**ALGORITHM 5.1**

**Bisection method** To determine a root of $f(x) = 0$ in the interval $[x_1, x_2]$.

**Input**: The function $f(x) = 0$, error limit of the root $\epsilon$, and the interval $[x_1, x_2]$, where $f(x_1)$ and $f(x_2)$ have opposite sign.

**Output**: Root $z$.

Step 1: [**Input**] Read $x_1, x_2, \epsilon$.

Step 2: [**Compute midpoint of the interval**] $z = \dfrac{x_1 + x_2}{2}$.

Step 3: [**Squeeze interval**] If $f(z)f(x_1) > 0$ then $x_1 = z$ else $x_2 = z$.

Step 4: [**Looping for refinement**] Repeat Steps 2 to 3 until $|x_1 - x_2| < \epsilon$.

Step 5: [**Output**] Print the root $z = \dfrac{x_1 + x_2}{2}$.

Step 6: [**Terminate the algorithm**] Stop.

The final value $(x_1 + x_2)/2$ is usually accepted as the approximation to the root.

**Note:** If the given interval $[x_1, x_2]$ does not obey the condition that $f(x_1)$ and $f(x_2)$ have opposite sign, then this algorithm fails to find a root.

LE 5.1 ... $1.0 \leq x \leq 2$ by the

ALGORITHM 5.2

**Regula falsi method** To find a root of $f(x) = 0$ in the interval $[x_1, x_2]$.
**Input:** The function $f(x) = 0$, error limit of the root $\epsilon$, the interval $[x_1, x_2]$, and the maximum number of iteration $n$ where $f(x_1)$ and $f(x_2)$ have opposite sign.
**Output:** Root $z$.

Step 1: [**Input**] Read $x_1, x_2, \epsilon, n$.
Step 2: [**Initialization**] $z = x_1$, count=1.
Step 3: [**Compute the point of intersection**]

$$x_3 = x_2 \frac{f(x_1)}{f(x_1) - f(x_2)} + x_1 \frac{f(x_2)}{f(x_2) - f(x_1)}$$

Step 4: [**Squeeze the interval**] If $f(x_1)f(x_3) > 0$ then $x_1 = x_3$ else $x_2 = x_3$.
Step 5: [**Increment iteration count**] count = count + 1.
Step 6: [**Check iteration limit**] If count $> n$ then
Print "Does not converge after $n$ iterations to the desired limit $\epsilon$"
Goto Step 9.
Step 7: [**Looping for error refinement**] Repeat Steps 3 to 6 until $|f(x_3)| < \epsilon$.
Step 8: [**Root**] $z = x_3$.
Step 9: [**Output**] Print the roots $z$.
Step 10: [**Terminate the algorithm**] Stop.

**Note:** The method of false position divides the interval, at every stage, unequally by approximating the behaviour of the function $f(x)$ in the interval uniformly by a straight line. When this approximation correctly depicts the behaviour of the function within the interval, this method may be expected to give the root with desired accuracy faster than the bisection method. On the other hand, for each computation of $x_3$, the bisection method requires fewer arithmetic operations (one division and one addition) than the method of false position (two addition/subtraction and four multiplication/division). So, except in a very few special cases, the bisection method is superior. The method of false position should be used only when the behaviour of $f(x)$ justifies approximation by a straight line in the interval.

LGORITHM 5.6

**Newton–Raphson method** To determine a root of $f(x) = 0$ with initial value of $x$ as $x_0$.

**Input**: The function $f(x) = 0$, with initial root $x_0$ and two error limits $\epsilon_1$ and $\epsilon_2$.

**Output**: Root $z$.

Step 1: [**Input**] Read $x_0$, $\epsilon_1$, and $\epsilon_2$.

Step 2: Compute $f(x_0)$ and $f'(x_0)$.

Step 3: Set $x_1 = x_0$.

Step 4: If $(f(x_0) \neq 0)$ and $(f'(x_0) \neq 0)$

[**Looping for refinement**]

Repeat

Set $x_0 = x_1$

Set $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

until $|x_0 - x_1| < \epsilon_1$ or $|f(x_1)| < \epsilon_2$.

Step 5: [**Root**] $z = x_1$.

Step 6: [**Output**] Print the root $z$.

Step 7: [**Terminate the algorithm**] Stop.