9.4.40

INDHIRESH S - EE25BTECH11027

1 October, 2025

Question

A train travels 360 km at a uniform speed. If the speed had been 5 km/hr more, it would have taken 1 hour less for the same journey. Find the speed of the train.

Let the uniform speed of the train be s km/hr.
Let the time taken for the journey be t hours.
From 1st journey:

$$360 = s \times t \tag{1}$$

$$t = \frac{360}{s} \tag{2}$$

For the second scenario:

$$360 = (s+5)(t-1) \tag{3}$$

Theoretical Solution

Now substitute Eq.2 in Eq.3

$$360 = (s+5)(\frac{360}{s} - 1) \tag{4}$$

$$s^2 + 5s - 1800 = 0 \tag{5}$$

Let

$$u = s^2 + 5s - 1800 \tag{6}$$

This can be expressed as:

$$\mathbf{x}^\mathbf{T}\mathbf{V}\mathbf{x} + 2\mathbf{u}^\mathbf{T}\mathbf{x} + f = 0 \tag{7}$$

Where,

$$\mathbf{x} = \begin{pmatrix} s \\ u \end{pmatrix}, \mathbf{V} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} 2.5 \\ -0.5 \end{pmatrix} \; and \;\; f = -1800 \tag{8}$$

Theoretical solution

Now finding the point of intersection of parabola with s-axis:

$$\mathbf{x} = \mathbf{h} + k\mathbf{m} \tag{9}$$

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \ \ and \ \ \mathbf{m} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{10}$$

$$\mathbf{x} = k \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{11}$$

Now substitute Eq.11 in Eq.7

$$k^2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} 2.5 \\ -0.5 \end{pmatrix}^T k \begin{pmatrix} 1 \\ 0 \end{pmatrix} - 1800 = 0 \tag{12}$$

$$k^2 + 5k - 1800 = 0 \tag{13}$$

Theoretical Solution

$$k = \frac{-5 \pm \sqrt{25 - 4(-1800)}}{4} \tag{14}$$

$$k = 40 \quad and \quad k = -45 \tag{15}$$

Speed cannot be negative. So,

$$k = 40 \tag{16}$$

Substitute in Eq.11

$$s = 40 \ km/hr \tag{17}$$

C Code

```c
#include <math.h>

int solve_quadratic(double a, double b, double c, double* root1,
    double* root2) {
    if (a == 0) {
        // Not a quadratic equation
        return 0;
    }

    double discriminant = b * b - 4 * a * c;

    if (discriminant < 0) {
        // No real roots
        return 0;
    } else if (discriminant == 0) {
        // One real root
        *root1 = -b / (2 * a);
        return 1;
    }
```

C Code

```c
    else {
      // Two real roots
      double sqrt_discriminant = sqrt(discriminant);
      *root1 = (-b + sqrt_discriminant) / (2 * a);
      *root2 = (-b - sqrt_discriminant) / (2 * a);
      return 2;
    }
}
```

Python Code

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

# --- 1. Load the C library ---
lib_name = 'quad.so'
if platform.system() == 'Windows':
    lib_name = 'quad.dll'

try:
    c_lib = ctypes.CDLL(f'./{lib_name}')
except OSError as e:
    print(fError loading shared library: {e})
    print(fPlease make sure you have compiled solver.c into {lib_name})
    exit()
```

Python Code

```
    # --- 2. Define the C function signature for Python ---
c_lib.solve_quadratic.argtypes = [
    ctypes.c_double, ctypes.c_double, ctypes.c_double,
    ctypes.POINTER(ctypes.c_double), ctypes.POINTER(ctypes.
        c_double)
]
c_lib.solve_quadratic.restype = ctypes.c_int

def solve_with_c(a, b, c):
    A Python wrapper that calls the C function.
    root1 = ctypes.c_double()
    root2 = ctypes.c_double()

    num_roots = c_lib.solve_quadratic(a, b, c, ctypes.byref(root1
        ), ctypes.byref(root2))

    if num_roots == 0: return None
    if num_roots == 1: return (root1.value,)
    return (root1.value, root2.value)
```

Python Code

```python
def plot_solution(a, b, c, roots):
Plots the quadratic function and its intersection points with the x-axis.
s_values = np.linspace(min(roots) - 20, max(roots) + 20, 400)
y_values = a * s_values**2 + b * s_values + c

plt.figure(figsize=(10, 6))

# Plot the parabola
plt.plot(s_values, y_values, label=f'Parabola: $y = s^2 + 5s - 1800$')

# Plot the x-axis for reference
plt.axhline(0, color='black', linestyle='--')
```

Python Code

```python
 # Plot the intersection points (roots)
plt.plot(roots, [0]*len(roots), 'ro', markersize=8, label=f'Intersection Points')
for root in roots:
    plt.text(root, 100, f'({root:.1f}, 0)', ha='center', fontsize=10)
positive_root = next(r for r in roots if r > 0)
plt.title(fSolution to the Train Problem (s = {positive_root:.0f} km/hr))
plt.xlabel(Speed (s))
plt.ylabel(y)
plt.grid(True)
plt.legend()
plt.savefig(/media/indhiresh-s/New Volume/Matrix/ee1030-2025/ee25btech11027/MATGEO/9.4.40/figs/figure1.png)
plt.show()
```

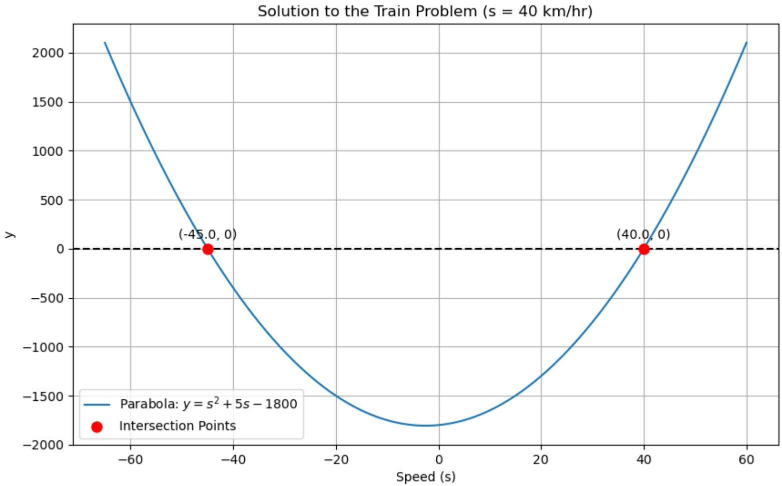
Python Code

```python
    # --- Main execution ---
if __name__ == __main__:
    # Coefficients from the train problem: s^2 + 5s - 1800 = 0
    a, b, c = 1.0, 5.0, -1800.0

    roots = solve_with_c(a, b, c)

    if roots:
        sorted_roots = sorted(roots)
        print(fRoots found via C function: {sorted_roots})
        plot_solution(a, b, c, sorted_roots)
    else:
```

Plot



Solution to the Train Problem (s = 40 km/hr)
(-45.0, 0)
(40.0, 0)
y
Speed (s)
2000
1500
1000
500
0
−500
−1000
−1500
−2000
−60
−40
−20
0
20
40
60
Parabola: $y = s^2 + 5s - 1800$
Intersection Points