7.4.3 Convergence Promotion Methods

We have seen that recycle calculations involve tearing a stream and converging the estimated and calculated values of the stream variables. The calculated values are obtained after working through the recycle loop. If we set $\mathbf{x}$ as the estimated recycle values of all variables in the tear stream and $\mathbf{f}(\mathbf{x})$ as the calculated recycle values of all variables in the tear stream after calculating through the loop, then convergence is achieved when

$$\mathbf{f}(\mathbf{x}) = \mathbf{x} \tag{7.6}$$

within a preset tolerance. The basic problem is then to solve the system of non-linear equations in eqn (7.6). Several solution methods have been developed, all requiring some form of iteration.

Direct Substitution Method

The direct substitution or successive substitution, as it is sometimes called, is the simplest of iterative methods. The method sets

$$\mathbf{x}^{(r+1)} = \mathbf{f}(\mathbf{x}^{(r)}) \tag{7.7}$$

where superscript r denotes rth iteration. In other words, new estimates of $\mathbf{x}$ are the most recently calculated values of $\mathbf{f}(\mathbf{x})$. This method is mostly stable if the initial guess is close to the actual value, but the convergence is very slow and a large number of iterations are required.

Bounded Wegstein Method

The simple Wegstein method for a single variable is described in the Appendix. When the Wegstein procedure is used for a system of equations, it is applied to each equation in turn.

$$x_i^{(r+1)} = q_i x_i^{(r)} + (1 - q_i) f_i(\mathbf{x}^{(r)}) \qquad (i = 1, 2, \ldots, n) \tag{7.8}$$

where q_i is an acceleration parameter for the ith variable

$$q_i = \frac{a_i}{a_i - 1} \tag{7.9}$$

and

$$a_i = \frac{f_i(\mathbf{x}^{(r)}) - f_i(\mathbf{x}^{(r-1)})}{x_i^{(r)} - x_i^{(r-1)}} \tag{7.10}$$

For $q_i = 0$ $(i = 1, 2, \ldots, n)$, eqn (7.8) reduces to direct substitution. Equation (7.8) is applied once every few steps to minimize computational efforts. Wegstein's method for a multivariable problem implicitly assumes that all the variables are uncoupled, which is often not true. Therefore, direct application of the Wegstein procedure may sometimes lead to instability. Rosen and Pauls[36] suggested that the acceleration parameter must be bound between the limits

$$0 \le q_i \le -5 \tag{7.11}$$

If q_i is outside this range, then q_i is set to zero, which is reverting to direct substitution for variable i. The bounding of q_i improves the stability of iteration.

Dominant Eigenvalue Method

This method was first proposed by Orbach and Crowe.[58]

$$x_i^{(r+1)} = q x_i^{(r)} + (1 - q) f_i(\mathbf{x}^{(r)}) \tag{7.12}$$

where q is a scalar defined as

$$q = \frac{\lambda}{\lambda - 1} \tag{7.13}$$

and λ is an estimate of the dominant eigenvalue of the iteration matrix. The iteration matrix $\mathbf{P}$ is a matrix of linear terms, obtained from the approximation

$$\mathbf{x}^{(r+1)} = \mathbf{P}\mathbf{x}^{(r)} + \mathbf{c} \tag{7.14}$$

Details of the derivation are given by Orbach and Crowe.[58] The estimate of maximal eigenvalue of P, namely λ, is given by

$$\lambda = \frac{\|\mathbf{f}(\mathbf{x}^{(r)}) - \mathbf{f}(\mathbf{x}^{(r-1)})\|}{\|\mathbf{x}^{(r)} - \mathbf{x}^{(r-1)}\|} \tag{7.15}$$

where $\|\cdot\|$ is the Euclidean norm.

Almost all simulators provide the direct substitution and Wegstein or bounded Wegstein methods as part of the routines for recycle calculations. Not all simulators provide the dominant eigenvalue method. PROCESS is one simulator which offers the choice of dominant eigenvalue method. Bounded Wegstein is perhaps the most popular of the methods currently used.

A.2 SOLUTION OF EQUATIONS

The solution of systems of linear algebraic equations is obtained by matrix methods, as described above. In a number of situations, it is required to find the roots of a single or of a system of non-linear algebraic or transcendental equations.

A.2.1 The Newton–Raphson Method

A popular and powerful gradient method to evaluate roots of non-linear and transcendental equations is the Newton–Raphson method. Sometimes it is also referred to as the Newton method.

The solution is required for

$$f(x) = 0 \tag{A.14}$$

If $x^{(r)}$ is an approximation to the root, then $f(x)$ can be expanded in a Taylor series in the neighbourhood of $x^{(r)}$, so that if the real root $\bar{x} = x^{(r)} + h$, then

$$0 = f(x^{(r)} + h) = f(x^{(r)}) + hf'(x^{(r)}) + 0(h^2) \tag{A.15}$$

Neglecting higher order terms, we have

$$h = -f(x^{(r)})/f'(x^{(r)}) \tag{A.16}$$

Therefore, the next approximation to the root is given by

$$x^{(r+1)} = x^{(r)} - f(x^{(r)})/f'(x^{(r)}) \tag{A.17}$$

The above procedure is repeatedly used until $|f(x^{(r)})| < \varepsilon$, where ε is a preset tolerance. This method converges rapidly if the initial guess is close to the actual root. Otherwise, depending on the nature of the function, the method could diverge. Even for diverging situations, the Newton method can be applied by constraining the value of the correction h in incrementing a previous approximation. This can be done in two ways.

(1) Multiply h by a relaxation factor $\alpha(0 < \alpha < 1)$, such that only a fraction of h is used to increment the approximation.

(2) Fix a maximum value h_{max} for $|h|$ for any iteration, such that if $|h| > h_{max}$, then h_{max} with the sign of h is used for incrementing the next approximation. The value of h_{max} can be taken as a fraction β of $x^{(r)}$.

Both the above procedures generally give reliable convergence. The values of α and β have to be decided by the user, depending on the nature of the problem. In a majority of instances, α can be taken as 0·4 and β can be taken as 0·1.

Extensive application of the Newton–Raphson method is to be found in the examples in the text, using both the step-limiting techniques.

A.2.2 Generalized Newton–Raphson Method

This method is an extension of Newton's method for simultaneous non-linear equations. Suppose we have n equations

$$f_i(\mathbf{x}) = 0 \qquad (i = 1, 2, \ldots, n) \tag{A.18}$$

where $\mathbf{x}$ is the root given by $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. Expanding f_i in a Taylor series in the neighbourhood of an approximation $\mathbf{x}^{(r)}$

$$0 = f_i(\mathbf{x}^{(r)} + \mathbf{h}) = f_i(\mathbf{x}^{(r)}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} h_j + 0(h^2) \quad (i = 1, 2, \ldots, n) \tag{A.19}$$

Equation (A.19), in matrix form, becomes

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \cdots \\ -f_n \end{bmatrix} \tag{A.20}$$

The vector $\mathbf{h}$ is obtained by solving the above system of linear equations. The next approximation is given by

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \mathbf{h} \tag{A.21}$$

The procedure is repeated until convergence is achieved, the criterion being

$$\sum_{i=1}^{n} (f_i^{(r)})^2 < \varepsilon \tag{A.22}$$

where ε is a preset tolerance.

The matrix in eqn (A.20), known as the Jacobian, need not be inverted at every iteration in solving for h. Considerable computation time can be saved by updating the inverse Jacobian every few iterations. To improve the convergence, the increments h can be restricted by multiplying by a relaxation factor α. Alternatively, h_j can be restricted to within preassigned values $(h_{max})_j$. However, convergence essentially depends upon a very good initial estimate of the roots.

An application of the generalized Newton method is provided in Chapter 4, Program P-4.4, in pipe network calculations.

A.2.3 Broyden's Method

The method of Broyden,[11] intended for simultaneous non-linear algebraic equations, is a modification of the generalized Newton–Raphson method,

designed to reduce the number of function evaluations required. The Jacobian in eqn (A.20) is evaluated only once, and in subsequent iterations corrections to the approximate inverse of the Jacobian are computed from values of the vector function f. The relaxation factor or scalar multiplier is chosen in such a way that the Euclidean norm of the vector function f is reduced in each successive iteration and, therefore, the process is prevented from diverging. The method has been used with considerable success in multi-component distillation calculations (see Chapter 3, Section 3.2.4). A summary of the method is outlined below.

Algorithm

(1) Assume an initial estimate $\mathbf{x}_0$ of the solution vector.

(2) Compute $f_i(x_0)$, $i = 1, 2, \ldots, n$.

(3) Calculate the coefficients of the Jacobian matrix J_0, by using a difference procedure for the partial derivatives

$$\frac{\partial f_i}{\partial x_j} \simeq \frac{f_i(x_j + \Delta x_j) - f_i(x_j)}{\Delta x_j} \tag{A.23}$$

where $\Delta x_j = 0{\cdot}001 x_j$.

(4) Invert the Jacobian and define

$$\mathbf{H}_0 = -\mathbf{J}_0^{-1} \tag{A.24}$$

(5) On the basis of the most recent values of H and f, say $\mathbf{H}^{(r)}$ and $\mathbf{f}^{(r)}$, compute

$$\mathbf{p}^{(r)} = \mathbf{H}^{(r)} \cdot \mathbf{f}^{(r)} \tag{A.25}$$

(6) Select a value of $t^{(r)}$ such that the norm of $\mathbf{f}(\mathbf{x}^{(r)} + t^{(r)}\mathbf{p}^{(r)})$ is less than the norm of $\mathbf{f}(\mathbf{x}^{(r)})$. The value of $\mathbf{t}^{(r)}$ is obtained as follows:

Initially try $t_{(1)}^{(r)} = 1$. If

$$\left[\sum_{i=1}^{n} f_i^2(\mathbf{x}^{(r)} + t^{(r)}\mathbf{p}^{(r)})\right]^{1/2} < \left[\sum_{i=1}^{n} f_i^2(\mathbf{x}^{(r)})\right]^{1/2} \tag{A.26}$$

then set $\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + t^{(r)}\mathbf{p}^{(r)}$ and go to step (7). Otherwise, compute a new value $t_{(2)}^{(r)}$ using the formula

$$t_{(2)}^{(r)} = \frac{(1 + 6\theta)^{1/2} - 1}{3\theta} \tag{A.27}$$

where

$$\theta = \frac{\sum_{i=1}^{n} f_i^2(\mathbf{x}^{(r)} + t_{(2)}^{(r)}\mathbf{p}^{(r)})}{\sum_{i=1}^{n} f_i^2(\mathbf{x}^{(r)})} \tag{A.28}$$

Test whether the inequality (A.26) is now satisfied with $t_{(2)}^{(r)}$. If the norm is not reduced by the use of $t_{(2)}^{(r)}$, then set $\mathbf{x}^{(r)} = \mathbf{x}^{(r)} + t_{(2)}^{(r)}\mathbf{p}^{(r)}$ and go to step (3) to re-evaluate the Jacobian.

(7) Test for convergence. The convergence criterion is

$$\sum_{i=1}^{n} f_i^2(\mathbf{x}^{(r+1)}) < \varepsilon \tag{A.29}$$

As a guide, the tolerance ε can be taken as 10^{-6} when the f_i values are normalized. If convergence is achieved, then exit from the subprogram.

(8) Calculate

$$y_i^{(r)} = f_i(\mathbf{x}^{(r+1)}) - f_i(\mathbf{x}^{(r)}) \qquad (i = 1, 2, \ldots, n) \tag{A.30}$$

(9) Calculate

$$\mathbf{H}^{(r+1)} = \mathbf{H}^{(r)} - \frac{(\mathbf{H}^{(r)}\mathbf{y}^{(r)} + t^{(r)}\mathbf{p}^{(r)})\mathbf{p}^{(r)\mathrm{T}}\mathbf{H}^{(r)}}{\mathbf{p}^{(r)\mathrm{T}}\mathbf{H}^{(r)}\mathbf{y}^{(r)}} \tag{A.31}$$

and return to step (5).

A subprogram for the Broyden method is given in Program P-3.3.

If, in a given problem, a particular variable x_i has a non-negativity constraint imposed on it, then Napthali[12] has suggested the following procedure.

Examine the value of $x_i^{(r)} + h_i$, where h_i is the correction obtained by the Newton or quasi-Newton algorithm. If $x_i^{(r)} + h_i$ is negative, then the correction applied takes the form

$$x_i^{(r+1)} = x_i^{(r)} \exp(h_i / x_i^{(r)}) \tag{A.32}$$

A.2.4 Accelerating the Rate of Convergence

Among the methods developed for accelerating the convergence rate, the algorithm of Wegstein[4] gives reliable predictions of the root even in systems that would not otherwise converge. An iterative pattern of the form

$$x^{(r+1)} = F(x^{(r)}) \tag{A.33}$$

is set up. The procedure is outlined below.

Algorithm

(1) Generate three values of the root $x^{(r-1)}$, $x^{(r)}$ and $x^{(r+1)}$ using eqn (A.33). The functional form depends on $f(x)$. If $f(x)$ contains an explicit linear term in x, then eqn (A.33) results straightaway. In non-linear functions of $f(x)$, the functional form $F(x)$ can be generated from a Newton method.

(2) Define

$$A = \frac{x^{(r+1)} - x^{(r)}}{x^{(r)} - x^{(r-1)}} \tag{A.34}$$

and

$$q = A/(A - 1) \tag{A.35}$$

(3) The next approximation to the root is given by

$$x^{(r+2)} = qx^{(r)} + (1 - q)x^{(r+1)} \tag{A.36}$$

(4) Repeat the procedure until convergence is achieved. The Wegstein method is very powerful for convergent systems.