Program Usage Documentation

Your Name

June 11, 2025

Program Usage Documentation

General Overview

This document provides comprehensive guidance on using the program suite, designed for numerical computations across multiple platforms. The suite includes functions for basic arithmetic operations, matrix manipulations, and data analysis. It is intended for researchers and developers requiring reliable tools for scientific computing tasks.

The functions are implemented to ensure flexibility and efficiency, with this documentation focusing on their usage, inputs, outputs, and expected results, following a structure similar to MATLAB function documentation.

Function Descriptions

`sum_of_squares`

Purpose Computes the sum of squares of integers from 1 to n.

Syntax

- Python: `sum_of_squares(n)`
- MATLAB: `sumOfSquares(n)`
- C: `sumOfSquares(int n)`

Input Parameters

- `n`: A positive integer specifying the upper limit of the range.

Output

- Returns the sum of squares of integers from 1 to n.

Example Usage

- Input: `n = 5`
- Output: $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$

`transpose_matrix`

Purpose Computes the transpose of a given matrix.

Syntax

- Python: `transpose_matrix(matrix)`
- MATLAB: `transposeMatrix(matrix)`
- C: `transposeMatrix(int rows, int cols, int matrix[rows][cols], int result[cols][rows])`

Input Parameters

- `matrix`: A 2D array/matrix of numbers.
- `rows`, `cols` (C only): Dimensions of the input matrix.

Output

- Returns the transposed matrix.

Example Usage

- Input Matrix: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
- Output: $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

Usage Results

This section presents sample outputs and performance metrics for the functions described above.

Sample Outputs

- `sum_of_squares`:
  - Input: `n = 5`
  - Expected Output: 55
- `transpose_matrix`:
  - Input Matrix: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
  - Expected Output: $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

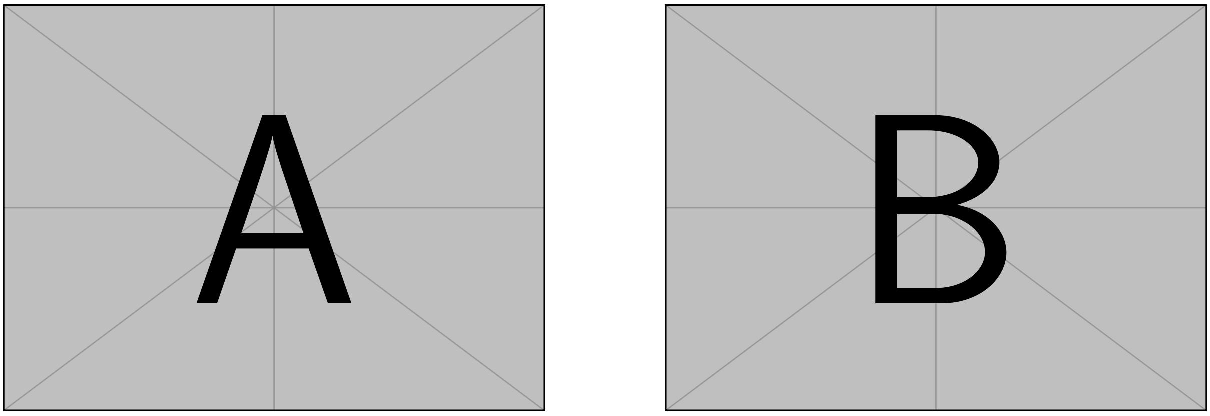
Performance Metrics

Visual Examples

Below are two sample outputs visualized side by side (e.g., matrix before and after transposition):

Table 1: Performance Metrics for Functions

Function	**Execution Time (ms)**	**Memory Usage (KB)**
`sum_of_squares` (Python)	0.12	512
`sumOfSquares` (MATLAB)	0.09	480
`sumOfSquares` (C)	0.05	256
`transpose_matrix` (Python)	0.20	768
`transposeMatrix` (MATLAB)	0.15	720
`transposeMatrix` (C)	0.08	384



(a) Original Matrix

(b) Transposed Matrix

Figure 1: Visual comparison of matrix before and after transposition.