL machines.

A century ago both Sylvester and Gibbs urged us to think in terms of arrays. Most computer languages and what Backus called (perhaps unfairly) the Von Neumann bottleneck, force us, however, to work with scalars. Within the confines of a few pages, I have attempted to trace the development of notation and methods from hieroglyphics to APL. I have tried to show that APL is much more than yet another computer language; that its intellectual importance is great; and that (yet again using Sylvester's words) APL continues "The wondrous tale of Multiple Quantity."

Figure 12 Separation of versor and tensor



The story will, of course, never be completed. We have seen the recent introduction of two hitherto undefined phrases now called *hooks* and *forks*.[116] One example of each must suffice here.

`+/ % # y` computes the sum over the reciprocals of the tally of `y`, which is unlikely to be useful, whereas, if we unify the phrase, placing it in parentheses, it becomes a fork `(+/ % #) y` equivalent to `(+/y) % (#y)`, which computes the mean (or means over the leading axis if the rank exceeds 1).

`(- mean) y` is a hook, equivalent to `y - (mean y)`, which gives the deviations from the mean, a necessary step in computing variance.

It should be noted that when we define the phrase, as for example `mean=. +/ % #` the phrase is unified without requiring parentheses. The functions used above for the Pauli identities are examples of forks. The statistical examples above include hook (sums of cross products) and fork (correlation coefficients). The function for interest on a declining balance (*ib*) includes a train of five functions, three of which (*i*, *r*, *b*) are forks, and it ends with an interesting hook.

In a paper published in 1866 we find Sylvester writing on the subject of operators. "The force of the bracket [i.e., parentheses] explains itself. This wonderful symbol has the faculty of extending itself