12.40

Harsha-EE25BTECH11026

September 12,2025

Question

If a rectangle is deformed into a parallelogram of equal area by simple shear deformation (with shear strain γ) parallel to the abscissa, the displacement matrix is ___________.

1. $\begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix}$
2. $\begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$
3. $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
4. $\begin{pmatrix} 0 & \gamma \\ 1 & 0 \end{pmatrix}$

Theoretical Solution

Due to the shear deformation, let x', y' be the new coordinates. As the deformation is along the direction of abscissa,

$$\therefore y' = y \tag{1}$$

Let the displacement due to the shear deformation be Δh.

$$\gamma = \frac{\Delta h}{y} \tag{2}$$

$$\therefore \Delta h = \gamma y \tag{3}$$

$$\implies x' = x + \Delta h = x + \gamma y \tag{4}$$

From (1) and (4),

$$\therefore \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + \gamma y \\ y \end{pmatrix} = \begin{pmatrix} 1 & \gamma \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5}$$

C Code -Finding displacement matrix

```c
#include<stdio.h>

void displacement_matrix(double *matrix, double gamma){
        matrix[0]=1;
        matrix[1]=gamma;
        matrix[2]=0;
        matrix[3]=1;
}
```

Python+C code

```python
import ctypes
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

lib=ctypes.CDLL("./lib_dis_matrix.so")

lib.displacement_matrix.argtypes=(np.ctypeslib.ndpointer(dtype=np
    .float64, ndim=1, flags="C_CONTIGUOUS"),ctypes.c_double)

def shear_matrix(gamma: float) -> np.ndarray:
    mat = np.zeros(4, dtype=np.float64) # flattened 2x2
    lib.displacement_matrix(mat, gamma)
    return mat.reshape((2, 2))
```

Python+C code

```
# Example usage
gamma = 0.5
A= shear_matrix(gamma)
B=sp.Matrix(A)
print("Displacement matrix:\n")
sp.pprint(B)

# Define rectangle corners (counter-clockwise)
rect = np.array([[0, 0],
                [2, 0],
                [2, 1],
                [0, 1],
                [0, 0]]) # closed loop

# Apply shear transformation
deformed = rect @ B.T
```

Python+C code

```
# Plot
plt.figure(figsize=(8, 8))
plt.plot(rect[:, 0], rect[:, 1], 'b-', label='Original Rectangle')
plt.plot(deformed[:, 0], deformed[:, 1], 'r--', label='Sheared Parallelogram')
plt.fill(rect[:, 0], rect[:, 1], 'b', alpha=0.2)
plt.fill(deformed[:, 0], deformed[:, 1], 'r', alpha=0.2)
plt.axhline(0, color='k', linewidth=0.5)
plt.axvline(0, color='k', linewidth=0.5)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend(loc="upper right")
plt.title(f"Simple Shear Deformation (gamma = {gamma})")
plt.savefig("/home/user/Matrix Theory: workspace/Matgeo_assignments/12.456/figs/Figure_1.png")
plt.show()
```

Python code

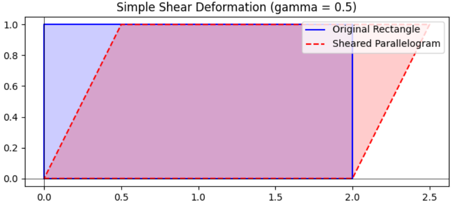
```python
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

# Example
gamma = 0.5
A=sp.Matrix([[1,gamma],[0,1]])
B=np.matrix([[1,gamma],[0,1]])
print("Displacement Matrix:\n")
sp.pprint(A)


# Define rectangle corners (counter-clockwise)
rect = np.array([[0, 0],
                 [2, 0],
                 [2, 1],
                 [0, 1],
                 [0, 0]]) # closed loop
```

```
# Apply shear transformation
deformed = rect @ B.T
# Plot
plt.figure(figsize=(8, 8))
plt.plot(rect[:, 0], rect[:, 1], 'b-', label='Original Rectangle'
    )
plt.plot(deformed[:, 0], deformed[:, 1], 'r--', label='Sheared
    Parallelogram')
plt.fill(rect[:, 0], rect[:, 1], 'b', alpha=0.2)
plt.fill(deformed[:, 0], deformed[:, 1], 'r', alpha=0.2)
plt.axhline(0, color='k', linewidth=0.5)
plt.axvline(0, color='k', linewidth=0.5)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend(loc="upper right")
plt.title(f"Simple Shear Deformation (gamma = {gamma})")
plt.savefig("/home/user/Matrix Theory: workspace/
    Matgeo_assignments/12.456/figs/Figure_1.png")
plt.show()
```

Plot



Simple Shear Deformation (gamma = 0.5)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
Original Rectangle
Sheared Parallelogram