Not everything is determined by a basis

We know that the entire table of values for a linear transformation is determined from the values on a basis of the domain. However, it is wrong to assume that *everything* is determined by that tiny little bit of data.

This point came up in the lecture of Dec 6. Stated in its bottom-line version, it reads: *Suppose that for an $n \times m$ matrix B, we have for some orthonormal basis of $\mathbb{R}^m$ $\mathfrak{B} = \{\vec{u}_1, ..., \vec{u}_m\}$, the largest of the m non-negative numbers $\|B\vec{u}_i\|$ is C. Is it true that $\|B\vec{u}\| \leq C$ for all unit vectors?* The answer: yes, if the collection of vectors $\{B(\vec{u}_1), ..., B(\vec{u}_m)\}$ are orthogonal, but not in general.

All we have to do is compute (with general symbols). Write the general unit vector as

$$\vec{u} = \sum_{i=1}^{m} a_i \vec{u}_i, \qquad \text{so } \sum_{i=1}^{m} a_i^2 = 1.$$

Apply B, using linearity as far as it will go:

$$B(\vec{u}) = \sum_{i=1}^{m} a_i B(\vec{u}_i), \qquad \text{with } \sum_{i=1}^{m} a_i^2 = 1.$$

To compute $\|B\vec{u}\|^2 = B(\vec{u}) \cdot B(\vec{u})$, use the bilinearity of inner product to yield

$$\|B\vec{u}\|^2 = \sum_{i=1}^{m} a_i^2 B(\vec{u}_i) \cdot B(\vec{u}_i) + \text{ [cross-terms]},$$

and the cross-terms are all zero when $\{B(\vec{u}_1), ..., B(\vec{u}_m)\}$ form an orthogonal collection of vectors. When this orthogonality holds, we get:

$$\|B\vec{u}\|^2 = \sum_{i=1}^{m} a_i^2 B(\vec{u}_i) \cdot B(\vec{u}_i) = \sum_{i=1}^{m} a_i^2 \|B\vec{u}_i\|^2 \leq \sum_{i=1}^{m} a_i^2 C^2 = C^2.$$

We note that for non-orthogonal $\vec{v}_i = B(\vec{u}_i)$, there is nothing preventing all cross-terms from being positive. To write down a counterexample, let $m = 2$, and assume that $\vec{v}_1$ and $\vec{v}_2$ are unit vectors (so $C = 1$) but $\vec{v}_1 \cdot \vec{v}_2 > 0$. (Draw a picture.) Then for (say) $a_1 = a_2 = 2^{-\frac{1}{2}}$,

$$\|B\vec{u}\|^2 = a_1^2 + a_2^2 + 2a_1a_2(\vec{v}_1 \cdot \vec{v}_2),$$

which is *more than* $C = 1$ whenever a_1 and a_2 are both positive (or both negative).