

[6.43] Dot product (scalar product) for complex vectors

The built-in `dotp()` function can be used to find the scalar dot product for two complex vectors, but its behavior may be puzzling. The vector dot product of \mathbf{a} and \mathbf{b} is defined as

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 \cos(\theta) \quad [1]$$

where θ is the angle between the two vectors, and

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^n a_i^2}$$

which is the L_2 (Euclidean) norm. Suppose that we want to find the dot product of a vector and itself, or

$$\mathbf{a} \cdot \mathbf{a} = \|\mathbf{a}\|_2 \|\mathbf{a}\|_2 \cos(\theta) = \|\mathbf{a}\|_2^2 \cos(0) = \|\mathbf{a}\|_2^2$$

In other words, the dot product of a vector with itself must be real and non-negative. `dotp()` correctly returns this result. Some users claim that the 89/92+ calculate this function in error, and that the complex conjugate of the second argument must be taken, to get the 'correct' result. However, suppose we apply this technique to the vector $[1+i, 1+i]$, where 'i' is the square root of -1. Then

$$\text{dotp}([1+i], [1+i]) = 4$$

as expected, but

$$\text{dotp}([1+i], \text{conj}([1+i])) = 4i$$

which must be wrong, because the result must be real and non-negative. Some of this confusion may result because there are actually two working definitions of dot product. In linear algebra, optimization and computer science, the inner product is defined as

$$\mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i \cdot y_i \quad [2]$$

While this definition provides consistent results in real linear algebra and optimization, it cannot be used for the complex vector dot product, because it fails in the case of finding the dot product of a vector with itself.

For real-valued vectors, there is no ambiguity, and `dotp()` returns the correct results. Note also that `dotp()` can accept lists as arguments, and for complex elements it also implicitly uses the definition in [1] above, not [2].

(Credit to Bhuvanesh Bhatt and Fabrizio)