

# NCERT-10.4.3.10

1

EE24BTECH11055 - Sai Akhila

## QUESTION:

An express train takes 1 hour less than a passenger train to travel 132 km between Mysore and Bangalore (without taking into consideration the time they stop at intermediate stations.) If the average speed of the express train is 11 km/h more than that of the passenger train, find the average speed of the two trains.

## THEORETICAL SOLUTION:

Let the speed of the express train be  $e$  and the speed of the passenger train be  $p$ . The time taken by the express train and the passenger train to travel from Mysore to Bangalore would respectively be :

$$t_e = \frac{132}{e} \quad (0.1)$$

$$t_p = \frac{132}{p} \quad (0.2)$$

$$(0.3)$$

Given that the time taken by express train is 1 hour less,

$$t_p - t_e = 1 \quad (0.4)$$

$$\Rightarrow \frac{132}{p} - \frac{132}{e} = 1 \quad (0.5)$$

According to the other condition given,

$$e = 11 + p \quad (0.6)$$

$$(0.7)$$

By substituting  $e$  in equation (0.6) in equation (0.5), we get:

$$\frac{132}{p} - \frac{132}{11 + p} = 1 \quad (0.8)$$

$$11 \times 132 = p^2 + 11p \quad (0.9)$$

$$p^2 + 11p - 1452 = 0 \quad (0.10)$$

$$p(p + 44) - 33(p + 44) = 0 \quad (0.11)$$

$$(p + 44)(p - 33) = 0 \quad (0.12)$$

Hence we get  $p$  (the average speed of the passenger train) to be 33 km/h. The average speed of the express train would be  $33 + 11 = 44$  km/h

## 1. NEWTON'S METHOD

Newton's Method is an iterative technique that refines an initial guess ( $p_0$ ) using the formula:

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}. \quad (0.13)$$

### *Advantages*

- Quadratic convergence when the initial guess is close to the root.
- Requires fewer iterations compared to other methods.

### *Disadvantages*

- Requires the computation of the derivative ( $f'(p)$ ).
- May fail to converge if the initial guess is not close to the root.
- Does not guarantee convergence for all functions.

## 2. BISECTION METHOD

The Bisection Method is a bracketing method that requires two initial guesses ( $a$ ) and ( $b$ ) such that ( $f(a) \cdot f(b) < 0$ ). The iterative formula is:

$$p_n = \frac{a + b}{2}, \quad (0.14)$$

where  $[a, b]$  is updated based on the sign of ( $f(p_n)$ ).

### *Advantages*

- Guaranteed convergence if ( $f(a) \cdot f(b) < 0$ ).
- Simple to implement and does not require the derivative of ( $f(p)$ ).

### *Disadvantages*

- Convergence is linear, which can be slow compared to Newton's Method.
- Requires a bracketing interval  $[a, b]$  where the root lies.

## 3. FIXED-POINT ITERATION

Fixed-Point Iteration solves equations by rewriting ( $f(p) = 0$ ) as ( $p = g(p)$ ) and iterating:

$$p_{n+1} = g(p_n). \quad (0.15)$$

### *Advantages*

- Simple to implement and does not require derivatives.
- Useful for equations that can be easily transformed into ( $p = g(p)$ ).

### Disadvantages

- Convergence is not guaranteed unless  $(|g'(p)| < 1)$  near the root.
- May converge slowly compared to other methods.

$$\text{SOLVING } (p^2 + 11p - 1452 = 0)$$

The given equation is:

$$p^2 + 11p - 1452 = 0. \quad (0.16)$$

We rewrite it as:

$$f(p) = p^2 + 11p - 1452. \quad (0.17)$$

### Using Newton's Method

The derivative is:

$$f'(p) = 2p + 11. \quad (0.18)$$

The iteration formula becomes:

$$p_{n+1} = p_n - \frac{p_n^2 + 11p_n - 1452}{2p_n + 11}. \quad (0.19)$$

### Using Bisection Method

Choose  $[a, b] = [30, 40]$ , since  $(f(30) < 0)$  and  $(f(40) > 0)$ . The iterative formula is:

$$p_n = \frac{a + b}{2}. \quad (0.20)$$

Update  $(a)$  or  $(b)$  based on the sign of  $(f(p_n))$ .

### Using Fixed-Point Iteration

Rewrite the equation as:

$$p = g(p) = \sqrt{1452 - 11p}. \quad (0.21)$$

The iteration formula is:

$$p_{n+1} = \sqrt{1452 - 11p_n}. \quad (0.22)$$

Convergence depends on the initial guess and the condition  $(|g'(p)| < 1)$ .

$$5. \text{ IDEAL METHOD FOR } (p^2 + 11p - 1452 = 0)$$

Given the nature of the equation:

- Newton's Method is ideal if a good initial guess is available, as it converges rapidly.
- Bisection Method is robust and guarantees convergence but is slower.
- Fixed-Point Iteration is less reliable and may require careful transformation to ensure convergence.

Newton's Method is recommended for this problem due to its efficiency.

# 1 USING NEWTON'S METHOD ALONG WITH BISECTION METHOD FOR IMPROVED EFFICIENCY

Now, we solve the quadratic equation

$$p^2 + 11p - 1452 = 0 \quad (0.23)$$

using a combination of Newton's method and the bisection method. While Newton's method is efficient and converges rapidly when close to the root, it can fail if the initial guess is far from the root or if the derivative is zero. The bisection method is slower but more robust, ensuring convergence as long as the function changes sign over an interval. By combining both methods, we achieve the robustness of the bisection method and the speed of Newton's method.

## 1.1 Combining the Methods

To combine these methods, we proceed as follows:

- 1) Use the bisection method to narrow the initial interval where the root is likely to be located.
- 2) Once the interval is sufficiently small, apply Newton's method to converge more rapidly to the root.
- 3) If Newton's method diverges or does not converge, fall back to the bisection method to refine the root estimate.

This strategy ensures that the method remains robust and guarantees convergence, while taking advantage of the faster convergence rate of Newton's method.

## 1.2 Results

We applied the combined method to the quadratic equation  $p^2 + 11p - 1452 = 0$  with an initial interval  $[-50, 33]$ . The method successfully converged to the root, demonstrating the effectiveness of combining the robustness of the bisection method with the fast convergence of Newton's method.

# 2 SOLVING USING EIGEN VALUES:

Alternatively, we can solve the question by using the eigen values of the companion matrix.

For a polynomial equation of form  $x^n + c_{n-1}x^{n-1} + \dots + c_2x^2 + c_1x + c_0 = 0$  the companion matrix is of the form:

$$\begin{pmatrix} 0 & 0 & \dots & 0 & -c_0 \\ 1 & 0 & \dots & 0 & -c_1 \\ 0 & 1 & \dots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -c_{n-1} \end{pmatrix} \quad (3.1)$$

The eigen values of this matrix are the roots of the polynomial equation. For this question,

$$n = 2 \quad (3.2)$$

$$c_0 = -1452 \quad (3.3)$$

$$c_1 = 11 \quad (3.4)$$

$$C = \begin{pmatrix} 0 & 1452 \\ 1 & -11 \end{pmatrix} \quad (3.5)$$

The QR algorithm is an iterative method used to find the eigenvalues of a square matrix. It is based on decomposing a matrix into its QR decomposition (where Q is an orthogonal matrix and R is an upper triangular matrix), and then iteratively improving the approximation of the eigenvalues.

### 2.1 The Gram-Schmidt orthogonalization

Consider a matrix A. It can be represented as

$$A = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_3]$$

$$\tilde{q}_1 = \mathbf{a}_1, \mathbf{q}_1 = \frac{\tilde{q}_1}{\|\tilde{q}_1\|}$$

$$\tilde{q}_2 = \mathbf{a}_2 - (a_2^T \mathbf{q}_1) \mathbf{q}_1$$

$$\mathbf{q}_2 = \frac{\tilde{q}_2}{\|\tilde{q}_2\|}$$

**Generalising the algorithm:**

$$\text{Initialise } \tilde{q}_1 = \mathbf{a}_1, \mathbf{q}_1 = \frac{\tilde{q}_1}{\|\tilde{q}_1\|}$$

for r=2 to k:

$$\tilde{q}_r = \mathbf{a}_r - \sum_{i=1}^{r-1} (a_r^T \mathbf{q}_i) \mathbf{q}_i$$

$$\mathbf{q}_r = \frac{\tilde{q}_r}{\|\tilde{q}_r\|}$$

Hance, we can write any matrix A as:

$$A = QR$$

where Q is the orthogonal matrix that consists of orthogonal basis of A, and image space of A = image space of Q. R is an upper triangular matrix, whose diagonal entries are the norms of the respective column vectors and the upper diagonal entries are dots products of the corresponding vectors with the orthogonal vectors in Q (in the indices less than the corresponding index).

### 2.2 Hessenburg reduction:

To reduce a matrix A to Hessenberg form, the Householder transformations are commonly used. The process works by successively applying orthogonal transformations to eliminate the elements below the first sub-diagonal.

Householder transformations are used to create zeros below the diagonal and the first subdiagonal of the matrix. The goal is to zero out the entries  $A_{i,j}$  for  $i > j + 1$ .

The Householder transformation is a reflection operation that transforms the given matrix by applying orthogonal matrices to it. A Householder matrix H is constructed as:

$$H = I - 2 \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

where  $\mathbf{v}$  is a vector that reflects the given column vector  $\mathbf{x}$  (the portion of the matrix you want to zero out) to align with a scalar multiple of a unit vector. This is done repeatedly for each column, creating the Hessenberg form.

Repeat for all columns.

After applying the transformations, the result will be an upper Hessenberg matrix where all elements below the first sub-diagonal are zero.

### 2.3 Iterating the QR decomposition:

- 1) Perform QR decomposition  $A_k = Q_k R_k$
- 2) Compute the next matrix  $A_{k+1} = R_k Q_k$
- 3) Repeat the process until  $A_k$  converges to a nearly upper triangular matrix
- 4) The diagonal entries of the converged matrix are the eigenvalues of A.

### 3 TIME COMPLEXITY:

The time complexity of the QR algorithm primarily depends on two factors:

- 1) The number of iterations required for convergence
- 2) The computational cost of each iteration

The total time complexity of the QR algorithm is the product of the cost of one QR decomposition and the number of iterations:

- 1) **Per iteration:** QR decomposition costs  $O(n^3)$  (using Gram-Schmidt algorithm)
- 2) **Number of iterations:** The algorithm typically converges in  $O(n)$  iterations.

Thus, the overall time complexity of the QR algorithm is:

$$O(n^3) \times O(n) = O(n^4)$$

For optimized version (**Hessenberg reduction**): The matrix is often reduced to Hessenberg form (H) using a pre-processing step that costs  $O(n^3)$ .

This reduces the cost of the QR decomposition to  $O(n^2)$  for Hessenberg matrices.

Time complexity with Hessenberg reduction:  $O(n^3)$  for reduction +  $O(n^2)$  per iteration

⇒ Overall complexity:  $O(n^3)$

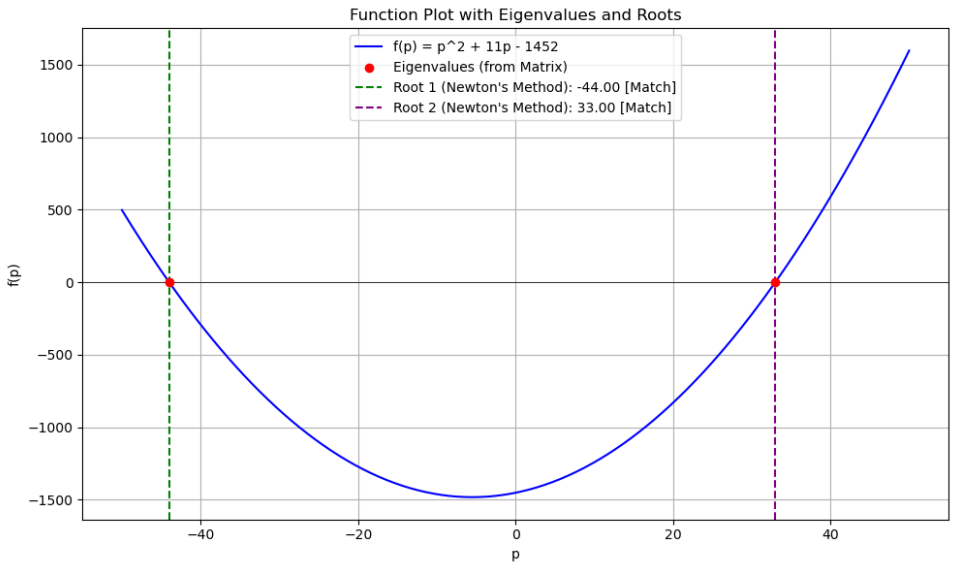


Fig. 2.1: Eigenvalues and Roots Highlighted