

Documentation for Matrix Operations (LDU Decomposition)

Overview

This JavaScript code implements the LDU (Lower, Diagonal, Upper) decomposition of a square matrix. It allows users to input a matrix of any order, performs matrix operations (like identity matrix creation, elimination, and multiplication), and displays intermediate results such as the matrices **L**, **D**, and **U**. The main functions include matrix input, identity matrix generation, matrix multiplication, and decomposition steps.

Features

1. **Matrix Input:** Users can input a matrix of any size, which is stored in a 2D array.
 2. **Identity Matrix:** Creates an identity matrix for a given order.
 3. **Matrix Multiplication:** Multiplies two matrices and returns the result.
 4. **L, D, U Matrices:** Supports basic operations to find and display the L, D, and U matrices, which are part of the LDU decomposition.
-

Functions

`order1()`

- **Description:** Retrieves the matrix order (size) from the user input.
- **Parameters:** None.
- **Returns:** None. Sets the global `order` variable.

`show(id, showmat)`

- **Description:** Displays the matrix in a readable string format.
- **Parameters:**
 - `id`: The HTML element ID where the matrix will be displayed.
 - `showmat`: The matrix (2D array) to be displayed.
- **Returns:** A string representing the matrix in a formatted manner.

`matrix1(id)`

- **Description:** Prompts the user to input a matrix of the given size and stores it in the `a` array.
- **Parameters:** `id` – The HTML element where the matrix will be shown.
- **Returns:** None. Displays the matrix on the webpage.

`identity()`

- **Description:** Generates and returns an identity matrix of the given order.
- **Parameters:** None.
- **Returns:** A 2D identity matrix.

`multiply(mat1, mat2)`

- **Description:** Multiplies two matrices `mat1` and `mat2` and returns the resulting matrix.
- **Parameters:**
 - `mat1`: The first matrix to be multiplied.
 - `mat2`: The second matrix to be multiplied.
- **Returns:** The product matrix as a 2D array.

`elimination(num, r, c)`

- **Description:** Creates an elementary matrix by modifying a specific element.
- **Parameters:**
 - `num`: The number to set in the matrix.
 - `r`: The row index to modify.
 - `c`: The column index to modify.
- **Returns:** The modified elementary matrix.

`umatrix(id)`

- **Description:** Initializes the `U` matrix (upper triangular matrix) and prepares for LDU decomposition.
- **Parameters:** `id` – The HTML element where the `U` matrix will be displayed.
- **Returns:** None. The matrix `U` is initialized.

`Lmatrix(id)`

- **Description:** Initializes the `L` matrix (lower triangular matrix) and prepares for LDU decomposition.
- **Parameters:** `id` – The HTML element where the `L` matrix will be displayed.
- **Returns:** None. The matrix `L` is initialized.

`Dmatrix(id)`

- **Description:** Initializes the `D` matrix (diagonal matrix) and displays it.
- **Parameters:** `id` – The HTML element where the `D` matrix will be displayed.
- **Returns:** None. The matrix `D` is displayed after modifying non-diagonal elements to 0.

U_deshmatrix(id)

- **Description:** Initializes and prepares the `U_desh` matrix for LDU decomposition.
- **Parameters:** `id` – The HTML element where the `U_desh` matrix will be displayed.
- **Returns:** None. The `U_desh` matrix is initialized.

LDUmatrix(id)

- **Description:** Displays the product of the `L`, `D`, and `U_desh` matrices to demonstrate the LDU decomposition.
 - **Parameters:** `id` – The HTML element where the results will be displayed.
 - **Returns:** None. The product of the matrices is displayed on the webpage.
-

Workflow

1. **User Input:**
 - The user is prompted to enter the order of the matrix and the matrix elements.
 - The `matrix1(id)` function handles matrix input.
 2. **Matrix Operations:**
 - The identity matrix is created using the `identity()` function.
 - The matrix can be manipulated using elementary matrices (`elimination()`).
 3. **LDU Decomposition:**
 - The `U` matrix is initialized and ready for upper triangular decomposition.
 - The `L` matrix is created as a lower triangular matrix.
 - The `D` matrix is created by retaining only the diagonal elements from `U`.
 - The `U_desh` matrix is used to complete the decomposition, and the result is displayed.
 4. **Matrix Multiplication:**
 - The matrices `L`, `D`, and `U_desh` are multiplied to verify the correctness of the decomposition using the `multiply()` function.
-

Notes

- **Order of Matrix:** The matrix order must be provided by the user, and the code adjusts to accommodate any size of square matrices.
- **User Interface:** The matrix input and results are shown on the webpage through dynamically updated HTML elements.
- **Limitations:** The code does not yet fully implement LU decomposition logic; placeholders for `U` and `L` matrix creation are present but need further development.

