

2.9.26

Dhanush Kumar A - AI25BTECH11010

September 10, 2025

Question

If $f(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$, prove that $f(\alpha)f(-\beta) = f(\alpha - \beta)$.

Solution

We have

$$f(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (1)$$

$$f(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$f(-\beta) = \begin{pmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

$$f(\alpha)f(-\beta) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

Solution

$$= \begin{pmatrix} \cos \alpha \cos \beta + \sin \alpha \sin \beta & \cos \alpha \sin \beta - \sin \alpha \cos \beta & 0 \\ \sin \alpha \cos \beta - \cos \alpha \sin \beta & \sin \alpha \sin \beta + \cos \alpha \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (6)$$

$$= \begin{pmatrix} \cos(\alpha - \beta) & -\sin(\alpha - \beta) & 0 \\ \sin(\alpha - \beta) & \cos(\alpha - \beta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (7)$$

$$= f(\alpha - \beta). \quad (8)$$

Thus proved.

Python code - Verify the result

```
import numpy as np

def f(theta):
    return np.array([
        np.cos(theta), -np.sin(theta), 0],
        [np.sin(theta), np.cos(theta), 0],
        [0, 0, 1]
    ])

# Take input
alpha = float(input("Enter alpha (in radians): "))
beta = float(input("Enter beta (in radians): "))

# Compute both sides
lhs = f(alpha) @ f(-beta)
rhs = f(alpha - beta)
```

Python code - Verify the result

```
# Check equality (within tolerance, since floats may not be exact
)
if np.allclose(lhs, rhs, atol=1e-9):
    print("Verified:  $f(\alpha) f(-\beta) = f(\alpha - \beta)$ ")
else:
    print(" Not equal")
    print("LHS =\n", lhs)
    print("RHS =\n", rhs)
```

Output of Python code

```
Enter alpha (in radians): 4
Enter beta (in radians): 5
Verified:  $f(\alpha) f(-\beta) = f(\alpha - \beta)$ 
```

C code - Verify the result

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "/home/dhanush-kumar-a/ee1030-2025/ai25btech11010/matgeo/2.9.26/codes/libs/matfun.h"

int main() {
    double alpha, beta;
    printf("Enter alpha (in radians): ");
    scanf("%lf", &alpha);
    printf("Enter beta (in radians): ");
    scanf("%lf", &beta);

    // Step 1: Create 3x3 rotation matrices
    double **F_alpha = createMat(3,3);
    double **F_minus_beta = createMat(3,3);
    double **F_alpha_minus_beta = createMat(3,3);
```


C code - Verify the result

```
// Fill the matrices manually
F_alpha[0][0] = cos(alpha); F_alpha[0][1] = -sin(alpha);
    F_alpha[0][2] = 0;
F_alpha[1][0] = sin(alpha); F_alpha[1][1] = cos(alpha);
    F_alpha[1][2] = 0;
F_alpha[2][0] = 0; F_alpha[2][1] = 0; F_alpha[2][2] = 1;

F_minus_beta[0][0] = cos(-beta); F_minus_beta[0][1] = -sin(-
    beta); F_minus_beta[0][2] = 0;
F_minus_beta[1][0] = sin(-beta); F_minus_beta[1][1] = cos(-
    beta); F_minus_beta[1][2] = 0;
F_minus_beta[2][0] = 0; F_minus_beta[2][1] = 0; F_minus_beta
    [2][2] = 1;

F_alpha_minus_beta[0][0] = cos(alpha - beta);
    F_alpha_minus_beta[0][1] = -sin(alpha - beta);
    F_alpha_minus_beta[0][2] = 0;
F_alpha_minus_beta[1][0] = sin(alpha - beta);
```

C code - Verify the result

```
// Step 2: Multiply f(alpha) * f(-beta)
double **lhs = Matmul(F_alpha, F_minus_beta, 3, 3, 3);

// Step 3: Compare lhs with f(alpha - beta)
int equal = 1;
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        if(fabs(lhs[i][j] - F_alpha_minus_beta[i][j]) > 1e-9){
            equal = 0;
            break;
        }
    }
    if(!equal) break;
}
```

C code - Verify the result

```
// Step 4: Print results
if(equal){
    printf("\nVerified:  $f(\alpha) * f(-\beta) = f(\alpha - \beta)$ 
           \n");
} else {
    printf("\nNot equal!\n\nLHS =\n");
    printMat(lhs, 3, 3);
    printf("\nRHS =\n");
    printMat(F_alpha_minus_beta, 3, 3);
}
```

C code - Verify the result

```
// Step 5: Free memory
for(int i=0;i<3;i++){
    free(F_alpha[i]); free(F_minus_beta[i]); free(
        F_alpha_minus_beta[i]);
    free(lhs[i]);
}
free(F_alpha); free(F_minus_beta); free(F_alpha_minus_beta);
free(lhs);

return 0;
}
```

Output of C code

```
Enter alpha (in radians): 2  
Enter beta (in radians): 1.23
```

```
Verified:  $f(\alpha) * f(-\beta) = f(\alpha - \beta)$ 
```