	name	pseudocode notation	math (Householder) notation	desirable output
1	inner product	vector ' * vector	$u^T u$	scalar
2	matvec	matrix * vector	Au	vector
3	outer product	vector * vector'	uv^T	matrix

Table 1: Desirable basic linear algebra outputs for a computer language with data structures {scalar,vector, matrix} and operators {prime,star} for transpose and multiply.

1 The Impossibility Theorem.

Most modern programming languages have separate data structures for scalars, vectors, and matrices. These data structures are concrete as opposed to the human (but not strictly mathematical) tendency to conflate scalars with 1×1 matrices and vectors with $n \times 1$ matrices (or perhaps row vectors and $1 \times n$ matrices.)

If we begin any exploration in a universe with exactly the set of data structures {scalar, vector, matrix} and we allow the operator symbols prime (') and star (*), it seems universally agreed that the desired output types ought to follow rules in Table 1.

We point out that the R language makes the choice that matrix times vector is a matrix and vector transpose is a matrix. In Python a vector transpose is a no-op meaning it returns its input so that the combination of a prime and star operator can not both be inner and outer product. One can always resort to long words rather than Householder notation, but that can feel less than ideal. Matlab is unique among computer languages in that it only has the data structure matrix from the set {scalar,vector,matrix} promoting the aforementioned confluence. This leads to other tradeoffs. Julia achieves the results in Table 1, but only by sidestepping the following scalar/vector/matrix impossibility theorem.

Theorem 1 *The {scalar,vector,matrix} trichotomy in isolation and the {prime,star} operations are inconsistent with Table 1.*

Proof: We begin by making the choice that matvec will result in a vector as defined by the second condition. If v' is a vector, we run into the python problem that v' * w is indistinguishable from v*w'. If v' is a matrix, then v'*w is a vector contradicting the first desirable result that v'*w is a scalar exhausting all reasonable possibilities. □