12.755

Bhargav - EE25BTECH11013

October 10, 2025

Question

Question:
Which one of the following vectors is an eigenvector corresponding to the eigenvalue $\lambda = 1$ for the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \tag{1}$$

is

Solution

The eigenvalue of the the the matrix **A** can be found out by (where $\lambda = 1$ is the eigenvalue, **x** is the eigenvector, **I** is the identity matrix)

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \implies \mathbf{A}\mathbf{x} = \mathbf{x} \tag{2}$$

$$(\mathbf{A} - \mathbf{I})\,\mathbf{x} = \mathbf{0} \tag{3}$$

$$\implies \begin{pmatrix} 0 & 1 & 0 \\ 1 & -2 & 0 \\ 1 & -1 & 0 \end{pmatrix} \mathbf{x} = \mathbf{0} \tag{4}$$

Solution

This can be solved by representing it as an augmented matrix and using row elimination

$$\left(\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 1 & -2 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{array}\right) \xleftrightarrow{R_1 \leftrightarrow R_2} \left(\begin{array}{ccc|c} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{array}\right) \xleftrightarrow{R_3 \leftarrow R_3 - R_1} \tag{5}$$

$$\left(\begin{array}{ccc|c} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right) \xleftrightarrow{R_3 \leftarrow R_3 - R_2} \left(\begin{array}{ccc|c} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \tag{6}$$

Solution

Thus $\mathbf{x} = t \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ where $t \in \mathbf{R}$

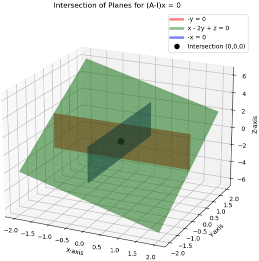
So, the eigenvector of $\mathbf{A}$ is $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

This can be further verified by the intersection of planes

$$x - 2y = 0 \tag{7}$$

$$y = 0 \tag{8}$$

Plot



Intersection of Planes for (A-I)x = 0
-y = 0
x - 2y + z = 0
-x = 0
Intersection (0,0,0)
X-axis
Y-axis
Z-axis
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
−6
−4
−2
0
2
4
6

C Code

```c
#include <stdio.h>
#include <stdlib.h>

void generate_plane1(double *Y, double *Z, double *X, int N) {
    for(int i = 0; i < N*N; i++) {
        X[i] = 2 * Y[i];
    }
}

void generate_plane2(double *X, double *Z, double *Y, int N) {
    for(int i = 0; i < N*N; i++) {
        Y[i] = 0;
    }
}
```

C Code

```c

void generate_intersection_line(double *x, double *y, double *z,
     int N) {
    double dz = 4.0/(N-1);
    for(int i = 0; i < N; i++) {
        x[i] = 0;
        y[i] = 0;
        z[i] = -2 + i*dz;
    }
}
```

Python + C Code

```
import numpy as np
import ctypes
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

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

N = 40
d = np.linspace(-2, 2, N)
Y, Z = np.meshgrid(d, d)
X = np.zeros_like(Y)
```

Python + C Code

```
# Convert to C-contiguous arrays
Y_c = np.ascontiguousarray(Y, dtype=np.double)
Z_c = np.ascontiguousarray(Z, dtype=np.double)
X_c = np.ascontiguousarray(X, dtype=np.double)

# Set argument types
lib.generate_plane1.argtypes = [np.ctypeslib.ndpointer(dtype=np.
    double),
                             np.ctypeslib.ndpointer(dtype=np.
                                 double),
                             np.ctypeslib.ndpointer(dtype=np.
                                 double),
                             ctypes.c_int]

lib.generate_plane2.argtypes = [np.ctypeslib.ndpointer(dtype=np.
    double),
                             np.ctypeslib.ndpointer(dtype=np.
                                 double),
                             np.ctypeslib.ndpointer(dtype=np.
```

Python + C Code

```python
# Generate plane1
lib.generate_plane1(Y_c, Z_c, X_c, N)

# Plotting
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X_c, Y_c, Z_c, alpha=0.5, color='g')

# Plane 2
X2, Z2 = np.meshgrid(d, d)
Y2 = np.zeros_like(X2)
lib.generate_plane2(X2, Z2, Y2, N)
ax.plot_surface(X2, Y2, Z2, alpha=0.5, color='r')

# Intersection line
x_line = np.zeros(N)
y_line = np.zeros(N)
```

Python + C Code

```python
z_line = np.zeros(N)
lib.generate_intersection_line(x_line, y_line, z_line, N)
ax.plot3D(x_line, y_line, z_line, color='k', linewidth=3)

# Intersection point at origin
ax.scatter(0,0,0,color='black',s=80)

# Legend
legend_elements = [
    Line2D([0],[0], color='g', lw=4, alpha=0.5, label='x - 2y = 0'),
    Line2D([0],[0], color='r', lw=4, alpha=0.5, label='y = 0'),
    Line2D([0],[0], color='k', lw=3, label='Intersection Line (x=0, y=0)'),
    Line2D([0],[0], marker='o', color='k', label='Origin (0,0,0)', markersize=8, linestyle='')
]
ax.legend(handles=legend_elements)
```

Python Code

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D

fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

d = np.linspace(-2, 2, 40)
Y, Z = np.meshgrid(d, d) # create grid

# Plane 1: x - 2y = 0 →X = 2Y, Z free
X1 = 2 * Y
Z1 = Z
ax.plot_surface(X1, Y, Z1, alpha=0.5, color='g')

# Plane 2: y = 0 →Y=0, X and Z free
X2, Z2 = np.meshgrid(d, d)
Y2 = np.zeros_like(X2)
ax.plot_surface(X2, Y2, Z2, alpha=0.5, color='r')
```

Python Code

```python
# Intersection line: x=0, y=0, z free
z_line = np.linspace(-2, 2, 20)
x_line = np.zeros_like(z_line)
y_line = np.zeros_like(z_line)
ax.plot3D(x_line, y_line, z_line, color='k', linewidth=3)

# Intersection point at origin
ax.scatter(0, 0, 0, color='black', s=80)
```

Python Code

```python

# Legend
legend_elements = [
    Line2D([0],[0], color='g', lw=4, alpha=0.5, label='x - 2y = 0'),
    Line2D([0],[0], color='r', lw=4, alpha=0.5, label='y = 0'),
    Line2D([0],[0], color='k', lw=3, label='Intersection Line (x=0, y=0)'),
    Line2D([0],[0], marker='o', color='k', label='Origin (0,0,0)', markersize=8, linestyle='')
]
ax.legend(handles=legend_elements)

ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('Intersection of Planes: x - 2y = 0 and y = 0')
ax.view_init(elev=25, azim=-60)
plt.savefig("/mnt/c/Users/bharg/Documents/backupmatrix/
```