JSC-61777-models/utils/math

# JSC Engineering Orbital Dynamics Math Model

**Simulation and Graphics Branch (ER7)**
**Software, Robotics, and Simulation Division**
**Engineering Directorate**

## Package Release JEOD v5.0

## Document Revision 1.0
## July 2022

**National Aeronautics and Space Administration**
**Lyndon B. Johnson Space Center**
**Houston, Texas**

JSC-61777-models/utils/math

# JSC Engineering Orbital Dynamics Math Model

**Document Revision 1.0**
**July 2022**

**Blair Thompson**

**Simulation and Graphics Branch (ER7)**
**Software, Robotics, and Simulation Division**
**Engineering Directorate**

**National Aeronautics and Space Administration**
**Lyndon B. Johnson Space Center**
**Houston, Texas**

**Abstract**

The JEOD Math Model performs commonly used mathematical operations on 3-by-3 (3x3) matrices and 3-by-1 (3x1) vectors. The model consists of functions that are called by other JEOD functions. The functions are *not* intended to be called at the Trick S_define level. Therefore, the Math Model is primarily of interest to simulation developers (programmers) more than simulation users (analysts) and integrators.

This document describes the implementation of the Math Model including the model requirements, specifications, mathematical theory, and architecture. A user guide is also provided to assist with implementing the model in Trick simulations.

# Contents

# Chapter 1

# Introduction

## 1.1 Model Description

This documentation describes the design and testing of matrix and vector math routines in the JSC Engineering Orbital Dynamics (JEOD) Math Model. These routines are derived from well-known rules and operators of linear algebra.

Included in this documentation are validation cases that describe tests done on the algorithms to verify that they are working correctly and computing the correct values for given input data.

There is also a User Guide which describes how to incorporate the above mentioned routines as part of a Trick simulation.

The parent document to this model document is the JEOD Top Level Document [1].

## 1.2 Document History

| Author | Date | Revision | Description |
|---|---|---|---|
| Blair Thompson | Nov 2009 | 1.0 | Initial document |

## 1.3 Document Organization

This document is formatted in accordance with the NASA Software Engineering Requirements Standard [3] and is organized into the following chapters:

**Chapter 1: Introduction** - This introduction contains three sections: description of model, document history, and organization. The first section provides the introduction to the Math Model and its reason for existence. It contains a brief description of the interconnections with other models, and references to any supporting documents. It also lists the document that is parent to this one. The second section displays the history of this document which includes author, date, and reason for each revision. The final section contains a description of the how the document is organized.

**Chapter 2: Product Requirements** - Describes requirements for the Math Model.

**Chapter 3: Product Specification** - Describes the underlying theory, architecture, and design of the Math Model in detail. It is organized in three sections: Conceptual Design, Mathematical Formulations, and Detailed Design.

**Chapter 4: User Guide** - Describes how to use the Math Model in a Trick simulation. It is broken into three sections to represent the JEOD defined user types: Analysts or users of simulations (Analysis), Integrators or developers of simulations (Integration), and Model Extenders (Extension).

**Chapter 5: Verification and Validation** - Contains Math Model verification and validation procedures and results.

# Chapter 2

# Product Requirements

The Math Model shall meet the JEOD project requirements specified in the JEOD Top Level Document [1].

*Requirement math_1: Matrix Math Operations*

**Requirement:**

The Math Model shall be capable of performing common matrix operations on 3x3 matrices. The operations shall include:

*1.1* Matrix Nullification

*1.2* Skew Symmetric Cross Product

*1.3* Matrix Outer Product

*1.4* Matrix Multiplication

*1.5* Matrix Addition

*1.6* Matrix Subtraction

*1.7* Matrix Transposition

*1.8* Matrix Inversion

*1.9* Various combinations of the operations listed above that are commonly used in aerospace simulations.

**Rationale:**

3x3 matrices are very common in aerospace simulations. There are also many common required matrix math operations. For ease of programming and speed of computation, a pre-established "library" of matrix math functions is necessary.

**Verification:**

Test

*Requirement math_2: Vector Math Operations*

**Requirement:**

The Math Model shall be capable of performing common vector operations on 3x1 (or 1x3) vectors. The operations shall include:

*2.1* Vector Normalization

*2.2* Vector Cross Product

*2.3* Vector Inner Product (Dot Product)

*2.4* Vector Magnitude

*2.5* Vector Addition

*2.6* Vector Subtraction

*2.7* Various combinations of the operations listed above that are commonly used in aerospace simulations.

**Rationale:**

3x1 vectors are very common in aerospace simulations. There are also many common required vector math operations. For ease of programming and speed of computation, a pre-established "library" of vector math functions is necessary.

**Verification:**

Test

# Chapter 3

# Product Specification

## 3.1 Conceptual Design

The Math Model is a collection of functions that peform common matrix and vector operations for aerospace simulations. The functions are based on well-known rules and operations of linear algebra.

## 3.2 Mathematical Formulations

The mathematical operations (or functions) used in the Math Model are based on common matrix and vector operations. The rules for these operations can be found in a variety of math texts (e.g., Kreyszig [2]).

## 3.3 Detailed Design

The Math Model functions are designed to be called by other JEOD functions. They are not intended to be called from the S_define level.

# Chapter 4

# User Guide

Three types of JEOD users are described in the model documents. However, the only type of user who can manipulate the Math Model are model Extenders. Therefore, Analysts and Integrators are not addressed in this particular user guide.

## 4.1 Analysis

(The Math Model functions are not intended for use at the simulation analyst/user level. Therefore, this section is not applicable.)

## 4.2 Integration

(The Math Model functions are not intended for use at the simulation integrator level. Therefore, this section is not applicable.)

## 4.3 Extension

### 4.3.1 Matrix Functions

In order to use the Math Model matrix functions, the following line must be included in the file of the calling function:

```
#include "utils/math/include/matrix3x3.hh"
```

An example call to the matrix function *copy* is:

```
Matrix3x3::copy(subject_body->composite_properties.inertia, inertia);
```

The matrix functions available in the Math Model are listed below. The equivalent math expression is shown along with a brief explanation, if necessary. Overloading has been used to repeat some function names. Note that all matrices are assumed to be 3x3.

**initialize,** $A \leftarrow null$ **(all elements of 3x3** $null$ **matrix are zero)**

**identity,** $A \leftarrow I$ **(**$I$ **is the 3x3 identity matrix)**

**cross_matrix,** $A \leftarrow \tilde{v}$ (returns skew symmetric cross product matrix from input vector $\bar{f}$)

$$A = \begin{bmatrix} 0 & -f_3 & f_2 \\ f_3 & 0 & -f_1 \\ -f_2 & f_1 & 0 \end{bmatrix}$$

**outer_product,** $A \leftarrow \bar{f} \otimes \bar{g}$ (returns outer product of two input vectors $\bar{f}$ and$\bar{g}$)

$$A = \begin{bmatrix} f_1g_1 & f_1g_2 & f_1g_3 \\ f_2g_1 & f_2g_2 & f_2g_3 \\ f_3g_1 & f_3g_2 & f_3g_3 \end{bmatrix}$$

**negate (in place),** $A \leftarrow (-A)$

**negate,** $A \leftarrow (-B)$

**transpose (in place),** $A \leftarrow A^T$

**transpose,** $A \leftarrow B^T$

**scale (in place),** $A \leftarrow sA$ **(**$s$ **is a scalar)**

**scale,** $A \leftarrow sB$ **(**$s$ **is a scalar)**

**incr (in place),** $A \leftarrow A + C$ **(**$C$ **is a 3x3 matrix of constants)**

**decr (in place),** $A \leftarrow A - C$ **(**$C$ **is a 3x3 matrix of constants)**

**copy,** $A \leftarrow B$

**add,** $A \leftarrow B + C$

**subtract,** $A \leftarrow B - C$

**product,** $A \leftarrow BC$

**product_left_transpose,** $A \leftarrow B^T C$

**product_right_transpose,** $A \leftarrow BC^T$

**product_transpose_transpose,** $A \leftarrow B^T C^T$

**transform_matrix,** $A \leftarrow BCB^T$ (i.e., *similarity transformation*)

**transpose_transform_matrix,** $A \leftarrow B^T CB$

**invert,** $A \leftarrow B^{-1}$

**invert_symmetric,** $A \leftarrow B^{-1}$ **(**$B$ **is symmetric)**

**print** (Prints matrix to standard output.)

### 4.3.2 Vector Functions

In order to use the Math Model vector functions, the following line must be included in the file of the calling function:

```
#include "utils/math/include/vector3.hh"
```

An example call to the vector function *transform_transpose* is:

```
Vector3::transform_transpose (subject_body->composite_body.state.rot.T_parent_this,torque);
```

The vector functions available in the Math Model are listed below. The equivalent math expression is shown along with a brief explanation, if necessary. Overloading has been used to repeat some function names. Note that all vectors are assumed to be of dimension three.

**initialize,** $\bar{A} \leftarrow null$ **(all elements of** $null$ **vector are zero)**

**unit,** $\bar{A} \leftarrow \hat{B}$ (returns 1.0 for only one user selected vector element, the other two elements are returned equal to 0.0)

**fill** (returns matrix with each element equal to user defined input scalar)

**zero_small** (return 0.0 for each element that has an absolute value less than a user specified limit)

**copy,** $\bar{A} \leftarrow \bar{B}$

**dot,** $A \leftarrow B \cdot C$ (vector dot product)

**vmagsq,** $A \leftarrow |B|^2$ (returns a scalar)

**vmag,** $A \leftarrow |B|$ (returns a scalar)

**normalize (in place),** $\hat{A} \leftarrow \bar{A}/|\bar{A}|$

**normalize,** $\hat{A} \leftarrow \bar{B}/|\bar{B}|$

**scale (in place),** $\bar{A} \leftarrow b\bar{A}$

**scale,** $\bar{A} \leftarrow s\bar{B}$

**negate (in place),** $\bar{A} \leftarrow -\bar{A}$

**negate,** $\bar{A} \leftarrow -\bar{B}$

**transform (in place),** $\bar{A} \leftarrow T\bar{A}$ **(**$T$ **is 3x3 transformation matrix)**

**transform,** $\bar{A} \leftarrow T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

**transform_transpose (in place),** $\bar{A} \leftarrow T^T\bar{A}$ **(**$T$ **is 3x3 transformation matrix)**

**transform_transpose,** $\bar{A} \leftarrow T^T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

**incr,** $\bar{A} \leftarrow \bar{A} + \bar{C}$

**incr,** $\bar{A} \leftarrow \bar{A} + \bar{C}_1 + \bar{C}_2$

**decr,** $\bar{A} \leftarrow \bar{A} - \bar{C}$

**decr,** $\bar{A} \leftarrow \bar{A} - \bar{C} + \bar{D}$

**sum,** $\bar{A} \leftarrow \bar{B} + \bar{C}$

**sum,** $\bar{A} \leftarrow \bar{B} + \bar{C} + \bar{D}$

**diff,** $\bar{A} \leftarrow \bar{B} - \bar{C}$

**cross,** $\bar{A} \leftarrow \bar{B} \times \bar{C}$

**scale_incr,** $\bar{A} \leftarrow \bar{A} + s\bar{B}$

**scale_decr,** $\bar{A} \leftarrow \bar{A} - s\bar{B}$

**cross_incr,** $\bar{A} \leftarrow \bar{A} + \bar{B} \times \bar{C}$

**cross_decr,** $\bar{A} \leftarrow \bar{A} - \bar{B} \times \bar{C}$

**transform_incr,** $A \leftarrow \bar{A} + T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

**transform_decr,** $A \leftarrow \bar{A} - T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

**transform_transpose_incr,** $\bar{A} \leftarrow \bar{A} + T^T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

**transform_transpose_decr,** $\bar{A} \leftarrow \bar{A} - T^T\bar{B}$ **(**$T$ **is 3x3 transformation matrix)**

# Chapter 5

# Verification and Validation

## 5.1 Verification

(No verification tests were performed on the Math Model functions.)

## 5.2 Validation

*Test math_1: Matrix Math*

To validate the Math Model matrix functions, 3x3 test matrices were built and sent to the various functions. The results were compared to the same operation(s) performed in Matlab.

The test matrices used were:

$$A = \begin{bmatrix} 7 & 8 & 2 \\ -1 & 5 & -3 \\ 6 & 4 & -7 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 & -3 & 6 \\ -3 & 7 & 4 \\ 6 & 4 & -8 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$f = \begin{bmatrix} 2 \\ -3 \\ 8 \end{bmatrix}$$

$$g = \begin{bmatrix} 7 \\ 2 \\ -1 \end{bmatrix}$$

$$s = 3$$

**Results**

**initialize,** $A \leftarrow null$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

**identity,** $A \leftarrow I$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**cross_matrix,** $A \leftarrow \tilde{f}$

$$A = \begin{bmatrix} 0 & -8 & -3 \\ 8 & 0 & -2 \\ 3 & 2 & 0 \end{bmatrix}$$

**outer_product,** $A \leftarrow \bar{f} \otimes \bar{g}$

$$A = \begin{bmatrix} 14 & 4 & -2 \\ -21 & -6 & 3 \\ 56 & 16 & -8 \end{bmatrix}$$

**negate (in place),** $A \leftarrow (-A)$

$$A = \begin{bmatrix} -7 & -8 & -2 \\ 1 & -5 & 3 \\ -6 & -4 & 7 \end{bmatrix}$$

**negate,** $A \leftarrow (-B)$

$$A = \begin{bmatrix} -5 & 3 & -6 \\ 3 & -7 & -4 \\ -6 & -4 & 8 \end{bmatrix}$$

**transpose (in place),** $A \leftarrow A^T$

$$A = \begin{bmatrix} 7 & -1 & 6 \\ 8 & 5 & 4 \\ 2 & -3 & -7 \end{bmatrix}$$

**transpose,** $A \leftarrow B^T$

$$A = \begin{bmatrix} 5 & -3 & 6 \\ -3 & 7 & 4 \\ 6 & 4 & -8 \end{bmatrix}$$

**scale (in place),** $A \leftarrow sA$ **(**$s$ **is a scalar)**

$$A = \begin{bmatrix} 21 & 24 & 6 \\ -3 & 15 & -9 \\ 18 & 12 & -21 \end{bmatrix}$$

**scale,** $A \leftarrow sB$ **(**$s$ **is a scalar)**

$$A = \begin{bmatrix} 15 & -9 & 18 \\ -9 & 21 & 12 \\ 18 & 12 & -24 \end{bmatrix}$$

**incr (in place),** $A \leftarrow A + C$

$$A = \begin{bmatrix} 8 & 10 & 5 \\ 3 & 10 & 3 \\ 13 & 12 & 2 \end{bmatrix}$$

**decr (in place),** $A \leftarrow A - C$

$$A = \begin{bmatrix} 6 & 6 & -1 \\ -5 & 0 & -9 \\ -1 & -4 & -16 \end{bmatrix}$$

**copy,** $A \leftarrow B$

$$A = \begin{bmatrix} 5 & -3 & 6 \\ -3 & 7 & 4 \\ 6 & 4 & -8 \end{bmatrix}$$

**add,** $A \leftarrow B + C$

$$A = \begin{bmatrix} 6 & -1 & 9 \\ 1 & 12 & 10 \\ 13 & 12 & 1 \end{bmatrix}$$

**subtract,** $A \leftarrow B - C$

$$A = \begin{bmatrix} 4 & -5 & 3 \\ -7 & 2 & -2 \\ -1 & -4 & -17 \end{bmatrix}$$

**product,** $A \leftarrow BC$

$$A = \begin{bmatrix} 35 & 43 & 51 \\ 53 & 61 & 69 \\ -34 & -32 & -30 \end{bmatrix}$$

**product_left_transpose,** $A \leftarrow B^T C$

$$A = \begin{bmatrix} 35 & 43 & 51 \\ 53 & 61 & 69 \\ -34 & -32 & -30 \end{bmatrix}$$

**product_right_transpose,** $A \leftarrow BC^T$

$$A = \begin{bmatrix} 17 & 41 & 65 \\ 23 & 47 & 71 \\ -10 & -4 & 2 \end{bmatrix}$$

**product_transpose_transpose,** $A \leftarrow B^T C^T$

$$A = \begin{bmatrix} 17 & 41 & 65 \\ 23 & 47 & 71 \\ -10 & -4 & 2 \end{bmatrix}$$

**transform_matrix,** $A \leftarrow BCB^T$

$$A = \begin{bmatrix} 352 & 400 & -26 \\ 496 & 544 & 10 \\ -254 & -242 & -92 \end{bmatrix}$$

**transpose_transform_matrix,** $A \leftarrow B^T CB$

$$A = \begin{bmatrix} 352 & 400 & -26 \\ 496 & 544 & 10 \\ -254 & -242 & -92 \end{bmatrix}$$

**invert,** $A \leftarrow B^{-1}$

$$A = \begin{bmatrix} 0.1053 & 0.0000 & 0.0789 \\ 0.0000 & 0.1111 & 0.0556 \\ 0.0789 & 0.0556 & -0.0380 \end{bmatrix}$$

**invert_symmetric,** $A \leftarrow B^{-1}$ **(**$B$ **is symmetric)**

$$A = \begin{bmatrix} 0.1053 & 0.0000 & 0.0789 \\ 0.0000 & 0.1111 & 0.0556 \\ 0.0789 & 0.0556 & -0.0380 \end{bmatrix}$$

*Test math_2: Vector Math*

To validate the Math Model vector functions, 3x1 test vectors were built and sent to the various functions. The results were compared to the same operation(s) performed in Matlab.

The test vectors used were:

$$\bar{A} = \begin{bmatrix} 2 \\ -3 \\ 8 \end{bmatrix}$$

$$\bar{B} = \begin{bmatrix} 7 \\ 2 \\ -1 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} 5 \\ -5 \\ 7 \end{bmatrix}$$

$$\bar{D} = \begin{bmatrix} 4 \\ 3 \\ -1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.819152 & 0.573576 & 0.000000 \\ -0.573576 & 0.819152 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix}$$

$$s = 3$$

**Results**

**initialize,** $\bar{A} \leftarrow null$

$$\bar{A} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

**unit,** $\bar{A} \leftarrow i$

$$\bar{A} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{i=1}$$

**fill**

$$\bar{A} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}_{3}$$

**zero_small**

$$\bar{A} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

**copy,** $\bar{A} \leftarrow \bar{B}$

$$\bar{A} = \begin{bmatrix} 7 \\ 2 \\ -1 \end{bmatrix}$$

**dot,** $A \leftarrow B \cdot C$ (vector dot product)

$$A = 18$$

**vmagsq,** $A \leftarrow |B|^2$ (returns a scalar)

$$A = 54$$

**vmag,** $A \leftarrow |B|$ (returns a scalar)

$$A = 7.3485$$

**normalize (in place),** $\hat{A} \leftarrow \bar{A}/|\bar{A}|$

$$\bar{A} = \begin{bmatrix} 0.2279 \\ -0.3419 \\ 0.9117 \end{bmatrix}$$

**normalize,** $\hat{A} \leftarrow \bar{B}/|\bar{B}|$

$$\bar{A} = \begin{bmatrix} 0.9526 \\ 0.2722 \\ -0.1361 \end{bmatrix}$$

**scale (in place),** $\bar{A} \leftarrow s\bar{A}$

$$\bar{A} = \begin{bmatrix} 6 \\ -9 \\ 24 \end{bmatrix}$$

**scale,** $\bar{A} \leftarrow s\bar{B}$

$$\bar{A} = \begin{bmatrix} 21 \\ 6 \\ -3 \end{bmatrix}$$

**negate (in place),** $\bar{A} \leftarrow -\bar{A}$

$$\bar{A} = \begin{bmatrix} -2 \\ 3 \\ -8 \end{bmatrix}$$

**negate,** $\bar{A} \leftarrow -\bar{B}$

$$\bar{A} = \begin{bmatrix} -7 \\ -2 \\ 1 \end{bmatrix}$$

**transform (in place),** $\bar{A} \leftarrow T\bar{A}$

$$\bar{A} = \begin{bmatrix} -0.0824 \\ -3.6046 \\ 8.0000 \end{bmatrix}$$

**transform,** $\bar{A} \leftarrow T\bar{B}$

$$\bar{A} = \begin{bmatrix} 6.8812 \\ -2.3767 \\ -1.0000 \end{bmatrix}$$

**transform_transpose (in place),** $\bar{A} \leftarrow T^T\bar{A}$

$$\bar{A} = \begin{bmatrix} 3.3590 \\ -1.3102 \\ 8.0000 \end{bmatrix}$$

**transform_transpose,** $\bar{A} \leftarrow T^T\bar{B}$

$$\bar{A} = \begin{bmatrix} 4.5869 \\ 5.6533 \\ -1.0000 \end{bmatrix}$$

**incr,** $\bar{A} \leftarrow \bar{A} + \bar{C}$

$$\bar{A} = \begin{bmatrix} 7 \\ -8 \\ 15 \end{bmatrix}$$

**incr,** $\bar{A} \leftarrow \bar{A} + \bar{C} + \bar{D}$

$$\bar{A} = \begin{bmatrix} 11 \\ -5 \\ 14 \end{bmatrix}$$

**decr,** $\bar{A} \leftarrow \bar{A} - \bar{C}$

$$\bar{A} = \begin{bmatrix} -3 \\ 2 \\ 1 \end{bmatrix}$$

**decr,** $\bar{A} \leftarrow \bar{A} - \bar{C} - \bar{D}$

$$\bar{A} = \begin{bmatrix} -7 \\ -1 \\ 2 \end{bmatrix}$$

**sum,** $\bar{A} \leftarrow \bar{B} + \bar{C}$

$$\bar{A} = \begin{bmatrix} 12 \\ -3 \\ 6 \end{bmatrix}$$

**sum,** $\bar{A} \leftarrow \bar{B} + \bar{C} + \bar{D}$

$$\bar{A} = \begin{bmatrix} 16 \\ 0 \\ 5 \end{bmatrix}$$

**diff,** $\bar{A} \leftarrow \bar{B} - \bar{C}$

$$\bar{A} = \begin{bmatrix} 2 \\ 7 \\ -8 \end{bmatrix}$$

**cross,** $\bar{A} \leftarrow \bar{B} \times \bar{C}$

$$\bar{A} = \begin{bmatrix} 9 \\ -54 \\ -45 \end{bmatrix}$$

**scale_incr,** $\bar{A} \leftarrow \bar{A} + s\bar{B}$

$$\bar{A} = \begin{bmatrix} 23 \\ 3 \\ 5 \end{bmatrix}$$

**scale_decr,** $\bar{A} \leftarrow \bar{A} - s\bar{B}$

$$\bar{A} = \begin{bmatrix} -19 \\ -9 \\ 11 \end{bmatrix}$$

**cross_incr,** $\bar{A} \leftarrow \bar{A} + \bar{B} \times \bar{C}$

$$\bar{A} = \begin{bmatrix} 11 \\ -57 \\ -37 \end{bmatrix}$$

**cross_decr,** $\bar{A} \leftarrow \bar{A} - \bar{B} \times \bar{C}$

$$\bar{A} = \begin{bmatrix} -7 \\ 51 \\ 53 \end{bmatrix}$$

**transform_incr,** $A \leftarrow \bar{A} + T\bar{B}$

$$\bar{A} = \begin{bmatrix} 8.8812 \\ -5.3767 \\ 7.0000 \end{bmatrix}$$

**transform_decr,** $A \leftarrow \bar{A} - T\bar{B}$

$$\bar{A} = \begin{bmatrix} -4.8812 \\ -0.6233 \\ 9.0000 \end{bmatrix}$$

**transform_transpose_incr,** $\bar{A} \leftarrow \bar{A} + T^T\bar{B}$

$$\bar{A} = \begin{bmatrix} 6.5869 \\ 2.6533 \\ 7.0000 \end{bmatrix}$$

**transform_transpose_decr,** $\bar{A} \leftarrow \bar{A} - T^T\bar{B}$

$$\bar{A} = \begin{bmatrix} -2.5869 \\ -8.6533 \\ 9.0000 \end{bmatrix}$$

## 5.3 Requirements Traceability

The table below cross-references each requirement of the Math Model to a corresponding verification and/or validation test described in sections 5.1 and 5.2 of this document.

Table 5.1: Requirements Traceability

| **Requirement** | **Inspection or Test** |
|---|---|
| math_1 - Matrix Math Operations | Test math_1 - Matrix Math |
| math_2 - Vector Math Operations | Test math_2 - Vector Math |

# Bibliography


[1] Jackson, A., Thebeau, C. JSC Engineering Orbital Dynamics. Technical Report JSC-61777-docs, NASA, Johnson Space Center, Houston, Texas, July 2022.

[2] Erwin Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., New York, 8th edition, 1997.

[3] NASA. NASA Software Engineering Requirements. Technical Report NPR-7150.2, NASA, NASA Headquarters, Washington, D.C., September 2004.