CHAPTER 4 NONLINEAR EQUATIONS

SOLVED EXAMPLE PROBLEMS

for

NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS With Pseudocodes

By Zekeriya ALTAÇ

November 2024

EXAMPLE 4.1: Implementing of Bisection Method

The discharge of charge of an RLC circuit, Q, is given by the following relation:

$$Q(t) = Q_0\, e^{-Rt/2L} \cos\left(t\sqrt{\frac{1}{LC} - \frac{R^2}{4L^2}}\right)$$

where R is the resistor (Ω), L is the inductor (H), C is the capacitor (F), and Q_0 is the initial charge. L and C are given as 4 H and 3×10^{-4} F, respectively. Using the *Bisection method* with the tolerance of $\varepsilon = 10^{-3}$, calculate the resistor R when the charge is reduced to 10% of its initial charge (i.e., $Q/Q_0 = 0.1$) in 0.1 seconds. The starting search interval for the resistor is given as [200, 300] Ω.

SOLUTION:

We seek the resistor R in the case the charge is reduced to 10% of its initial charge (i.e., $Q/Q_0 = 0.1$) in 0.1 seconds for the circuit with $L = 4$ H and $C = 3 \times 10^{-4}$ F. First, we rearrange the above equations and define a function of R as

$$f(R) = \frac{Q(t)}{Q_0} - e^{-Rt/2L} \cos\left(t\sqrt{\frac{1}{LC} - \frac{R^2}{4L^2}}\right) = 0$$

and substituting the given data into the above equation gives

$$f(R) = 0.1 - e^{-0.0125R} \cos\left(0.1\sqrt{\frac{2500}{3} - \frac{R^2}{64}}\right) = 0$$

We note that $f(R_a) > 0$ and $f(R_b) < 0$ for the search interval $(R_a, R_b) = (200, 300)$ Ω. This guarantees, by the Intermediate Value Theorem, that there exists at least one root in the prescribed interval.

We begin the bisection procedure by first evaluating the function at the midpoint. $R_m = (R_a + R_b)/2$; i.e., $f(R_m)$ is calculated. If $f(R_m) = 0$, then R_m is the *true root*, and the process terminates. If $f(R_m)$ is sufficiently close to zero (i.e., $|f(R_m)| < \varepsilon$), then $R = R_m$ can be accepted as the approximation of the root, and the process can be terminated. If $f(R_m)$ and $f(R_a)$ have opposite signs (i.e., $f(R_a)f(R_m) < 0$), then the root lies in the interval $[R_a, R_m]$, so the search interval is narrowed down as $[R_a, R_b = R_m]$. Otherwise, $f(R_m)$ and $f(R_a)$ have the same signs (i.e., $f(R_a)f(R_m) > 0$). In this case, the root lies in the interval $[R_m, R_b]$, so the search interval is updated as $[R_a = R_m, R_b]$. This procedure is repeated for the current interval until the search interval is sufficiently small (i.e., $|R_a - R_b| < \varepsilon$), or the function value at the midpoint is close enough to zero $|f(R_m)| < \varepsilon$.

Table 4.1: The computation history for the bisection method.

p	$R_a^{(p)}$	$R_b^{(p)}$	$f(R_a^{(p)})$	$f(R_b^{(p)})$	$R_m^{(p)}$	$\lvert f(R_m^{(p)})\rvert$
0	200	300	0.08960	−0.02986	250	0.020658
1	250	300	0.02066	−0.02986	275	0.006384
2	250	275	0.02066	−0.00638	262.50	0.006640
3	262.5	275	0.00664	−0.00638	**268.75**	$\mathbf{1.03 \times 10^{-5}}$

Notice that the term $(1/LC - R^2/4L^2)$, argument of the cosine function, becomes negative for $R > 2\sqrt{L/C} = 230.94\,\Omega$, in which case the cosine term is replaced with $\cosh(t\sqrt{R^2/4L^2 - 1/LC})$.

The result of the bisection procedure is summarized in Table 4.1. Note that at the 3rd bisection, we end up with $|f(R_m^{(3)})| = 1.03 \times 10^{-5} < 10^{-3}$ for $R = 268.75$ Ω. It turns out that this value is in fact very close to the true value of $R = 268.75989$, yielding the true absolute error of $|R_a^{(3)} - R| = 0.00989$.

In this problem, the $|f(R_m)| < \varepsilon$ criterion was adopted as the stopping criterion because the resistance value of any resistor is not absolute. In other words, a resistor is manufactured to meet certain tolerance requirements due to a variety of factors expected during production. This variability is accounted for by the tolerance rating (e.g., $\pm 1\%$, $\pm 5\%$, $\pm 0.1\%$), which indicates how much the actual resistance of a resistor can differ from its nominal value. In this context, there is no point in trying to determine the resistance value with high accuracy by also enforcing the $|R_a - R_b| < \varepsilon$ criterion.

Discussion: The bisection method is a numerical technique used to find roots of continuous functions. It is particularly useful when the function is well-behaved over a specified interval. The method can be reviewed under five headings:

1) Guaranteed Convergence: Convergence of the bisection method is guaranteed as long as the nonlinear function, expressed as $f(x) = 0$, is continuous on $[a, b]$, and $f(a)$ and $f(b)$ are of opposite signs. In other words, if a root lies within the starting search interval, the method will always converge to it. This property makes it highly reliable for solving nonlinear equations. As the method is guaranteed to converge, its convergence can be relatively slow compared to other methods. The convergence of the bisection method is *linear* (with a steady convergence rate of 0.5); this is because the algorithm systematically narrows down the search interval by halving it at each step.

2) Simplicity: The bisection algorithm is straightforward to implement, involving only function evaluations and simple arithmetic (bisecting intervals). Unlike some methods, it does not require the derivative of the function, which is desired in the case of non-differentiable functions or functions that are difficult or too complicated to differentiate.

3) Robustness: The method is not sensitive to a poor initial guess compared to other root-finding methods. Also, computational errors (e.g., rounding errors) do not cause numerical issues that would prevent convergence.

4) Bounded Intervals: The method depends on the selection of the starting search interval $[a, b]$. It is impossible for the current estimate $x^{(p+1}$ to jump outside of the interval and to diverge. The method can only find one root in the specified interval, even if multiple roots exist within that interval. To find multiple roots, the search intervals should be refined before applying the bisection method.

5) Error Estimation: The error is directly tied to the interval size, making it straightforward to determine how many bisections are needed to achieve a desired level of accuracy; that is, after p steps, the error can be estimated by $(b-a)/2^{p+1}$. For this reason, starting with a tight initial search interval could considerably reduce the computational efforts.

EXAMPLE 4.2: Implementing the Method of False Position

Heat generation resulting from exothermic reactions occurring in a chemical reactor is described by the Arrhenius reaction rate, $q_{\text{gen}} = C\exp(-E_a/RT)$, where C is a reaction-dependent constant, $R = 8.314$ J/mol-K is the universal gas constant, E_a is the activation energy, and T is the reactor temperature in Kelvin. In steady operation, where the heat losses from the reactor occur by both convection and radiation, the conservation of energy can be expressed as

$$\underset{\text{heat generation}}{C\ \exp(-E_a/RT)} = \underset{\text{convection heat loss}}{hA(T-T_\infty)} + \underset{\text{radiation heat loss}}{\varepsilon\sigma A(T^4-T_\infty^4)}$$

where A is the surface area of the reactor, h is the convection heat transfer coefficient, T_∞ is the ambient temperature, T is the mean reactor temperature, ε is the surface emissivity of the reactor, and $\sigma = 5.67\times 10^{-8}$ W/m^{2}K^4 is the Stefan-Boltzmann constant. Using the *method of false position* and given data, find the mean reactor temperature *accurate to one decimal place*, assuming the generated heat is lost (a) by convection only, (b) by radiation only, and (c) by both convection and radiation. *Given*: $E_a = 18.5$ kJ/mol, $C = 3.8$ MW, $T_\infty = 290$ K, $h = 35$, W/m^2·K, $A = 5.85$, m^2, and $\varepsilon = 0.9$.

SOLUTION:

(a) In case of heat loss only by convection, we neglect the 'radiation heat loss' term in the conservation of energy. Substituting the given data and rearranging it into an equation with the zero right-hand side, we obtain:

$$f(T) = 3.8\times 10^6\, e^{-2225.16/T} - 204.75\,(T-290) = 0$$

which is a non-linear equation whose roots cannot be determined using analytical means.

The method of false position is another bracketing method, requiring a starting interval that contains a root. Thus, we need to do some preliminary work to find the starting search interval $[a, b]$ such that $f(a)f(b) < 0$. For part (a), we start with [400, 550] as $f(400) < 0$ and $f(550) > 0$ as well as the convergence tolerance (for *one decimal place accuracy*) as $\epsilon = 0.5\times 10^{-1}$. The results of the iteration steps are presented in Table 4.2.

Table 4.2: The computation history for Part (a).

p	$a^{(p)}$	$b^{(p)}$	$f(a^{(p)})$	$f(b^{(p)})$	$T^{(p)}$	$f(T^{(p)})$	
0	400	550	−7939.5	13252.3	456.197	−5091.6	
1	456.197	550	−5091.6	13252.3	482.234	−1704.3	
2	482.234	550	−1704.3	13252.3	489.956	−444.92	
3	489.956	550	−444.92	13252.3	491.906	−108.44	
4	491.906	550	−108.44	13252.3	492.378	−25.986	
5	492.378	550	−25.986	13252.3	492.491	−6.2015	
6	492.491	550	−6.2015	13252.3	492.517	−1.4785	
7	492.517	550	−1.4785	13252.3	492.524	−0.35243	
8	492.524	550	−0.35243	13252.3	492.525	−0.08400	
9	492.525	550	−0.08400	13252.3	**492.526**	**−0.020021**	$< \epsilon$

The procedure converges to an estimate of 492.526 in 9 steps to within two digits of accuracy; that is, the $T^{(7)}$, $T^{(8)}$, and $T^{(9)}$ values are accurate to two decimal places. Note that during bracketing, the search interval remains anchored on the right end ($b^{(p)}$) while it is narrowed down from the left side. Unlike the bisection method, where the midpoint of the interval is used, the new estimates $T^{(p)}$ are obtained using Eq. (4.4), which is derived from linear interpolation.

(b) In case of heat losses only by radiation, we will neglect the 'convection heat loss' term in the conservation of energy. Likewise, substituting the given data and rearranging it, we find

$$f(T) = 3.8 \times 10^6\, \mathrm{e}^{-2225.16/T} - 2.9853 \times 10^{-7}(T^4 - 290^4) = 0$$

which is also a non-linear equation.

After a few trials, we set the initial search interval to [1000, 1300] due to $f(1000) > 0$ and $f(1300) < 0$. The results of this procedure are tabulated in Table 4.3. The method converges to an estimate of 1179.864 with three-digit accuracy in 10 steps, i.e., $|T^{(10)} - T^{(9)}| = 0.13 \times 10^{-3} < 0.5 \times 10^{-3}$). Similarly, the search interval is anchored at the right end, $b^{(p)}$.

Table 4.3: The computation history for Part (b).

p	$a^{(p)}$	$b^{(p)}$	$f(a^{(p)})$	$f(b^{(p)})$	$T^{(p)}$	$f(T^{(p)})$	
0	1000.	1300	114172.	−164371	1122.967	51247.6	
1	1122.97	1300	51247.6	−164371	1165.043	14859.9	
2	1165.04	1300	14859.9	−164371	1176.233	3742.63	
3	1176.23	1300	3742.63	−164371	1178.988	908.632	
4	1178.99	1300	908.632	−164371	1179.653	218.621	
5	1179.65	1300	218.621	−164371	1179.813	52.4873	
6	1179.81	1300	52.4873	−164371	1179.852	12.5948	
7	1179.85	1300	12.5948	−164371	1179.861	3.02184	
8	1179.86	1300	3.02184	−164371	1179.863	0.72500	
9	1179.86	1300	0.725003	−164371	1179.864	0.17394	
10	1179.86	1300	0.173942	−164371	**1179.864**	**0.04173**	$< \epsilon$

(c) In case of heat losses by both convection and radiation, employing the same procedure, we obtain the following nonlinear equation:

$$f(T) = 3.8 \times 10^6\, \mathrm{e}^{-2225.16/T} - 204.75(T - 290) - 2.9853 \times 10^{-7}(T^4 - 290^4) = 0$$

The iteration history obtained with the starting search interval chosen as [600, 800] ($f(1000) < 0$ and $f(1300) > 0$) is presented in Table 4.4. It is observed that the search interval is anchored at the left side. The convergence is very fast because the nonlinear function depicts a change that is almost linear.

Table 4.4: The computation history for Part (c).

p	$a^{(p)}$	$b^{(p)}$	$f(a^{(p)})$	$f(b^{(p)})$	$T^{(p)}$	$f(T^{(p)})$	
0	600	800.	−6905.64	10815.8	677.935	2288.17	
1	600	677.935	−6905.64	2288.17	658.539	25.9684	
2	600	658.539	−6905.64	25.9684	658.319	−0.0833	
3	658.319	658.539	−0.0832863	25.9684	**658.320**	**1.23×10^{-5}**	$< \epsilon$

Discussion: The method of false position is an iterative root-finding method that combines aspects of the bisection method and the secant method. It uses a linear interpolation between two points to approximate the root of a function, adjusting one of the points after each iteration. The method converges linearly, meaning the error decreases at a rate proportional to the previous error.

The method will approach the root if the function behaves well and the initial interval is not poorly chosen. In other words, the convergence rate of the method depends on the function being solved. It can converge very quickly under certain conditions, but this rapid convergence is not guaranteed in all cases. There are the conditions under which the *Method of False Position* tends to converge faster.

1) Function with a single root that behaves nearly linearly: If the nonlinear function $f(x)$ behaves nearly linearly near the root, the method tends to converge rapidly. In this case, the linear interpolation step closely approximates the actual root, and each iteration brings the current estimate closer to the root very quickly. For example, in Figure 4.1, the variation of $f(T)$ of Part (c) on $600 \leqslant T \leqslant 800$ is depicted. The root (≈658.32 K) is inside the interval $600 \leqslant T < 750$, where $f(T)$ behaves almost linearly. That is why the root converged in 3 steps.

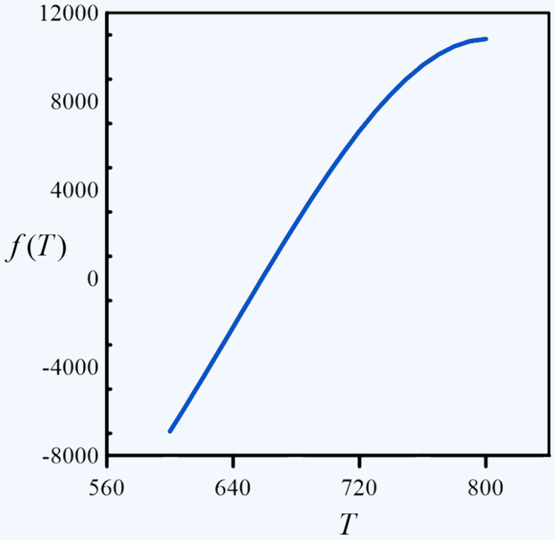

Figure 4.1: The variation of $f(T)$ in Part (c) on $600 \leqslant T \leqslant 800$.

2) The starting search interval is close to the root: When the starting search interval is already very close to the root, the method tends to converge more quickly. The error between successive estimates is small, and the linear interpolation quickly moves the estimate closer to the root.

3) Function is smooth and monotonic in the interval: If the function is smooth and monotonic in the search interval, the method often converges quickly, as the approximation to the root using the secant line is likely to be very accurate.

4) The root is near one of the search interval endpoints: The method can sometimes converge faster if the root is closer to one of the endpoints of the starting search interval. In such cases, the method will adjust one of the interval endpoints faster, leading to a quicker approach to the root.

5) Function has a well-behaved derivative: The method often works better if the function's derivative is continuous and does not change too abruptly. A smooth, well-behaved function allows the method to take advantage of the secant interpolation, making each step more accurate. The method tends to converge very fast when the function has a large slope near the root, ensuring that the secant line intersects the x-axis quickly. However, if the root is near a region where the slope of the function is very small, the method may stall or yield slow convergence.

Compared to the bisection method, the method of false position can outperform it, especially in cases mentioned above.

EXAMPLE 4.3: Rootfinding with the Newton-Rapson Method

The pressure (P), volume (V), and temperature (T) of a real gas are related to each other through the van der Waals equation:

$$\left(P+\frac{n^2a}{V^2}\right)(V-nb)=nRT$$

where $R=0.08206$ atm-L/mol-K is the gas constant, a and b are the real gas constants, and n is the molarity of the gas. 3 mol of BCl_3 (Boron trichloride gas) yields a pressure of 1.15 atm at 310 K. Find the volume of BCl_3 using the *Newton-Raphson method* and compare it with the value obtained from the ideal gas law. Use $\varepsilon=10^{-3}$ as convergence tolerance and $V^{(0)}=20$ L as the initial guess. *Given*: $a=15.39$ atm·L^2/mol^2 and $b=0.1222$ L/mol.

SOLUTION:

We rearrange the Van der Waals equation as follows:

$$f(V)=\left(P+\frac{n^2a}{V^2}\right)(V-nb)-nRT=0$$

Substituting the numerical values, we write $f(V)$ and its derivative df/dV as

$$f(V)=(V-0.3666)\left(1.15+\frac{138.51}{V^2}\right)-76.3158=0,\qquad \frac{df}{dV}=1.15+\frac{101.556-138.51V}{V^3}$$

The volume of BCl_3 with the *ideal gas* assumption yields

$$V=\frac{nRT}{P}=\frac{(3\,\text{mol})(0.08206\,\text{atm}\cdot\text{L/mol}\cdot\text{K})(310\,\text{K})}{(1.15\,\text{atm})}=\mathbf{66.362}\,\text{L}$$

As a real gas, the volume will be estimated using the Newton-Raphson iteration equation expressed as follows:

$$V^{(p+1)}=V^{(p)}-f(V^{(p)})/\frac{df}{dV}(V^{(p)})$$

where p, as usual, denotes the iteration step.

Table 4.5: The computation history for the Newton-Raphson solution.

p	$V^{(p)}$	$f(V^{(p)})$	$f'(V^{(p)})$	$\lvert V^{(p+1)}-V^{(p)}\rvert$
0	20	-46.939	0.816419	57.4935
1	77.4935	14.1591	1.12715	12.5618
2	64.9317	0.05521	1.11752	0.04940
3	**64.8823**	$\mathbf{1.215\times10^{-6}}$	1.11747	$\mathbf{1.087\times10^{-6}}<\epsilon$

Starting with $V^{(0)}=20$ guess, the Newton-Raphson method converges to $V=64.8823$ L within the specified tolerance with only three iterations, i.e., in fact better than the desired (specified) tolerance: $|f(V^{(3)})|=1.215\times10^{-6}<10^{-3}$ and $|V^{(p+1)}-V^{(p)}|=1.087\times10^{-6}<10^{-3}$. The iteration history is presented in Table 4.7. The difference between volumes estimates calculated under *real* and *ideal gas* conditions is $\Delta V=64.882-66.362=-1.48$ L; in other words, the volume as the real gas is 2.3% lower than that of the volume computed by the ideal gas assumption.

Discussion: The Newton-Raphson method is a powerful and widely used iterative technique for finding approximate solutions to equations of the form $f(x)=0$. It is the foundation of optimization techniques such as Newton's method for finding critical points of a function. The benefits of the method are:

1) Fast Convergence: The method is known for its rapid convergence when the function behaves well and the initial guess is close to the root. It has *quadratic convergence* property, meaning the number of correct digits approximately doubles with each iteration near the root.

2) Efficiency: Typically, fewer iterations are required compared to methods such as the bisection or secant methods, making it computationally efficient for systems with high precision requirements.

3) Applicability Fields: The method can be applied to a broad range of non-linear equations, provided that $f(x)$ is differentiable and suitable initial guess is available. The method can be easily extended to multivariate problems.

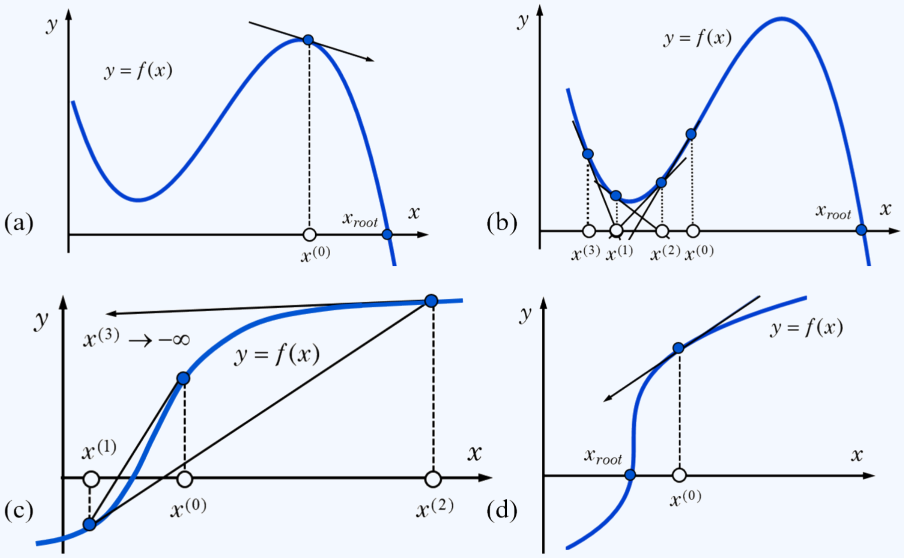

Figure 4.2: Cases where the Newton-Raphson method fails.

The Newton-Raphson method, however, can fail or exhibit undesirably under certain conditions.

1) Poor Initial Guess: In most cases, the convergence of the method depends on the choice of $x^{(0)}$. If an initial guess $x^{(0)}$ (or any $x^{(p)}$) happens to correspond to or land near a local maximum or local minimum point where $f'(x^{(p)}) \to 0$, then the correction term, $\delta^{(p)}$, will tend to be very large, which will result in throwing $x^{(p+1)}$ away from the root (*see* Fig. 4.2a). An initial guess may lead to estimates that cycle back and forth (oscillate) between the sides of the nearest local maximum or local minimum (*see* Fig. 4.2b). An initial guess made to a root near an inflection point, the successive estimates may also tend to diverge from the true root (*see* Fig. 4.2c). However, in most cases, it will be sufficient to change the initial guess to ensure a speedy convergence to a root.

2) Nature of $f(x)$: The convergence of the method also depends on the nature of the nonlinear equation $f(x)$ and its derivative $f'(x)$. In cases of $f(x)$ with nearly horizontal slopes (i.e., very small $f'(x)$), the method may produce very large corrections, $\delta^{(p)}$, leading to slow convergence or divergence. If $f(x)$ or $f'(x)$ is discontinuous at any point, the method can also fail since it relies on the continuity and smoothness of $f(x)$ and $f'(x)$. If $f(x)$ is not continuously differentiable in the vicinity of a root or has a vertical tangent line, the method will fail or diverge (*see* Fig. 4.2d). Functions with multiple roots or roots with multiplicity $m > 1$ slow down convergence because $f'(x^{(p)}) \to 0$ near the root, which makes the method prone to failure. Likewise, when an $f(x)$ has two or more roots that are very close, say within the interval of uncertainty, they are considered to have multiple roots and cannot be effectively isolated. In some cases, especially for functions that depict sharp changes or oscillations, the method can exhibit chaotic behavior, where successive approximations behave unpredictably.

EXAMPLE 4.4: Implementing Fixed Point Iteration Method

Use the Fixed Point Iteration (FPI) to estimate the root of the following nonlinear equation in the interval $[0, \pi/3]$ to an accuracy of $\varepsilon = 5\times 10^{-4}$.

$$\tan\left(\frac{\pi}{5}-x\right)+\tan\left(\frac{x}{5}\right)\tan\left(x+\frac{\pi}{10}\right)-x=0$$

SOLUTION:

The easiest way to define the iteration function is to isolate x as follows:

$$x=g(x)=\tan\left(\frac{\pi}{5}-x\right)+\tan\left(\frac{x}{5}\right)\tan\left(x+\frac{\pi}{10}\right)$$

and express the iteration equation as

$$x^{(p+1)}=g(x^{(p)})=\tan\left(\frac{\pi}{5}-x^{(p)}\right)+\tan\left(\frac{x^{(p)}}{5}\right)\tan\left(x^{(p)}+\frac{\pi}{10}\right)$$

Starting with $x^{(0)}=0.5$ (roughly the midpoint of the interval), the first iteration gives

$$x^{(1)}=g(0.5)=\tan\left(\frac{\pi}{5}-0.5\right)+\tan\left(\frac{05}{5}\right)\tan\left(0.5+\frac{\pi}{10}\right)=0.235306$$

and

$$e^{(0)}=|x^{(1)}-x^{(0)}|=0.264694$$

The second iteration yields

$$x^{(2)}=g(x^{(1)})=g(0.235306)=0.443421 \quad \text{and} \quad e^{(1)}=|x^{(2)}-x^{(1)}|=0.208115$$

The subsequent iterations are performed out in the same manner, and the results obtained are presented in Table 4.6, where $e^{(p)}=|x^{(p+1)}-x^{(p)}|$ is the absolute error, the ratio $e^{(p+1)}/e^{(p)}$ approximates the convergence rate, and $|g'(x^{(p)})|$ denotes the theoretical convergence rate. It is seen that the method converges to an approximate solution in 32 iterations.

Table 4.6: The computation history for the Newton Raphson solution.

p	$x^{(p)}$	$g(x^{(p)})$	$x^{(p+1)})$	$e^{(p)}$	$e^{(p+1)}/e^{(p)}$	$\lvert g'(x^{(p)})\rvert$
0	0.500000	0.235306	0.235306	0.264694		0.589757
1	0.235306	0.443421	0.443421	0.208115	0.786246	0.984375
2	0.443421	0.271137	0.271137	0.172284	0.827831	0.675848
3	0.271137	0.409164	0.409164	0.138027	0.801159	0.928202
4	0.409164	0.295149	0.295149	0.114014	0.826032	0.725871
5	0.295149	0.387314	0.387314	0.092165	0.808361	0.891816
6	0.387314	0.311354	0.311354	0.075960	0.824175	0.757414
7	0.311354	0.373058	0.373058	0.061704	0.812322	0.867707
8	0.373058	0.322298	0.322298	0.050760	0.822639	0.777951
9	0.322298	0.363650	0.363650	0.041352	0.814658	0.851589
10	0.363650	0.329681	0.329681	0.033969	0.821466	0.791514
⋮	⋮	⋮	⋮	⋮	⋮	⋮
30	0.345173	0.344552	0.344552	0.000621	0.818683	0.818233
31	0.344552	0.345061	0.345061	0.000508	0.818765	0.819133
32	0.345061	0.344644	**0.344644**	**0.000416**$<\varepsilon$	0.818670	0.818396

Discussion: The success of the fixed point iteration method depends on several critical factors, which determine whether the method converges to a solution and how efficiently it does so.

1) The proper choice of iteration function $g(x)$: This is perhaps the most crucial aspect of the method, since the goal is to transform $f(x) = 0$ into $x = g(x)$, which can be done in a variety of ways. If $g(x)$ is differentiable on $[a, b]$ and $|g'(x^{(p)})| < 1$ for all $x \in [a, b]$, the fixed point iteration will converge to the fixed point α, starting from any initial guess $x^{(0)} \in [a, b]$. Thus, it is important to choose $g(x)$ that avoids oscillatory or divergent behavior near the fixed point.

2) Initial guess $x^{(0)}$: The initial guess $x^{(0)}$ should be close enough to the root for the iteration to converge. If $x^{(0)}$ is too far from the root, the method might diverge if $|g'(x)| \geqslant 1$ at or near $^{(0)}$, or it might cycle or converge to an unintended fixed point.

3) Continuity of $g(x)$: The iteration function $g(x)$ should be continuous in the interval where the fixed point lies. Discontinuities or sharp changes in $g(x)$ can lead to divergence or slow convergence.

4) Numerical errors: Rounding errors and truncation errors in numerical computations can impact convergence, especially for poorly conditioned $g(x)$. A well-chosen $g(x)$, minimizes these issues by reducing oscillations or instability. Working in high precision can resolve slow converge or divergence problems in ill-conditioned cases.

5) 'Rate of convergence' and 'order of convergence': The convergence of this method depends on $|g'(x)|$, i.e., the choice of $g(x)$. For the method to converge to a fixed point on the interval $[a, b]$, the sufficient condition is that $|g'(x)| < 1$ for all x. If $|g'(\alpha)| \neq 0$ but $|g'(\alpha)| < 1$, the convergence is *linear*, meaning that the true error $|x^{(p+1)} - \alpha|$ decreases at a constant rate proportional to $|g'(\alpha)|$. If $|g'(\alpha)| = 0$ but $|g''(\alpha)| \neq 0$, the convergence is *quadratic* (*order of convergence*), meaning that the error decreases much faster, i.e., $|x^{(p+1)} - !\alpha| \propto |x^{(p)} - \alpha|^2$.

For a sequence that converges with order n, we may write

$$\lim_{p\to\infty} \frac{|x^{(p+1)} - \alpha|}{|x^{(p)} - \alpha|^n} = K \tag{A}$$

Since α is unknown, for sufficiently large p, we may replace Eq. (A) with

$$\lim_{p\to\infty} \frac{|x^{(p+1)} - x^{(p)}|}{|x^{(p)} - x^{(p-1)}|^n} = \lim_{p\to\infty} \frac{|e^{(p)}|}{|e^{(p-1)}|^n} \approx K \tag{B}$$

Notice that the ratio $|e^{(p)}|/|e^{(p-1)}|$ (i.e., by setting $n = 1$ in Eq. (B)) in Table 4.6 approaches 0.81873, which confirms linear convergence. Also noting that $|e^{(p-1)}||e^{(p-2)}|^n \approx K$, we eliminate K to estimate the order of convergence as follows:

$$n \approx \frac{\log |e^{(p)}/e^{(p-1)}|}{\log |e^{(p-1)}/e^{(p-2)}|} \tag{C}$$

In this example, using Eq. (C) gives $n \approx \log(0.81867)/\log(.818765) = 1.0006$, which shows that the convergence rate is linear and its order is 1.

EXAMPLE 4.5: Applying the Secant Method

The flight path of a rocket is well described by the parametric equations $x(t) = 0.05t^{3/2}$, $y(t) = t(1.125 + 0.225t)$, and $z(t) = 0.25e^{0.2t}(t - 5) - 1.25$, where t is in seconds and x, y, and z are in km. How long will it take for the rocket to reach a speed of 6 km/s? Apply the *secant method* and use $\varepsilon = 0.5 \times 10^{-4}$ as tolerance.

SOLUTION:

The velocity is found by $u = dx/dt$, $v = dy/dt$, and $w = dz/dt$ as follows:

$$u = \frac{dx}{dt} = 0.075\sqrt{t}, \quad v = \frac{dy}{dt} = 1.125 + 0.45\,t, \quad \text{and} \quad w = \frac{dz}{dt} = 0.05\,t\,e^{0.2t}$$

The speed requirement is 6 km/s, so we may write

$$f(t) = u^2 + v^2 + w^2 - 6^2 = 0.005625\,t + (0.45\,t + 1.125)^2 + 0.0025\,t^2\,e^{0.4t} - 36 = 0$$

Thus, the problem is set up as a root-finding problem of a nonlinear equation.

In order to be able to apply the secant method, we need two initial guesses. We choose $t^{(0)} = 5$ and $t^{(1)} = 7$ seconds as the initial guesses since it is going to take at least a few seconds for the rocket to reach the indicated speed. For the starting initial values, we obtain

$$f(t^{(0)}) = f(5) = 0.005625(5) + (0.45(5) + 1.125)^2 + 0.0025\,(5)^2\,e^{0.4(5)} - 36 = -24.11943$$

$$f(t^{(1)}) = f(7) = 0.005625(7) + (0.45(7) + 1.125)^2 + 0.0025\,(7)^2\,e^{0.4(7)} - 36 = -15.67053$$

Using Eq. (4.22), the first estimate is found as

$$t^{(2)} = \frac{t^{(1)} f(t^{(0)}) - t^{(0)} f(t^{(1)})}{f(t^{(0)}) - f(t^{(1)})} = \frac{7(-24.119434) - 5(-15.67053)}{(-24.119434) - (-15.67053)} = 10.709483$$

which yields $|f(t^{(2)})| = 20.18696 > \varepsilon$ and $e^{(2)} = |t^{(2)} - t^{(1)}| = 3.709483 > \varepsilon$.

Using Eq. (4.22), the second estimate yields

$$t^{(3)} = \frac{t^{(2)} f(t^{(1)}) - t^{(1)} f(t^{(2)})}{f(t^{(1)}) - f(t^{(2)})} = \frac{10.709483(-15.67053) - (7)(20.186962)}{(-15.67053) - (20.186962)} = 8.621127$$

which gives $|f(t^{(2)})| = -5.062396 > \varepsilon$ and $e^{(3)} = |t^{(3)} - t^{(2)}| = 2.088355 > \varepsilon$. This procedure is continued in the same manner until the convergence criteria are met. This procedure is terminated after 8 iterations when $|f(t^{(8)})| < \varepsilon$ and $|e^{(8)}| < \varepsilon$. The iteration history along with the absolute error at the p'th iteration step, $e^{(p)} = |t^{(p+1)} - t^{(p)}|$, is summarized in Table 4.7.

Table 4.7: The computation history for solution using the Secant Method.

p	$t^{(p)}$	$f(t^{(p)})$	$t^{(p+1)})$	$e^{(p)}$	$e^{(p+1)}/e^{(p)}$
0	5	-24.1194			
1	7	-15.6705	10.70948	2.000000	
2	10.709483	20.18696	8.621127	3.709483	1.854741
3	8.621128	-5.062396	9.039835	2.088355	0.562977
4	9.039835	-1.385694	9.197639	0.418707	0.200496
5	9.197639	0.137731	9.183372	0.157804	0.376885
6	9.183372	-0.003387	9.183714	0.014267	0.090409
7	9.183714	-8.07×10^{-6}	9.183715	0.000342	0.024002
8	**9.183715**	4.74×10^{-10}	9.183715	8.1×10^{-7}	0.002388

The last column of the table gives the ratio of two consecutive absolute errors. We can use this data to estimate the order of convergence using Eq. (C) derived in Example 4.4. For the 7th and 8th iteration steps, the approximations for the order of convergence, n, yield 1.6187 and 1.6137, respectively. This indicates that the convergence of the secant method is typically *superlinear*, meaning that the error decreases faster than in linear convergence, but not as fast as quadratic convergence (like Newton's method).

Discussion: The secant method can be viewed as a simplification of Newton's method, which requires the exact derivative of the function. The idea behind the method is to approximate $f'(x)$ by using the finite difference between two consecutive function values. This is similar to the slope of the secant line connecting the points $\big(x^{(p-1)}, f(x^{(p-1)})\big)$ and $\big(x^{(p)}, f(x^{(p)})\big)$. This *secant line* intersects the x-axis at a point, and this intersection point becomes the next approximation.

1) Initial Guess: The Secant method uses two initial guesses, $x^{(0)}$ and $x^{(1)}$, to approximate the root of the function $f(x) = 0$. The method iteratively refines the estimates to the root by generating successive approximations: $x^{(2)}$, $x^{(3)}$, $x^{(4)}$, and so on. If the initial guesses are not close to the root, convergence may be slow, or the method may fail entirely.

2) Convergence Behavior: For the Secant method, the order of convergence is approximately $n \approx (1+\sqrt{5})/2 = 1.618$, which is the golden ratio, indicating that the Secant method has *superlinear convergence*; that is, it generally converges faster than methods like *Bisection* and *Regula Falsi* but slower than *Newton-Raphson*. Unlike methods like the bracketing methods, which always converge as long as $f(x)$ changes signs over an interval, the Secant method does not have guaranteed convergence. The method may fail if $f(x)$ behaves irregularly or the initial guesses are poorly chosen. Also, if $f(x)$ has multiple roots or inflection points close to the root, the method may struggle or diverge. If the function has a flat region near the root, the method might converge slowly, or if the guesses are too far from the root or the function has multiple roots, the method might fail to converge.

3) Iteration Equation: Unlike Newton-Raphson, which requires $f'(x)$ in the iteration equation, the Secant method only needs $f(x^{(p-1)})$ and $f(x^{(p)})$. This makes it useful when computing the derivative is difficult or expensive.

EXAMPLE 4.6: Applying Aitken's and Steffenses's Accelertion Techniques

Consider the sequence of the first six iterates of **Example 4.4**. (a) Apply Aitken's acceleration technique to the sequence to improve its convergence and calculate the convergence rates for both $t^{(p)}$) and $\alpha^{(p)}$) sequences; (b) Apply Steffensen's acceleration to estimate the root to within $\varepsilon = 5 \times 10^{-4}$.

SOLUTION:

(a) The first six iterates, $\{x^{(p)}\}$, obtained with FPI are read from Table 4.6 instead of recalculating. Starting with $p = 1$, we find

$$\begin{aligned}
\Delta x^{(1)} &= x^{(1)} - x^{(0)} = 0.235306 - 0.5 = -0.2646940 \\
\Delta x^{(2)} &= x^{(2)} - x^{(1)} = 0.443421 - 0.235306 = 0.208115 \\
\Delta^2 x^{(2)} &= \Delta x^{(2)} - \Delta x^{(1)} = 0.208115 - (-0.264694) = 0.472809
\end{aligned}$$

Then, for the current step, we find

$$\begin{aligned}
\alpha^{(1)} &= x^{(2)} - \frac{\left(\Delta x^{(2)}\right)^2}{\Delta^2 x^{(2)}} = 0.443421 - \frac{(0.208115)^2}{0.472809} = 0.351816 \\
\lambda^{(1)} &\approx \left|\Delta x^{(2)} / \Delta x^{(1)}\right| = |\,0.208115/0.264694\,| = 0.786248
\end{aligned}$$

When this procedure is repeated for $p = 2$, 3, and so on, it leads to a new sequence $\{\alpha^{(p)}\}$ of five elements. It is noted that the actual calculations were performed in double precision but rounded to six decimal places. Six iteration steps of Aitken's acceleration parameters are presented in Table 4.8 for $x^{(p)}$ and $\alpha^{(p)}$. The approximate convergence rates for the sequences $x^{(p)}$ and $\alpha^{(p)}$ are computed from $\lambda^{(p)} \approx \Delta x^{(p+1)} / \Delta x^{(p)}$ and $\lambda_\alpha{}^{(p)} \approx \Delta \alpha^{(p+1)} / \Delta \alpha^{(p)}$, respectively.

Table 4.8

p	$x^{(p)}$	$\Delta x^{(p)}$)	$\lvert\lambda^{(p)}\rvert$	$\alpha^{(p)}$	$\Delta\alpha^{(p)}$	$\lvert\lambda_\alpha{}^{(p)}\rvert$
0	0.5					
1	0.235306	−0.264694	0.786248	0.351816		
2	0.443421	0.208115	0.827831	0.349165	−0.002651	0.526566
3	0.271137	−0.172284	0.801160	0.347769	−0.001396	0.747924
4	0.409164	0.138027	0.826034	0.346725	−0.001044	0.584636
5	0.295149	−0.114015	0.808359	**0.346115**	**−0.000610**	
6	0.387314	0.092165				

The fixed point iteration sequence $\{x^{(p)}\}$ depicts an average convergence rate of $\lambda \approx 0.81$. An estimate for the order of convergence may be calculated using the last iteration steps as $n \approx \log(0.114015)/\log(0.138027) = 1.097$ and $n \approx \log(0.092165)/\log(0.114015) = 1.098$, respectively, i.e., linear convergence. However, the sequence $\{\alpha^{(p)}\}$ approaches the true root value of 0.3448317665 faster with an average convergence rate $\lambda_\alpha \to 0.62$. The order of convergence estimate using $\Delta\alpha$'s also depicts linear convergence, $n \approx \log(0.00061)/\log(0.001044) = 1.078$.

(b) To start Steffensen's acceleration procedure, we begin with the first three iterates of fixed point iteration, $x^{(0)}$, $x^{(1)}$, $x^{(2)}$. Using Eq. (4.29), the first estimate with Aitken's acceleration, $\alpha^{(1)}$, is found as

$$\alpha^{(1)} = x^{(2)} - \frac{\left(\Delta x^{(2)}\right)^2}{\Delta^2 x^{(2)}} = 0.443421 - \frac{(0.208115)^2}{0.4728090} = 0.351816$$

with $\Delta\alpha^{(1)} = \alpha^{(1)} - x^{(0)} = 0.351816 - 0.5 = -0.148184$.

We use $\alpha^{(1)}$ to restart fixed point iterations, $x^{(1)} = \alpha^{(1)}$, and calculate the next two iterates as

$$x^{(2)} = g(x^{(1)}) = g(0.351816) = 0.339149$$
$$x^{(3)} = g(x^{(2)}) = g(0.339149) = 0.349508$$

Next, we compute

$$\begin{aligned}
\Delta x^{(2)} &= x^{(2)} - x^{(1)} = 0.339149 - 0.351816 = -0.012667 \\
\Delta x^{(3)} &= x^{(3)} - x^{(2)} = 0.349508 - 0.339149 = 0.010359 \\
\Delta^2 x^{(3)} &= \Delta x^{(3)} - \Delta x^{(2)} = 0.010359 - (-0.012667) = 0.023026
\end{aligned}$$

Substituting $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$ into Eq. (4.29), we obtain

$$\alpha^{(2)} = x^{(3)} - \frac{\left(\Delta x^{(3)}\right)^2}{\Delta^2 x^{(3)}} = 0.349508 - \frac{0.010359^2}{0.023026} = 0.344847$$

The iteration history of Steffensen's acceleration is summarized in Table 4.9. As shown in this exercise, this technique required only three iterations to converge the root within the desired tolerance, i.e., $|\alpha^{(3)} - \alpha^{(2)}| = 1.58 \times 10^{-5} < \varepsilon$ and $|\alpha^{(3)} - g(\alpha^{(3)})| = 1.06 \times 10^{-5} < \varepsilon$. The order of convergence for $\{\alpha\}$'s gives $n \approx \log(0.006968)/\log(0.148184) = 2.601$ and $n \approx \log(1.58{\times}10^{-5})/\log(0.006968) = 2.226$, indicating a quadratic order or better. Considering that the FPI method converged in 32 iterations, the speed at which Steffessen's acceleration converged has been clearly demonstrated.

Table 4.9

p	$t^{(p-1)}$	$t^{(p)})$	$t^{(p+1)}$	$\lvert\alpha^{(p)}\rvert$	$\Delta\alpha^{(p)}$	$\lvert\alpha^{(p)} - g(\alpha^{(p)})\rvert$
1	0.5	0.235306	0.443421	0.351816	−0.148184	0.091605
2	0.351816	0.339149	0.349577	0.344847	−0.006968	0.004660
3	0.344848	0.344819	0.344842	**0.344832**	**-1.58×10^{-5}**	**1.06×10^{-5}**

Discussion: Aitken's method is used to accelerate the convergence of a sequence. Specifically, it improves the rate of convergence of sequences that converge to a limit, especially when the sequence converges slowly. Applying Aitken's method can produce a new sequence that converges to L much more quickly. By accelerating the convergence, the method can yield more accurate approximations with fewer terms. This can be particularly helpful when calculating *limits* or evaluating *series* where calculating the exact value requires many terms. The Aitken's formula requires only three consecutive terms of the sequence to compute an accelerated value. The method is useful for sequences that converge linearly, where the sequence approaches the limit in a steady, but slow manner. In such cases, Aitken's method can improve the convergence rate to quadratic or even faster.

Steffensen's method is a technique to accelerate the convergence of a sequence, particularly in the context of iterative methods. It does not require derivatives of $f(x)$; instead, it works directly with the sequence, making it highly useful for problems where derivatives are difficult or costly to compute. The method generally results in a faster rate of convergence—usually from linear to quadratic convergence or even higher. This formula is similar to Aitken's method but makes use of updated values as soon as they become available, which leads to a more rapid convergence. However, the method assumes that the original sequence is converging. If it does not converge, or if it oscillates or diverges, the method is ineffective.

EXAMPLE 4.7: Applying Newton's Method to A System of Nonlinear Equations

At 1000 K, the principal reactions between methane and steam are

$$\underbrace{CH_4}_{1-x} + \underbrace{H_2O}_{5-x} \rightleftarrows \underbrace{CO}_{x} + \underbrace{3H_2}_{3x}, \qquad \underbrace{CO}_{x-y} + \underbrace{H_2O}_{5-x-y} \rightleftarrows \underbrace{CO_2}_{y} + \underbrace{H_2}_{3x+y}$$

where x and y are fractional compositions of [CO] and [CO_2], respectively. Equilibrium constants for the reactions are given as $K_1 = 4.046$ and $K_2 = 1.237$, respectively. Starting with 1 mole of methane and 5 moles of steam, the equilibrium constant expression can be written as follows:

$$\frac{(x-y)(3x+y)^3}{(1-x)(5-x-y)(6+2x)^2} = 4.046, \qquad \frac{y(3x+y)}{(x-y)(5-x-y)} = 1.237$$

Find the equilibrium compositions of the reaction products correct to *four-decimal places* using *Newton's method* with $x^{(0)} = 0.8$ and $y^{(0)} = 0.5$ as starting values.

SOLUTION:

If we rearrange these equations as follows, we have a system of two nonlinear equations and two unknowns:

$$\begin{aligned} f(x,y) &= -4.046(1-x)(6+2x)^2(5-x-y) + (x-y)(3x+y)^3 = 0 \\ g(x,y) &= -1.237(5-x-y)(x-y) + y(3x+y) = 0 \end{aligned}$$

To construct the Jacobian, the partial derivatives of f and g with respect to x and y are found as follows:

$$\frac{\partial f}{\partial x} = 388.416 + 712.096\,x + 43.264\,x^3 - 48.552y - 161.84\,xy - 48.552\,x^2y - 36xy^2 - 8y^3$$

$$\frac{\partial f}{\partial y} = 145.656 - 48.552x - 80.92x^2 - 16.184x^3 - 36x^2y - 24xy^2 - 4y^3$$

$$\frac{\partial g}{\partial x} = -6.185 + 2.474x + 3y, \qquad \frac{\partial g}{\partial y} = 6.185 + 3x - 0.474y$$

Then the Newton's iteration equation can be written in matrix equation notation as

$$\mathbf{x}^{(p+1)} = \mathbf{x}^{(p)} - \left[\mathbf{J}(\mathbf{x}^{(p)})\right]^{-1}\mathbf{f}(\mathbf{x}^{(p)})$$

We start with the initial guess $\mathbf{x}^{(0)} = [\,0.8 \;\; 0.5\,]^T$ and evaluate $\mathbf{f}$ and $\mathbf{J}$:

$$\mathbf{f}(\mathbf{x}^{(0)}) = \begin{bmatrix} -165.62 \\ 0.07693 \end{bmatrix}, \qquad \mathbf{J}(\mathbf{x}^{(0)}) = \begin{bmatrix} 867.495328 & 29.919392 \\ -2.7058 & 8.348 \end{bmatrix}$$

The new (current) estimate is found as follows:

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - \left[\mathbf{J}(\mathbf{x}^{(0)})\right]^{-1}\mathbf{f}(\mathbf{x}^{(0)}) \\ &= \begin{bmatrix} 0.8 \\ 0.5 \end{bmatrix} - \begin{bmatrix} 0.00114 & -0.0040858 \\ 0.0003695 & 0.1184649 \end{bmatrix} \begin{bmatrix} -165.62 \\ 0.07693 \end{bmatrix} = \begin{bmatrix} 0.98912 \\ 0.55208 \end{bmatrix} \end{aligned}$$

The iterative procedure is repeated until $|\mathbf{f}(\mathbf{x}^{(p)})| < \varepsilon$. In this example, the system converged to a solution set at the end of the 3rd iteration with $|\mathbf{f}(\mathbf{x}^{(2)})| < 0.5 \times 10^{-4}$ and $|\boldsymbol{\delta}^{(2)}| < 0.5 \times 10^{-4}$.

Table 4.10

p	$x^{(p)}$	$y^{(p)})$	$\delta_x^{(p)}$	$\delta_y^{(p)}$	$\Vert\boldsymbol{\delta}^{(p)}\Vert_2$
0	0.8	0.5	−0.018910	−0.00521	0.01962
1	0.989120	0.552083	0.009805	0.01052	0.01438
2	0.979315	0.541560	-7.711×10^{-7}	4.518×10^{-5}	4.519×10^{-5}
3	0.979316	0.541515			

Discussion: Numerical methods for solving a system of nonlinear equations provide approximate solutions when analytical solutions are difficult or impossible to obtain. Newton's method is an iterative method that linearizes the nonlinear system near an initial guess. It is particularly useful for large and complex systems where an exact solution is not feasible.

1) Convergence: Newton's method is a generalization of the Newton-Raphson method, and it converges quickly (with *quadratic convergence*) when the initial guess is close to the solution and the Jacobian matrix is well-conditioned.

2) Stability: The method can fail if the procedure is started with a poor set of initial guess (i.e., far away from the true solution) or the Jacobian is singular, near-singular, or ill-conditioned.

3) Sensitivity to Initial Guess: This method can be highly sensitive to the choice of initial guess. It requires a good initial guess to ensure convergence; a poor initial guess may lead to divergence or convergence to a wrong (one of the other, but physically unreasonable) solution.

3) Accuracy: The method is highly accurate if it converges, especially for well-behaved systems.

4) Computation Cost: The method requires the construction of the inverse of the Jacobian matrix and computation of its inverse at each iteration, which can be computationally expensive for large systems.

4) Best Performance: Newton's method is one of the most effective of the numerical methods widely used in solving nonlinear equation systems. It performs best when the Jacobian is easy to compute and a good initial guess is available. It is highly effective for small-to-medium-sized systems with a well-behaved Jacobian.

EXAMPLE 4.8: Roots of a Polynomial using the Bairstow's Method

Consider **Example 4.3**, yielding a third-degree polynomial. Apply the Bairstow's method with $p^{(0)}=-1$, $q^{(0)}=1$ as the initial guess and use $|\delta p|+|\delta q|<5\times 10^{-6}$ as the stopping criterion.

SOLUTION:

Rearranging the Van der Waals equation leads to the following 3rd-degree polynomial.

$$V^3 - 66.7282V^2 + 120.443V - 44.1546 = 0$$

where the coefficients of the 3rd degree polynomial are $a_0 = 1$, $a_1 = -66.7282$, $a_2 = 120.443$, and $a_3 = -44.1546$.

1st iteration: We start with $p^{(0)} = -1$, $q^{(0)} = 1$. The b_k's and c_k's are computed from Eq. (4.46) and Eq. (4.56), respectively.

k	a_k	b_k	c_k	
0	1	1	1	$\bar{c} = c_2 - b_2 = -65.7282$
1	-66.7282	-65.7282	-64.7282	$\delta p = -0.834726$
2	120.443	53.7148	-12.0134	$\delta q = -0.315525$
3	-44.1546	75.2884		$\lvert\delta p\rvert+\lvert\delta q\rvert=1.1502>\varepsilon$

The first iteration gives

$$\begin{aligned} p^{(1)} &= p^{(0)} + \delta p = -1 + (-0.834726) = -1.834726 \\ q^{(1)} &= q^{(0)} + \delta q = 1 + (-0.315525) = 0.684475 \end{aligned}$$

2nd iteration: Since $|\delta p|+|\delta q| > \varepsilon$, we continue with $p^{(1)} = -1.834726$, $q^{(1)} = 0.684475$. The polynomial coefficients b_k's and c_k's are likewise computed and tabulated below:

k	a_k	b_k	c_k	
0	1	1	1	$\bar{c} = c_2 - b_2 = -116.38$
1	-66.7282	-64.8935	-63.05875	$\delta p = -0.011112$
2	120.443	0.696768	-115.6832	$\delta q = -0.003941$
3	-44.1546	1.541755		$\lvert\delta p\rvert+\lvert\delta q\rvert=0.0151>\varepsilon$

The second iteration yields

$$\begin{aligned} p^{(2)} &= p^{(1)} + \delta p = -1.834726 + (-0.011112) = -1.845838 \\ q^{(2)} &= q^{(1)} + \delta q = 0.684475 + (-0.003941) = 0.680534 \end{aligned}$$

3rd iteration: The convergence criterion $|\delta p|+|\delta q| > \varepsilon$ has yet to be satisfied, so we iterate one more time with $p^{(2)}=-1.845838$, $q^{(1)}=0.680534$. The computed coefficients are tabulated below:

k	a_k	b_k	c_k	
0	1	1	1	$\bar{c} = c_2 - b_2 = -117.0358$
1	-66.7282	-64.8824	-63.03652	$\delta p = -1.97 \times 10^{-6}$
2	120.443	0.000124	-117.0356	$\delta q = -6.54 \times 10^{-7}$
3	-44.1546	0.000272		$\lvert\delta p\rvert+\lvert\delta q\rvert=2.6\times 10^{-6}<\varepsilon$

At the end of the 3rd iteration, the convergence criterion is satisfied; thus, the iteration process can be terminated. Then, the final (3rd) iteration yields

$$p^{(3)} = p^{(2)} + \delta p = -1.845838 + (-1.97 \times 10^{-6}) = -1.8458402$$
$$q^{(3)} = q^{(2)} + \delta q = 0.680534 + (-6.54 \times 10^{-7}) = 0.6805332$$

which are the coefficients of a quadratic factor: $(V^2 - 1.8458402V + 0.6805332)$. Note that the deflated polynomial is first degree, $(V - 64.8824)$, and b_2 and b_3 denoting the remainder terms (R and S) approach zero. Finally, the 3rd degree polynomial can be expressed as follows:

$$V^3 - 66.7282V^2 + 120.443V - 44.1546 = (V^2 - 1.8458402V + 0.6805332)(V - 64.8824) = 0$$

Next, solving the quadratic equation, we find $V = 0.509099$, $V = 1.33674$, and $V = 64.882$ L. The largest root is the solution of the **Example 4.3**, but the method produced all the roots rather than the one we were seeking. We could have solved the same problem faster with less effort if we had used the Newton-Raphson method with $V^{(0)} = 66.362$ (obtained by using the ideal gas assumption).

Discussion: Bairstow's method can be used to find the roots of polynomials, particularly for solving quadratic factors of real-coefficient polynomials. Unlike Newton's method, Bairstow's method does not require derivatives and only relies on polynomial coefficients and synthetic division, which makes it computationally attractive in certain cases. The method has several benefits, making it a valuable tool for polynomial root-finding:

1. Suitability: Bairstow's method directly finds quadratic factors, making it particularly useful for polynomials with real coefficients. This feature allows the users to find both real and imaginary roots without requiring explicit complex arithmetic.

2. Efficiency: By targeting quadratic factors, the method finds two roots simultaneously, which can reduce the total number of iterations needed compared to methods that solve one root at a time.

3. Recursive application: It works well for polynomials of high degree, especially when combined with iterative refinement and proper initial guesses. That is, once a quadratic factor is determined, synthetic division can be used to reduce the degree of the polynomial. This reduced polynomial can then be solved recursively using the same method.

4. Handles complex roots automatically: The method inherently finds conjugate complex roots of real-coefficient polynomials as pairs, avoiding the need for separate handling of complex arithmetic.

While Bairstow's method is efficient and versatile, it can encounter sensitivity and stability issues. For example, like other iterative methods, it may fail to converge or converge slowly if the initial guesses, $p^{(0)}$ and $q^{(0)}$, are poor. On the other hand, numerical instability can arise in high-degree polynomials or in cases with closely spaced roots. Such problems, however, can be mitigated by using alternative root-finding methods to obtain a good starting initial guess, refining solutions with techniques like deflation, or by increasing numerical precision.