2.9.11

Namaswi-EE25BTECH11060

september 10,2025

Question

If $\bar{a}$ and $\bar{b}$ are unit vectors and θ is angle between them then prove that $\sin\frac{\theta}{2} = \frac{1}{2}\,|\bar{a} - \bar{b}|$

$$\sin^2 \frac{\theta}{2} = \frac{1}{4} |\bar{a} - \bar{b}|^2$$

Solution

consider RHS,

$$\implies \frac{1}{4}||\hat{a} - \hat{b}||^2 \tag{1}$$

$$= \frac{1}{4}\left(||\hat{a}||^2 + ||\hat{b}||^2 - 2\hat{a}^\top \hat{b}\right) \tag{2}$$

$$= \frac{1}{4}\left(2 - 2\cos\theta\right) \tag{3}$$

$$= \frac{1}{2}\left(2\sin^2\frac{\theta}{2}\right) \tag{4}$$

$$\tag{5}$$

$$= \sin^2\frac{\theta}{2} \tag{6}$$

$$= LHS \tag{7}$$

Hence, $\sin\frac{\theta}{2} = \frac{1}{2}\ |\bar{a} - \bar{b}|$

C Code

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

double dot_product(double a[], double b[]) {
    return a[0]*b[0] + a[1]*b[1];
}

double norm(double v[]) {
    return sqrt(v[0]*v[0] + v[1]*v[1]);
}

void normalize(double v[]) {
    double n = norm(v);
    if (n != 0) {
        v[0] /= n;
        v[1] /= n;
    }
}
```

C Code

```c
int main() {
    double a[2] = {1, 2};
    double b[2] = {2, 1};

    normalize(a);
    normalize(b);

    double cos_theta = dot_product(a, b);
    double theta = acos(cos_theta);

    double diff[2] = {0.5 * (a[0] - b[0]), 0.5 * (a[1] - b[1])};
    double lhs = norm(diff);
```

C Code

```c
double rhs = sin(theta / 2.0);

    printf("Angle  (in degrees): %.6f\n", theta * (180.0 / M_PI));
    printf("||0.5(a - b)|| = %.6f\n", lhs);
    printf("sin( / 2) = %.6f\n", rhs);
    printf("Difference = %.6e\n", fabs(lhs - rhs));

    return 0;
}
```

Python Code

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Arc

def plot_vectors_with_angle(a, b):
    a = np.array(a)
    b = np.array(b)

    # Calculate angle (in radians and degrees)
    dot_product = np.dot(a, b)
    norm_a = np.linalg.norm(a)
    norm_b = np.linalg.norm(b)
    cos_theta = dot_product / (norm_a * norm_b)
    theta_rad = np.arccos(np.clip(cos_theta, -1.0, 1.0))
    theta_deg = np.degrees(theta_rad)
```

Python Code

```
# Setup plot
   fig, ax = plt.subplots()
   ax.set_aspect('equal')
   ax.grid(True)

   # Calculate plot limits
   max_val = max(np.linalg.norm(a), np.linalg.norm(b)) + 1
   ax.set_xlim(-1, max_val)
   ax.set_ylim(-1, max_val)

   # Plot vectors
   origin = [0, 0]
   ax.quiver(*origin, *a, angles='xy', scale_units='xy', scale=1, color='r', label='a')
   ax.quiver(*origin, *b, angles='xy', scale_units='xy', scale=1, color='b', label='b')
```

Python Code

```python
# Draw angle arc
arc_radius = 0.5
arc = Arc(origin, arc_radius*2, arc_radius*2, angle=0,
          theta1=0, theta2=theta_deg, color='green')
ax.add_patch(arc)

# Annotate angle
mid_angle = theta_rad / 2
label_radius = arc_radius * 1.4
x_text = label_radius * np.cos(mid_angle)
y_text = label_radius * np.sin(mid_angle)
ax.text(x_text, y_text, f' = {theta_deg:.1f}', fontsize=12,
    color='green')
```

Python Code

```
        ax.add_patch(arc)

    # Annotate angle
    mid_angle = theta_rad / 2
    label_radius = arc_radius * 1.4
    x_text = label_radius * np.cos(mid_angle)
    y_text = label_radius * np.sin(mid_angle)
    ax.text(x_text, y_text, f' = {theta_deg:.1f}', fontsize=12,
        color='green')
```

C and Python Code

```
    # Labels and title
    ax.legend()
    ax.set_xlabel('X-axis')
    ax.set_ylabel('Y-axis')
    ax.set_title('Angle Between Vectors a and b')
    plt.show()

# Example usage with any two 2D vectors:
# Replace these with your own vectors
a = [3, 2]
b = [2, 4]

plot_vectors_with_angle(a, b)
```

C and Python Code

```python
import ctypes
import numpy as np
import platform
import os

# Load C shared library
if platform.system() == 'Windows':
```

C and python Code

```
    lib = ctypes.CDLL('./vector_math.dll')
else:
    lib = ctypes.CDLL('./libvector.so')
# Define argtypes and restype
lib.half_diff_norm.argtypes = [ctypes.c_double]*4
lib.half_diff_norm.restype = ctypes.c_double
lib.sin_theta_over_2.argtypes = [ctypes.c_double]
lib.sin_theta_over_2.restype = ctypes.c_double
# Define two vectors (they will be normalized in Python)
a = np.array([1, 2], dtype=np.float64)
b = np.array([2, 1], dtype=np.float64)
# Normalize
a_hat = a / np.linalg.norm(a)
b_hat = b / np.linalg.norm(b)
```

C and Python Code

```
    # Compute dot product
dot_ab = np.dot(a_hat, b_hat)
# Call C functions
lhs = lib.half_diff_norm(a_hat[0], a_hat[1], b_hat[0], b_hat[1])
rhs = lib.sin_theta_over_2(dot_ab)
# Show results
print(f'||0.5(a - b)|| = {lhs:.6f}')
print(f'sin( / 2) = {rhs:.6f}')
print(f'Difference = {abs(lhs - rhs):.6e}')
import ctypes
import numpy as np
import platform
import os
# Load C shared library
if platform.system() == 'Windows':
    lib = ctypes.CDLL('./vector_math.dll')
else:
    lib = ctypes.CDLL('./libvector.so')
```

C and Python Code

```python

# Define argtypes and restype
lib.half_diff_norm.argtypes = [ctypes.c_double]*4
lib.half_diff_norm.restype = ctypes.c_double

lib.sin_theta_over_2.argtypes = [ctypes.c_double]
lib.sin_theta_over_2.restype = ctypes.c_double

# Define two vectors (they will be normalized in Python)
a = np.array([1, 2], dtype=np.float64)
b = np.array([2, 1], dtype=np.float64)

# Normalize
a_hat = a / np.linalg.norm(a)
b_hat = b / np.linalg.norm(b)
```

C and Python Code

```python
# Compute dot product
dot_ab = np.dot(a_hat, b_hat)

# Call C functions
lhs = lib.half_diff_norm(a_hat[0], a_hat[1], b_hat[0], b_hat[1])
rhs = lib.sin_theta_over_2(dot_ab)

# Show results
print(f'||0.5(a - b)|| = {lhs:.6f}')
print(f'sin( / 2) = {rhs:.6f}')
print(f'Difference = {abs(lhs - rhs):.6e}')
```