5.8.8

Application problem

EE25BTECH11010 - Arsh Dhoke

Question

Places A and B are 100km apart on a highway. One car starts from A and another from B at the same time. If the cars travel in the same direction at different speeds, they meet in 5 hrs. If they travel towards each other, they meet in 1 hr. Find the speeds of the two cars.

Solution: Step 1

Cars meet in 1 hr when moving towards each other:

$$v_1 + v_2 = \frac{100}{1} = 100 \tag{1}$$

Cars meet in 5 hr when moving in the same direction:

$$v_1 - v_2 = \frac{100}{5} = 20 \tag{2}$$

Solution: Vector Form

The equations can be expressed as

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 100, \tag{3}$$

$$\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 20. \tag{4}$$

Let

$$\mathbf{r_1} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad \mathbf{r_2} = \begin{pmatrix} 1 & -1 \end{pmatrix}. \tag{5}$$

Then $\mathbf{r_1} \cdot \mathbf{r_2} = 0$, showing that the two rows are orthogonal.

Solution: Using Orthogonality

Let

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = c_1 \mathbf{r_1}^T + c_2 \mathbf{r_2}^T. \tag{6}$$

Taking dot products with $\mathbf{r_1}$ and $\mathbf{r_2}$,

$$\mathbf{r_1} \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = c_1 \mathbf{r_1} \cdot \mathbf{r_1} = 100, \tag{7}$$

$$\mathbf{r_2} \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = c_2 \mathbf{r_2} \cdot \mathbf{r_2} = 20. \tag{8}$$

Solution: Compute Coefficients

Since $\|\mathbf{r_1}\|^2 = \|\mathbf{r_2}\|^2 = 2$,

$$c_1 = \frac{100}{2} = 50, \quad c_2 = \frac{20}{2} = 10. \tag{9}$$

Therefore,

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 50\,\mathbf{r_1}^T + 10\,\mathbf{r_2}^T \tag{10}$$

$$= 50\begin{pmatrix} 1 \\ 1 \end{pmatrix} + 10\begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{11}$$

$$= \begin{pmatrix} 60 \\ 40 \end{pmatrix}. \tag{12}$$

Final Answer

$$\boxed{v_1 = 60 \text{ km/h}}, \quad \boxed{v_2 = 40 \text{ km/h}} \tag{13}$$

Graphical Representation

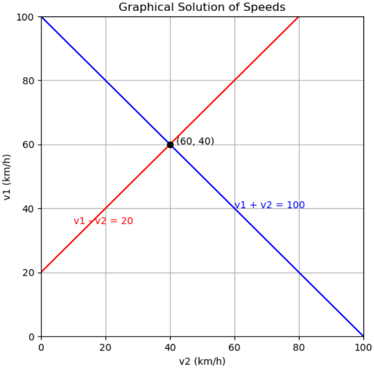

Figure: Graphical solution of speeds

C Code

```c
#include <stdio.h>

void solve_car_speeds(double distance, double time_same_dir,
    double time_towards, double *v1, double *v2) {
    double sum_speeds = distance / time_towards;
    double diff_speeds = distance / time_same_dir;

    *v1 = (sum_speeds + diff_speeds) / 2.0;
    *v2 = sum_speeds - *v1;
}
```

Python Code

```
import matplotlib.pyplot as plt
import numpy as np

# Define v2 range
v2 = np.linspace(0, 100, 200)

# Equations
v1_towards = 100 - v2 # v1 + v2 = 100
v1_same = 20 + v2 # v1 - v2 = 20

# Plot
plt.figure(figsize=(6,6))
plt.plot(v2, v1_towards, color='blue')
plt.plot(v2, v1_same, color='red')

# Intersection point
v2_meet = 40
v1_meet = 60
```

Python Code

```
plt.plot(v2_meet, v1_meet, 'ko') # solution point

# Annotate the lines
plt.text(60, 40, 'v1 + v2 = 100', color='blue')
plt.text(10, 35, 'v1 - v2 = 20', color='red')

# Annotate solution point
plt.text(v2_meet + 2, v1_meet, '(60, 40)', color='black')

plt.xlabel('v2 (km/h)')
plt.ylabel('v1 (km/h)')
plt.title('Graphical Solution of Speeds')
plt.xlim(0, 100)
plt.ylim(0, 100)
plt.grid(True)
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /5.8.8/figs/speed.png")
plt.show()
```

Python+ C Code

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

# Load the shared library
lib = ctypes.CDLL('./code.so')

# Define argument and return types
lib.solve_car_speeds.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double)
]

# Function to call C function
def get_speeds(distance, time_same_dir, time_towards):
    v1 = ctypes.c_double()
    v2 = ctypes.c_double()
```

Python+ C Code

```python
    lib.solve_car_speeds(distance, time_same_dir, time_towards,
        ctypes.byref(v1), ctypes.byref(v2))
    return v1.value, v2.value

# Example
distance = 100.0
time_same_dir = 5.0
time_towards = 1.0

v1, v2 = get_speeds(distance, time_same_dir, time_towards)
print(f"v1 = {v1}, v2 = {v2}")

# Plot the lines v1 + v2 = distance / time_towards and v1 - v2 =
    distance / time_same_dir
v2_vals = np.linspace(0, 100, 200)
v1_towards = distance / time_towards - v2_vals
v1_same = distance / time_same_dir + v2_vals
```

Python+ C Code

```
plt.plot(v2_vals, v1_towards, color='blue')
plt.plot(v2_vals, v1_same, color='red')
plt.plot(v2, v1, 'ko') # intersection
plt.xlabel('v2 (km/h)')
plt.ylabel('v1 (km/h)')
plt.grid(True)
plt.savefig("/home/arsh-dhoke/ee1030-2025/ee25btech11010/matgeo
    /5.8.8/figs/speed.png")
plt.show()
```