Matrix Primer

Add two 3x3 matrices

Pre-conditions: two non-empty 3x3 matrices of integer/ real / complex type
Post-conditions: a new 3x3 matrix of the same type with the elements added

Pseudo-code:

1. Let the matrices A, B be the input matrices
2. Let the matrix holding the sum be called Sum.
3. For I in 1.. 3 loop
    i. For J in 1.. 3 loop
        1. Sum(I,J) := A(I,J) + B(I,J)
4. Return matrix Sum

Multiply two 3x3 matrices

Suppose that A and B are two matrices and that A is an m x n matrix (m rows and n columns) and that B is a p x q matrix. To be able to multiply A and B together, A must have the same number of columns as B has rows (i.e., n=p). The product will be a matrix with m rows and q columns. To find the entry in row r and column c of the new matrix we take the "dot product" of row r of matrix A and column c of matrix B (pair up the elements of row r with column c, multiply these pairs together individually, and then add their products).

Mathematically,

$$C(I,J) = \sum_{K=1}^{n} A(I,K)\, B(K,J)$$

Where

- I ranges from 1.. m
- J ranges from 1.. q
- K ranges from 1.. n = p

Pre-conditions: two non-empty 3x3 matrices of integer/ real / complex type
Post-conditions: a new 3x3 matrix of the same type with the product of the matrices

Pseudo-code:

1. Let the matrices A, B be the input matrices
2. Let the matrix holding the product be called Product.
3. Use a local variable sum to store the intermediate value of product.
4. For I in 1.. 3 loop

   i. For J in 1.. 3 loop
      1. Sum := 0;
      2. For K in 1 .. 3 loop
         a. Sum := Sum + A(I,K) * B(K,J);
      3. End K loop
      4. Product(I,J) := Sum;
   ii. End J loop
5. End I loop
6. Return matrix Product

Transpose a 3x3 matrix

Pre-conditions: A non-empty 3x3 matrix

Post-conditions: A new 3x3 matrix of the same type with the elements in rows and columns exchanged

Pseudo-code:

1. Let the input matrix be A
2. Let the matrix holding the transpose be called Transpose.
3. For I in 1 .. 3 loop
   i. For J in 1.. 3 loop
      1. Transpose(I,J) := A(J,I)
4. Return matrix Transpose

Inverse of a 3x3 matrix

The inverse of a 3×3 matrix is given by:

$$A^{-1} = \frac{\mathrm{adj}A}{\det A}$$

We use *cofactors* to determine the adjoint of a matrix.

The *cofactor* of an element in a matrix is the value obtained by evaluating the determinant formed by the elements not in that particular row or column.

We find the *adjoint matrix* by replacing each element in the matrix with its cofactor and applying a + or - sign as follows:

$$\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$$

and then finding the *transpose* of the resulting matrix

The *determinant* of an *n*-by-*n* matrix *A*, denoted det *A* or |*A*|, is a number whose value can be defined recursively as follows. If n=1, i.e., if *A* consists of a single element a_{11}, det *A* is equal to a_{11}; for $n > 1$, det *A* is computed by the recursive formula

$$\det A = \sum_{j=1}^{n} s_j a_{1j} \det A_j,$$

where s_j is +1 if *j* is odd and.1 if *j* is even, a_{1j} is the element in row 1 and column *j* , and A_j is the *n* . 1-by-*n* . 1 matrix obtained from matrix *A* by deleting its row 1 and column *j* .

For a 3x3 matrix, the formula can be determined as:

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$= a_{11} \det \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} - a_{12} \det \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} + a_{13} \det \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

$$= a_{11}a_{22}a_{33}+a_{12}a_{23}a_{31}+a_{21}a_{32}a_{13}-a_{31}a_{22}a_{13}-a_{21}a_{12}a_{33}-a_{32}a_{23}a_{11}.$$

Preconditions: A 3x3 invertible matrix
Postconditions: A new 3x3 matrix which is the inverse of the input matrix

Pseudocode:

1. Let the input matrix be A
2. Let the cofactor matrix be Cofactor
3. Let Determinant be the variable used to store the determinant
4. For I in 1.. 3 loop
    a. Compute the indices of the elements to compute the determinant using the formula:
        i. I1 := (I + 1) Rem 3. If I1 = 0 then I1 = 3
        ii. I2 := (I + 2) Rem 3. If I2 = 0 then I2 = 3
    b. For J in 1..3 loop
        i. Compute the indices of the elements to compute the determinant using the formula:
            1. J1 := (J + 1) Rem 3. If J1 = 0 then J1 = 3
            2. J2 := (J + 2) Rem 3. If J2 = 0 then J2 = 3

        ii. Cofactor(I,J):=A(I1,J1)*A(I2,J2)- A(I1,J2) * A(I2,J1)

5. Compute determinant as

   Determinant := A(1,1)*Cofactor(1,1) + A(1,2)*Cofactor (1,2)+ A(1,3)*Cofactor (1,3)

6. Compute the transpose of the Cofactor matrix
7. For I in 1.. 3
   a. For J in 1..3
      i. Inverse(I,J) := Cofactor(I,J) / Determinant
8. Return Inverse

Note: The method for computing the cofactor automatically generates the required signs in the cofactor matrix.