Determinants by Geometric Memorization

Abstract
This work presents a constructive and didactic method for computing determinants based on an ordered sequence of pivot selections and matrix reductions. The determinant is obtained as a structured product of pivots, emphasizing geometric intuition and memorization rather than formal expansion rules.

1. Introduction
Traditional determinant computation methods often obscure structural understanding behind symbolic manipulation. The present approach proposes a geometric and recursive construction, where each step contributes transparently to the final result.

2. Conceptual Foundation
The determinant is constructed through successive reductions of the original matrix. At each stage, a pivot is selected and used to generate a lower-order submatrix. The process continues until the minimal configuration is reached.

3. Symbolic Representation
Conceptually, the determinant can be represented as an ordered product of pivots, following the construction rule imposed by the method's logomark. This expression is mnemonic and explanatory rather than computational.

4. Recursive Structure
Each reduction step preserves equivalence with the original determinant. The geometric interpretation highlights how local choices influence the global result without invoking cofactor expansion.

5. Pivot Selection and the Zero Exception
A valid pivot must be nonzero at every stage to ensure the construction proceeds. The only exception occurs in the 1×1 case, where no choice exists: the single element acts as a mandatory pivot, completing the process.

6. Didactic and Computational Aspects
While the symbolic product shown in the logomark conveys the core idea, it is not suitable for large matrices. In practice, each reduced matrix is normalized by dividing its elements by the previous pivot. This redistributes pivot values into fractional factors, forming a new equivalent product that is numerically stable and computationally efficient.

Conclusion
The method reveals the determinant as a transparent constructive process. By separating conceptual representation from practical computation, it provides both pedagogical clarity and algorithmic viability.