3D Geometry for Computer Graphics - Exercise 2 Solution

1. Let V be a vector space with inner product. Prove that if $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\} \in V$ is a set of (pairwise) orthogonal vectors then $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ are linearly independent.

   Solution: Suppose that
   $$\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \ldots + \alpha_k \mathbf{v}_k = 0. \tag{1}$$
   Multiply both sides by $\mathbf{v}_i$ for some $i \in \{1, 2, \ldots, k\}$:
   $$\alpha_1 < \mathbf{v}_1, \mathbf{v}_i > + \alpha_2 < \mathbf{v}_2, \mathbf{v}_i > + \ldots + \alpha_k < \mathbf{v}_k, \mathbf{v}_i >= 0$$
   Since $< \mathbf{v}_j, \mathbf{v}_i >= 0$ for all $j \neq i$, we are left with
   $$\alpha_i < \mathbf{v}_i, \mathbf{v}_i >= 0$$
   $$\alpha_i \|\mathbf{v}_i\|^2 = 0$$
   For all i, $\mathbf{v}_i \neq 0$, so $\|\mathbf{v}_i\| \neq 0$, thus it follows that $\alpha_i = 0$. Therefore, assumption (1) leads to $\alpha_i = 0\ \forall i$, and then the vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k$ are linearly independent by definition.

2. Prove that if A is an orthogonal matrix then $\det(A) = \pm 1$.

   Solution: if A is orthogonal then $A^{-1} = A^T$. Then:
   $$1 = \det(I) = \det(AA^{-1}) = \det(A)\det(A^{-1}) = \det(A)\det(A^T) = \det(A)\det(A).$$
   We have use the facts that $\det(AB) = \det(A)\det(B)$ and that $\det(A) = \det(A^T)$. We get that $(\det(A))^2 = 1$, so $\det(A) = 1$ or $\det(A) = -1$.

3. True or not true? A linear operator $A : \mathbb{R}^n \to \mathbb{R}^n$ is 1-1 $\Leftrightarrow$ a matrix representing A with respect to some basis of $\mathbb{R}^n$ is non-singular.

   Solution: True.

   [$\Rightarrow$] A is 1-1 means: if $A\mathbf{v} = A\mathbf{u}$ then $\mathbf{v} = \mathbf{u}$. Suppose A is singular. This means the column-vectors of A are linearly dependent. Denote the columns by $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$. There exist $\alpha_1, \alpha_2, \ldots, \alpha_n$, not all zero, such that
   $$\alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \ldots + \alpha_k \mathbf{a}_k = 0.$$
   We know that not all $\mathbf{a}_i$ are zero (otherwise A were the zero matrix, which is not 1-1). So, the vector $\mathbf{v} = \alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \ldots + \alpha_k \mathbf{a}_k$ is not zero. Then we have $A\mathbf{v} = 0$ and also $A0 = 0$, which contradicts the fact that A is 1-1 transformation.

   [$\Leftarrow$] Assume A is non-singular. Then for any vector $\mathbf{b}$ there is only one $\mathbf{x}$ such that $A\mathbf{x} = \mathbf{b}$. In particular, $A\mathbf{x} = 0 \to \mathbf{x} = 0$. So, if $A\mathbf{v} = A\mathbf{u}$ then $A\mathbf{v} - A\mathbf{u} = 0 \to A(\mathbf{v} - \mathbf{u} = 0 \to \mathbf{v} - \mathbf{u} = 0 \to \mathbf{v} = \mathbf{u}$. Thus, A is 1-1.

4. Let A be a square matrix. Prove that if $\lambda_1, \lambda_2, \ldots, \lambda_k$ are *distinct* eigenvalues of A then corresponding eigenvectors are linearly independent.

Solution: We'll prove by induction. One vector is always independent (assuming it's not zero, and eigenvectors are non-zero by definition). Suppose we proved that any group of $k-1$ eigenvectors that correspond to distinct eigenvalues, is linearly independent. Let's prove for k vectors. Assume

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_k\mathbf{v}_k = 0. \quad (*)$$

Apply A to both sides:

$$A(\alpha_1\mathbf{v}_1) + A(\alpha_2\mathbf{v}_2) + \ldots + A(\alpha_k\mathbf{v}_k) = 0. \quad (**)$$
$$\lambda_1\alpha_1\mathbf{v}_1 + \lambda_2\alpha_2\mathbf{v}_2 + \ldots + \lambda_k\alpha_k\mathbf{v}_k = 0.$$

Multiply $(*)$ by λ_i:

$$\lambda_i\alpha_1\mathbf{v}_1 + \lambda_i\alpha_2\mathbf{v}_2 + \ldots + \lambda_i\alpha_k\mathbf{v}_k = 0. \quad (***)$$

Subtract $(***)$ from $(**)$:

$$(\lambda_1 - \lambda_i)\alpha_1\mathbf{v}_1 + \ldots + (\lambda_{i-1} - \lambda_i)\alpha_{i-1}\mathbf{v}_{i-1} + (\lambda_{i+1} - \lambda_i)\alpha_{i+1}\mathbf{v}_{i+1} + \ldots + (\lambda_k - \lambda_i)\alpha_k\mathbf{v}_k = 0.$$

We arrive at a linear combination of $k-1$ eigenvectors that gives zero. By induction, we know the vectors are independent, so all the coefficients $(\lambda_j - \lambda_i)\alpha_j$ must be zero. But $\lambda_j \neq \lambda_i$, so $\alpha_j = 0$ for all $j \neq i$. Substitution of $\alpha_j = 0$ in $(*)$ gives us $\alpha_i\mathbf{v}_i = 0$, and then follows that α_i must be zero as well.

5. Prove that if A is a symmetric matrix and λ, μ are its eigenvalues ($\lambda \neq \mu$) then corresponding eigenvectors are orthogonal (i.e. if $A\mathbf{v} = \lambda\mathbf{v}$ and $A\mathbf{w} = \mu\mathbf{w}$ then $<\mathbf{v}, \mathbf{w}> = 0$).

   Solution:

$$\mathbf{v}^T A\mathbf{w} = \mathbf{v}^T\mu\mathbf{w} = \mu\mathbf{v}^T\mathbf{w}$$
$$\mathbf{w}^T A\mathbf{v} = \mathbf{w}^T\lambda\mathbf{v} = \lambda\mathbf{w}^T\mathbf{v}$$

   Since $A = A^T$, we have:

$$(\mathbf{w}^T A\mathbf{v})^T = \mathbf{v}^T A^T\mathbf{w} = \mathbf{v}^T A\mathbf{w}.$$

   Thus,

$$\lambda\mathbf{v}^T\mathbf{w} = \mu\mathbf{v}^T\mathbf{w}$$
$$(\lambda - \mu)\mathbf{v}^T\mathbf{w} = 0.$$

   Since $\lambda \neq \mu$, we arrive at

$$\mathbf{v}^T\mathbf{w} = 0 \;\; \Rightarrow \; <\mathbf{v}, \mathbf{w}> = 0.$$

6. **Computer-science question**: Suppose you are implementing a matrix library. You want to be prepared to handle operations on big matrices. As mentioned in class, if A is a matrix and $\mathbf{b}$ is a column-vector, then the entries of $A\mathbf{b}$ are scalar products of $\mathbf{b}$ with the rows of A. Alternatively, we can look at $A\mathbf{b}$ as a linear combination of the columns of A, where the coefficients of the linear combination are the entries of $\mathbf{b}$. When is the first interpretation more computationally-advantageous, and when is the second? Hint: matrices are usually stored in the computer as arrays of numbers. They can be stored by rows or by columns. Think what happens when the matrix A is very big, in terms of access to its elements during the computation of $A\mathbf{b}$.

   Answer: If A is stored by columns, it's better to use the second method (column-oriented multiplication). If A is stored by rows, it's better to use the first method (rows-oriented). This is because this way we will access the elements of A sequentially, in the same order as they are stored in memory (i.e. we will scan all the columns one-by-one, or, respectively, the rows). Consecutive access to memory is better than random (scattered) – it usually reduces paging and allows more efficient caching.