12.145

INDHIRESH S - EE25BTECH11027

10 October, 2025

Question

If a square matrix **A** is real and symmetric, then the eigenvalues

1. are always real
2. are always real and positive
3. are always real and non-negative
4. occur in complex conjupairs

Theoretical Solution

The correct statement is (1). This is a fundamental property of real symmetric matrices.
Let $\mathbf{A}$ be a real and symmetric matrix, which means

$$\mathbf{A} = \mathbf{A}^T \quad \textit{and} \quad \bar{\mathbf{A}} = \mathbf{A} \tag{1}$$

Let λ be an eigenvalue of $\mathbf{A}$ with a corresponding non-zero eigenvector $\mathbf{x}$.
The eigenvalue equation is:

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

To prove that λ is real, we must show it is equal to its own complex conjugate, i.e., $\lambda = \bar{\lambda}$. We take the conjugate transpose (Hermitian conjugate) on both sides:

$$(\mathbf{A}\mathbf{x})^H = (\lambda\mathbf{x})^H \tag{3}$$

$$\mathbf{x}^H\mathbf{A}^H = \bar{\lambda}\mathbf{x}^H \tag{4}$$

Theoretical solution

For a real and symmetric matrix, its conjugate transpose is itself:

$$\mathbf{A}^H = \bar{\mathbf{A}}^T = \mathbf{A}^T = \mathbf{A} \tag{5}$$

Substituting this into the Eq.4 gives:

$$\mathbf{x}^H\mathbf{A} = \bar{\lambda}\mathbf{x}^H \tag{6}$$

Now, we pre-multiply the Eq.2 by $\mathbf{x}^H$:

$$\mathbf{x}^H\mathbf{A}\mathbf{x} = \lambda(\mathbf{x}^H\mathbf{x}) \tag{7}$$

And we post-multiply Eq.6 by $\mathbf{x}$:

$$\mathbf{x}^H\mathbf{A}\mathbf{x} = \bar{\lambda}(\mathbf{x}^H\mathbf{x}) \tag{8}$$

Theoretical solution

By comparing Eq.7 and Eq.8 , we see that:

$$\lambda(\mathbf{x}^H\mathbf{x}) = \bar{\lambda}(\mathbf{x}^H\mathbf{x}) \tag{9}$$

This can be rearranged to:

$$(\lambda - \bar{\lambda})(\mathbf{x}^H\mathbf{x}) = 0 \tag{10}$$

Since an eigenvector $\mathbf{x}$ is non-zero by definition, its magnitude $||\mathbf{x}||^2$ is a positive real number.So,

$$\lambda - \bar{\lambda} = 0 \tag{11}$$

$$\lambda = \bar{\lambda} \tag{12}$$

A number that is equal to its own complex conjugate must be a real number.

From above statement it is clear that eigenvalues are always real

Options (2) and (3) are incorrect because a real symmetric matrix can have negative eigenvalues.
Example:
Consider the matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{13}$$

This matrix is real and symmetric.Now finding the eihgen value for the matrix:

$$\begin{vmatrix} -\lambda & 1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 - 1 = 0 \tag{14}$$

The eigenvalues are

$$\lambda = 1 \;\; and \;\; \lambda = -1 \tag{15}$$

Therefore the eigenvalues can be negative
Option (1) is correct

C Code

```c
#include <math.h>
void find_eigenvalues_2x2(double a, double b, double c, double d,
     double* eig1, double* eig2) {
    double trace = a + d;
    double determinant = a * d - b * c;
    double discriminant = trace * trace - 4 * determinant;

    if (discriminant >= 0) {
        double sqrt_discriminant = sqrt(discriminant);
        *eig1 = (trace + sqrt_discriminant) / 2.0;
        *eig2 = (trace - sqrt_discriminant) / 2.0;
    }
}
```

Python Code

```python
import ctypes


eigen_lib = ctypes.CDLL('./eigen.so')

# Define the function signature (argument types and return type).
# This helps ctypes correctly handle data marshalling.
find_eigs = eigen_lib.find_eigenvalues_2x2
find_eigs.argtypes = [ctypes.c_double, ctypes.c_double,
                      ctypes.c_double, ctypes.c_double,
                      ctypes.POINTER(ctypes.c_double),
                      ctypes.POINTER(ctypes.c_double)]
find_eigs.restype = None # The C function returns void
```

Python Code

```
 # Define the matrix elements
a, b = 0.0, 1.0
c, d = 1.0, 0.0

# Create C-compatible double variables to store the results
eig1 = ctypes.c_double()
eig2 = ctypes.c_double()

# Call the C function from Python
# We pass the result variables by reference using byref()
find_eigs(a, b, c, d, ctypes.byref(eig1), ctypes.byref(eig2))

print(Eigenvalues found using Python wrapper for C library:)
# Access the value of the ctypes object with .value
print(fLambda 1: {eig1.value})
print(fLambda 2: {eig2.value})
```