1
Large-Scale Algebraic Systems

Guido Buzzi–Ferraris and Davide Manca

1.1
Introduction

In this section, we address the solution of a system of N nonlinear equations:

$$\mathbf{f}(\mathbf{x}) = 0 \tag{1}$$

in N unknowns, $\mathbf{x}$, with particular attention given to large systems.

It is worth noting that the equations of system (1) must not necessarily be algebraic but may originate, for example, from the solution of a differential system with some initial conditions, or by the evaluation of the upper limit of an integral equation.

The solution of nonlinear equations is therefore significant not only as a problem *per se*, but it is also connected to the solution of differential-algebraic equation (DAE) and ordinary differential equation (ODE) stiff problems.

In the following, we will describe some iterative methods for the solution of system (1). With the term "iteration" we mean that given a previous point $\mathbf{x}_i$ the following one is determined by the equation:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i \tag{2}$$

The numerical methods for the solution of Non Linear Systems, NLSs, are characterized by the selection of direction $\mathbf{p}_i$ *and by the amplitude of the movement* α_i along $\mathbf{p}_i$.

Some methods require the evaluation of the Jacobian matrix defined as:

$$\mathbf{J}(\mathbf{x}_i) = \{J_{m,n}(\mathbf{x}_i)\} = \left\{ \frac{\partial f_m(\mathbf{x}_i)}{\partial x_n} \right\} \quad m = 1, \ldots, N \quad n = 1, \ldots, N \tag{3}$$

In the following, $\mathbf{f}_i$ represents $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{J}_i$ represents $\mathbf{J}(\mathbf{x}_i)$.

As far as large systems are concerned, it is instinctive to try to reduce the dimensions of the problem.

Using this idea, some previous numerical techniques, such as *tearing and partitioning*, were developed to automatically rearrange the system in order to minimize the

Computer Aided Process and Product Engineering. Edited by Luis Puigjaner and Georges Heyen
Copyright © 2006 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 3-527-30804-0

number of equations to be solved simultaneously. Unfortunately, the following problems should not be underestimated:

- It is not certain that the solution of a small NLS requires less time then a larger one.
- In an NLS, the role of unknowns within an equation is not symmetric. In other words, a function can be easily solved with respect to a variable but it can be difficult to find the solution when another unknown is involved.
- In spite of the original NLS being well-conditioned, the reduced NLS obtained from the original one can be ill-conditioned.

The first problem arises from the fact that it is not possible to evaluate the nonlinearity of a system of equations. In other words, contrary to the linear case, it is not possible to determine *a priori* the time required to solve an NLS as a function of its dimension. For example, system (4) was shown to be easier to solve than the smaller system (5) by Powell (1970).

$$\begin{aligned} f_1 &= x_1x_2 - 1 = 0 \\ f_2 &= x_1x_3 - 1 = 0 \\ f_3 &= x_1 - 1 = 0 \end{aligned} \tag{4}$$

$$\begin{aligned} f_1 &= 10000x_1x_2 - 1 = 0 \\ f_2 &= e^{-x_1} + e^{-x_2} - 1.0001 = 0 \end{aligned} \tag{5}$$

The second problem is once again bound to the nonlinearity of the system. An example is given by the conversion c in an adiabatic reactor as a function of the reaction temperature T. Often, it is possible to write the following energy balance:

$$c = g(T) \tag{6}$$

Evidently, it is trivial to determine the conversion when T is assigned. Conversely, it may not be easy to evaluate the temperature that produces a specific conversion.

The third problem is due to the fact that a rearrangement of the NLS can introduce an ill-conditioning that was not originally present. Let us suppose we have to solve the following system:

$$\begin{aligned} x_1 + 1000x_4 &= 1001 \\ 1000x_1 + x_2 &= 1001 \\ 1000x_2 + x_3 &= 1001 \\ x_1 + x_2 + x_3 + x_4 &= 4 \end{aligned} \tag{7}$$

whose solution is: $x_1 = x_2 = x_3 = x_4 = 1$.

System (7) can be rearranged into:

$$\begin{aligned} x_1 &= 1001 - 1000x_4 \\ x_2 &= 1001 - 1000x_1 \\ x_3 &= 1001 - 1000x_2 \\ f_4 &= x_1 + x_2 + x_3 + x_4 - 4 = 0 \end{aligned} \tag{8}$$

The new problem (8), although being characterized by only one equation with unknown x_4 has an extremely ill-conditioned form. As a matter of fact, if x_4 is evaluated numerically as: $x_4 = 0.99999$, then the following values are obtained: $x_1 = 1.010014$, $x_2 = -9.013580$, $x_3 = 10{,}014.58$ and $f(x_4) = 10{,}003.58$. It should be emphasized that system (8) is linear (therefore a simpler problem) and that it is reduced to a very simple equation in one unknown: x_4. Quite often, it is better to avoid any manipulation of the system if the goal is to reduce its dimensions. Actually, it is advisable to leave the NLS in its original form since it comes from modeling a physical phenomenon. Doing so, there are more guarantees that the numerical system is well-posed because it describes a real problem. What should be done is something that is apparently similar to rearranging the system but it is conceptually quite different. It is advisable to try exploiting the structure of the system without manipulating it.

A very simple example is represented by the solution of a steady-state distillation column. The liquid-vapor equilibria and the material balances of the unit should not be solved stage by stage, in a top-bottom sequence, while iterating towards convergence through the overall material balance of the column so to have the input/output flowrates consistent. By doing so, the physical structure of the problem would shift to obey to a sequential mathematical algorithm that would solve several apparently simplified subproblems. Such an approach would not respect the physical structure of the equilibrium stage in the sense that it would be equivalent to solving a flash problem starting from the known composition of one output stream to determine the compositions of the input and second output streams.

On the contrary, the intrinsic structure of a distillation column brings tridiagonal organization to the correlated mathematical problem. Such a tridiagonal structure should be exploited to efficiently solve the numerical problem.

1.2 Convergence Tests

Working at the implementation of a numerical algorithm for the solution of NLSs, a quite important matter must be addressed. How do we determine if the new estimate, $\mathbf{x}_{i+1}$, is better or worse than the previous one, $\mathbf{x}_i$?

The typical approach is to accept the new value, $\mathbf{x}_{i+1}$, if:

$$\|\mathbf{f}(\mathbf{x}_{i+1})\|_2 < \|\mathbf{f}(\mathbf{x}_i)\|_2 \tag{9}$$

or, equivalently, if there is a decrease in the merit function:

$$\Phi(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{N} f_j^2(\mathbf{x}) = \frac{1}{2}\mathbf{f}^T\mathbf{f} \tag{10}$$

The criterion represented by Eqs. (9) and (10) should be avoided within a general-purpose program whenever the functions **f** are unbalanced and have significantly different orders of magnitude. In those cases, the equations with lower orders of magnitude do not contribute to the merit function. As an example, we can report the evaluation of a flash, or the modeling of a distillation column. The stoichiometric equations (order of magnitude: 1) stay together with the enthalpy balance equations (order of magnitude: 1.E6–1.E9) and significant differences in terms of orders of magnitude are present in the resulting NLS.

An improvement of the previous criterion is given by weighting each equation with a suitable weight, w_j. Consequently, the merit function becomes:

$$\Phi_w(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{N} w_j^2 f_j^2(\mathbf{x}) \tag{11}$$

By introducing the diagonal matrix, **W**, which has elements equal to the weights, w_j, the matrix notation follows:

$$\Phi_w(\mathbf{x}) = \frac{1}{2}(\mathbf{W}\mathbf{f})^T(\mathbf{W}\mathbf{f}) = \frac{1}{2}\mathbf{f}^T\mathbf{W}^2\mathbf{f} \tag{12}$$

More generally, the weights can vary with the iterations. Consequently, the weight matrix becomes $\mathbf{W}_i$.

A reasonable criterion for the definition of the weights is to make all equations have the same order of magnitude. To do so, it is sufficient to use weights equal to the inverse of the order of magnitude of the corresponding equations.

This criterion can be implemented in the following ways:

- The user directly writes the equations in an adimensionalized form.
- The user assigns the weights to be used by the numerical solver.
- The numerical solver evaluates the weights of Eq. (11).

Another approach is to consider whether vector, **x**, is sufficiently near the solution of the problem. Let us suppose we have the following linear system:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{13}$$

The distance of a point $\mathbf{x}_i$ from one of the planes, j:

$$a_{j1}x_1 + a_{j2}x_2 + \ldots + a_{jN}x_N = b_j \tag{14}$$

is determined by calculating the point at which the orthogonal line passing through $\mathbf{x}_i$ intersects the plane itself. The square of the distance between that point and $\mathbf{x}_i$ is:

$$d_j = \frac{\left[a_{j1}(x_1)_i + a_{j2}(x_2)_i + \ldots + a_{jN}(x_N)_i - b_i\right]^2}{\sum\limits_{m=1}^{N} a_{jm}^2} \tag{15}$$

By adopting the following weights:

$$w_j^2 = \frac{1}{\sum_{m=1}^{N} a_{jm}^2} \tag{16}$$

every term of summation (11) evaluated at point $\mathbf{x}_i$ represents the square of the distance between such a point and the planes of system (13). As far as NLSs are concerned, matrix $\mathbf{A}$ becomes an approximation of the Jacobian, $\mathbf{J}$. Since the Jacobian matrix changes with the iterations, the weights should also be modified. This strategy may be adopted to automatically evaluate the weights in Eq. (11).

Unfortunately, the aforementioned strategy does not benefit from the following property (Buzzi-Ferraris and Tronconi, 1993):

Given a merit function, $F(\mathbf{x})$, applied to a linear system, it is assumed that if $F(\mathbf{x}_{i+1}) < F(\mathbf{x}_i)$ then point $\mathbf{x}_{i+1}$ is closer to the solution than point $\mathbf{x}_i$.

The criteria described so far do not benefit from this property except for the linear system (13) consisting of orthogonal planes.

With reference to system (13), let us suppose we know the exact solution, $\mathbf{x}_s$, that makes the residuals null:

$$\mathbf{b} - \mathbf{A}\mathbf{x}_s = 0 \tag{17}$$

Given a point $\mathbf{x}_i$, other than $\mathbf{x}_s$, we have the residual:

$$\mathbf{b} - \mathbf{A}\mathbf{x}_i = \mathbf{f}_i \tag{18}$$

by subtracting Eq. (17) from Eq. (18) we obtain:

$$\mathbf{A}(\mathbf{x}_s - \mathbf{x}_i) = \mathbf{f}_i \tag{19}$$

Formally, the Euclidean norm of the distance $\mathbf{x}_i - \mathbf{x}_s$ is:

$$\|\mathbf{x}_i - \mathbf{x}_s\|_2 = \left\|\mathbf{A}^{-1}\mathbf{f}_i\right\|_2 \tag{20}$$

and the geometric interpretation of Eq. (20) is that the quantity $\|\mathbf{A}^{-1}\mathbf{f}_i\|_2$ measures the distance of $\mathbf{x}_i$ from the solution $\mathbf{x}_s$. With regards to NLSs, Eq. (20) is a measure of the distance of point $\mathbf{x}_i$ from the solution of the linearized system, where $\mathbf{A} = \mathbf{J}_i$ represents the Jacobian matrix evaluated using $\mathbf{x}_i$.

Finally, the distance of a new point $\mathbf{x}_{i+1}$ from the solution of the same system is:

$$\|\mathbf{x}_{i+1} - \mathbf{x}_s\|_2 = \left\|\mathbf{A}^{-1}\mathbf{f}_{i+1}\right\|_2 \tag{21}$$

Whenever a nonlinear system is concerned, the new point $\mathbf{x}_{i+1}$ must be accepted if

$$\left\|\mathbf{J}_i^{-1}\mathbf{f}_{i+1}\right\|_2 < \left\|\mathbf{J}_i^{-1}\mathbf{f}_i\right\|_2 \tag{22}$$

given that, in the case of linear systems, Eq. (22) means that $\mathbf{x}_{i+1}$ is closer to the solution than $\mathbf{x}_i$ is. It is worth highlighting that the Jacobian of Eq. (22) is kept constant while $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$ are the residuals at points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$.

If Newton's method is adopted to solve the NLS (see subsequent paragraphs) then the Jacobian matrix $\mathbf{J}_i$ has already been factored to solve the linear system produced by the method itself. The evaluation of the merit function using the two points, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, is therefore straightforward and manageable.

Often, besides normalizing the functions, it is also advisable to normalize the variables. A practical way to implement the normalization is to scale the variables by multiplying them for a coefficient so that all the variables have the same order of magnitude. By indicating with **D** a suitable diagonal matrix of multiplying coefficients, the proposed transformation is:

$$\mathbf{z} = \mathbf{D}\mathbf{x} \tag{23}$$

Consequently, the merit function with the new variables becomes:

$$\Phi_{wD}(\mathbf{z}) = \frac{1}{2}\mathbf{f}\left(\mathbf{D}^{-1}\mathbf{z}\right)^{\mathrm{T}}\mathbf{W}^2\mathbf{f}\left(\mathbf{D}^{-1}\mathbf{z}\right) \tag{24}$$

1.3
Substitution Methods

Before applying the substitution method to the solution of an NLS it is necessary to transform the equations into the following formulation:

$$\mathbf{h}(\mathbf{x}) = \mathbf{q}(\mathbf{x}) \tag{25}$$

where system $\mathbf{h}(\mathbf{x})$ should be easily solvable if the value of $\mathbf{q}(\mathbf{x})$ is known.

The method consists of applying the iterative formula:

$$\mathbf{h}(\mathbf{x}_{i+1}) = \mathbf{q}(\mathbf{x}_i) \tag{26}$$

where $\mathbf{x}_{i+1}$ is obtained from $\mathbf{x}_i$.

The easiest iterative formula is:

$$x_j = g_j\left(x_1, x_2, \ldots, x_{j-1}, x_{j+1}, \ldots, x_N\right) \tag{27}$$

where each variable is obtained explicitly from the corresponding function.

The procedure shown in (27) has the same shortcomings as the monodimensional case. Moreover, it is quite difficult to find a proper formulation that converges to the solution.

1.4
Gradient Method (Steepest Descent)

The gradient of a function is a vector. The function changes more rapidly in the direction of the gradient. With reference to the merit function (10), the gradient in $\mathbf{x}_i$ is given by:

$$\mathbf{g}(\mathbf{x}_i) = \mathbf{g}_i = \mathbf{J}_i^T\mathbf{f}_i \tag{28}$$

Consequently, vector

$$\mathbf{p}(\mathbf{x}_i) = \mathbf{p}_i = -\mathbf{g}_i = -\mathbf{J}_i^T\mathbf{f}_i \tag{29}$$

describes the direction where the merit function (10) decreases more rapidly. Obviously, the direction of the gradient changes whenever a different merit function is adopted. When the merit function (11) is involved, the search direction becomes:

$$\mathbf{p}_i = -\mathbf{g}_w = -\mathbf{J}_i^T\mathbf{W}_i^2\mathbf{f}_i \tag{30}$$

If the variables are also weighted and the merit function (24) is adopted, then the search direction becomes:

$$\mathbf{p}_i = -\mathbf{g}_{wD} = -\mathbf{J}_i^T\mathbf{W}_i^2\mathbf{f}_i\mathbf{d}_i^{-2} \tag{31}$$

The evaluation of the space increment, α_i, is performed by a monodimensional search. The procedures that adopt the gradient (steepest descent) method, as the search direction, have major limits if used alone. Actually, such methods are efficient only at the initial steps of the solving procedure.

The gradient method may be efficiently coupled with Newton's method since it is quite complementary to it. Newton's method is rather efficient in the final steps of the search while the gradient method is efficient in the initial ones.

1.5 Newton's Method

If the $\mathbf{f}_i$ of the NLS can be expanded in terms of a Taylor series:

$$\mathbf{f}(\mathbf{x}_i + \mathbf{d}_i) = \mathbf{f}_i + \mathbf{J}_i\mathbf{d}_i + O(\|\mathbf{d}_i\|^2) \tag{32}$$

and point $\mathbf{x}_i$ is rather close to the solution, it is possible to stop the expansion to the first order terms. In this case, the correction vector, $\mathbf{D}_i$, to be summed with point $\mathbf{x}_i$ comes from the solution of the system:

$$\mathbf{f}(\mathbf{x}_i + \mathbf{d}_i) \approx \mathbf{f}_i + \mathbf{J}_i\mathbf{d}_i = 0 \tag{33}$$

The following iterative procedure represents the elementary formulation of Newton's method:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{p}_i = \mathbf{x}_i + \mathbf{d}_i \tag{34}$$

where $\mathbf{d}_i$ comes from the solution of the linear system (33):

$$\mathbf{J}_i\mathbf{d}_i = -\mathbf{f}_i \tag{35}$$

Consequently, Newton's method has the following search direction:

$$\mathbf{p}_i = \mathbf{d}_i \tag{36}$$

and $\alpha_i = 1$

Whenever Newton's method converges, its convergence rate is quadratic.

It is possible to identify one difference between the solution of nonlinear systems and multidimensional optimization. Usually, the Jacobian matrix of system (35) is not symmetric. Thus, it is not possible to either solve the linear system with the Cholesky algorithm or to halve the memory allocation. The most efficient methods adopted for the Jacobian factorization require twice as much time as the Cholesky algorithm.

The correction, $\mathbf{d}_i$, obtained from system (35) is independent from either a change of scale in the variables or the merit function. In fact, by introducing the scale change:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{c} \tag{37}$$

the new Jacobian, with respect to the variables $\mathbf{y}$, becomes:

$$\mathbf{J}_y = \mathbf{J}_x\mathbf{C}^{-1} \tag{38}$$

and the Newton's method estimate for the $\mathbf{x}$ variables is:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{J}_x^{-1}\mathbf{f}_i \tag{39}$$

while for the new $\mathbf{y}$ variables is:

$$\mathbf{y}_{i+1} = \mathbf{y}_i - \mathbf{J}_y^{-1}\mathbf{f}_i = \mathbf{C}\mathbf{x}_i + \mathbf{c} - \mathbf{C}\mathbf{J}_x^{-1}\mathbf{f}_i = \mathbf{C}\left(\mathbf{x}_i - \mathbf{J}_x^{-1}\mathbf{f}_i\right) + \mathbf{c} = \mathbf{C}\mathbf{x}_{i+1} + \mathbf{c} \tag{40}$$

As a result, the Newton's method estimate is invariant with respect to a linear transformation using the variables $\mathbf{x}$ as well as the merit function (22).

Vector $\mathbf{d}_i$ represents a direction where all the merit functions decrease. Actually:

$$\mathbf{d}_i = -\mathbf{J}_i^{-1}\mathbf{f}_i \tag{41}$$

$$\mathbf{g}_i^{\mathrm{T}}\mathbf{d}_i = \left(\mathbf{J}_i^{\mathrm{T}}\mathbf{f}_i\right)^{\mathrm{T}}\left(-\mathbf{J}_i^{-1}\mathbf{f}_i\right) = -\mathbf{f}_i^{\mathrm{T}}\mathbf{J}_i\mathbf{J}_i^{-1}\mathbf{f}_i = -\mathbf{f}_i^{\mathrm{T}}\mathbf{f}_i < 0 \tag{42}$$

$$\mathbf{g}_w^{\mathrm{T}}\mathbf{d}_i = \left(\mathbf{J}_i^{\mathrm{T}}\mathbf{W}_i^2\mathbf{f}_i\right)^{\mathrm{T}}\left(-\mathbf{J}_i^{-1}\mathbf{f}_i\right) = -\mathbf{f}_i^{\mathrm{T}}\mathbf{W}_i^2\mathbf{J}_i\mathbf{J}_i^{-1}\mathbf{f}_i = -\mathbf{f}_i^{\mathrm{T}}\mathbf{W}_i^2\mathbf{f}_i < 0 \tag{43}$$

$$\mathbf{g}_{wD}^{\mathrm{T}}\mathbf{d}_i = \left(\mathbf{J}_i^{\mathrm{T}}\mathbf{W}_i^2\mathbf{f}_i\mathbf{D}_i^{-2}\right)^{\mathrm{T}}\left(-\mathbf{J}_i^{-1}\mathbf{f}_i\right) = -\mathbf{D}_i^{-2}\mathbf{f}_i^{\mathrm{T}}\mathbf{W}_i^2\mathbf{f}_i < 0 \tag{44}$$

Such a property is valid if the following two conditions are satisfied:

1. the Jacobian matrix is not singular, i.e., the inverse matrix must exist;
2. the Jacobian in Eq. (41) must be a good approximation of the true Jacobian matrix and not a generic matrix $\mathbf{B}_i$, otherwise:

$$\mathbf{J}_i\mathbf{B}_i^{-1} \neq \mathbf{I} \tag{45}$$

and the previous Eqs. (42–44) may not be true.

It is also possible to outline another difference between the solution of nonlinear systems and multidimensional optimization.

As far as multidimensional optimization problems are concerned, matrix $\mathbf{B}_i$ may also be a bad approximation of the Hessian (provided it is positive and definite) and at the same time be able to guarantee a reduction of the merit function. Conversely, matrix $\mathbf{B}_i$ involved in the solution of NLSs should be a good estimate of the Jacobian.

Besides the previously mentioned advantages, Newton's method also presents some disadvantages that suggest not using it with the trivial iterative formulation of Eqs. (34) and (35).

Three different categories for the classification of the problems related to Newton's method can be outlined:

1. Problems related to the Jacobian matrix
   - The method undergoes a critical point if the Jacobian is either singular or ill-conditioned.
2. Problems related to the convergence of the method
   - The method may not converge to the solution;
   - The new prediction may be worse than the previous one with respect to all merit functions.
3. Problems related to the Jacobian evaluation and the linear system solution
   - Every new iteration requires the evaluation of the Jacobian matrix. If the Jacobian is evaluated numerically, this means that the nonlinear system (1) is called N times;
   - Each new iteration requires the solution of the linear system (35).

The algorithms derived by the original Newton's method may be divided into two classes depending on how the Jacobian matrix is evaluated.

The first class comprises Newton's modified methods where the Jacobian is evaluated analytically or numerically approximated at point $\mathbf{x}_i$.

The second class comprises the quasi-Newton methods that update the Jacobian by means of the information gathered during the iterative process.

For both classes, as soon as the Jacobian matrix has been either evaluated or updated, it is recommended to immediately execute a Newton's iteration in order to exploit the efficiency of such a method.

Consequently, both of the aforementioned classes first verify the point:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{d}_i = \mathbf{x}_i - \mathbf{J}_i^{-1}\mathbf{f}_i \tag{46}$$

Such a point is accepted if it satisfies at least one of the following tests:

$$\left\|\mathbf{J}_i^{-1}\mathbf{f}(\mathbf{x}_{i+1})\right\|_2 < \left\|\mathbf{J}_i^{-1}\mathbf{f}(\mathbf{x}_i)\right\|_2 (1-\gamma) \tag{47}$$

$$\mathbf{f}_{i+1}^{\mathrm{T}}\mathbf{W}^2\mathbf{f}_{i+1} < \mathbf{f}_i^{\mathrm{T}}\mathbf{W}^2\mathbf{f}_i(1-\gamma) \tag{48}$$

The γ parameter guarantees a satisfactory improvement of the merit functions.

1.6 Modified Newton's Methods

In these methods the Jacobian matrix is evaluated analytically or is approximated numerically. Since the Jacobian is recalculated at each iteration it is also necessary to solve the linear system (35).

Several expedients and precautions are necessary to reduce the drawbacks of Newton's method.

1.6.1 Singular or Ill-conditioned Jacobian Matrix

The solution of system (35) is performed through the factorization of the Jacobian matrix. Whichever factorization is adopted, it is mandatory to evaluate the condition number of the system and to properly operate if such a number is too high.

Actually, it is possible to introduce the auxiliary function:

$$M(\mathbf{x}_i + \mathbf{d}_i) = \frac{1}{2}(\mathbf{J}_i\mathbf{d}_i + \mathbf{f}_i)^{\mathrm{T}}(\mathbf{J}_i\mathbf{d}_i + \mathbf{f}_i) \tag{49}$$

The minimum of function (49) is:

$$\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i\mathbf{d}_i = -\mathbf{J}_i^{\mathrm{T}}\mathbf{f}_i \tag{50}$$

which is equivalent to the prediction of Newton's method (35) applied to the solution of the nonlinear system.

If the Jacobian is well-conditioned then the correction, $\mathbf{D}_i$, is achieved by solving system (35) instead of system (50). Conversely, the use of function (49) becomes interesting when the Jacobian matrix is quite ill-conditioned or even singular. As a matter of fact, the system matrix, $\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i$, is the Hessian of function (49) and it is symmetric. By using function (49) instead of the merit function (10), there is the advantage of knowing the Hessian without having to evaluate the second derivatives. At the same time, it is possible to apply to matrix $\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i$, all the expedients exploited when an ill-conditioned minimum problem is involved.

Two algorithms implement the aforementioned idea:

- The Levenberg-Marquardt method modifies system (50) in the following way:

$$(\mathbf{J}_i^{\mathrm{T}}\mathbf{J}_i + \mu_i\mathbf{I})\mathbf{d}_i = -\mathbf{J}_i^{\mathrm{T}}\mathbf{f}_i \tag{51}$$

The μ parameter may be chosen so to transform matrix $(\mathbf{J}^T{}_i\mathbf{J}_i + \mu_i\mathbf{I})$ in a well-conditioned matrix. Besides being an artifice to reduce the ill-conditioning of the Jacobian, the Levenberg-Marquardt method is also an algorithm that couples Newton's method to the gradient one. Actually, if the Jacobian is not singular, the solution of system (51), with $\mu = 0$, is equivalent to Newton's estimate. Conversely, when high values of parameter μ are involved, the search direction tends to the gradient of the merit function (10).

The Gill-Murray criterion represents the second alternative. The idea is to make the diagonal coefficients of matrix $\mathbf{J}_i^T\mathbf{J}_i$ positive. If the Jacobian $\mathbf{J}$ is **QR** factored and matrix **R** is worked out in order to avoid any zeros in the main diagonal, then matrix $\mathbf{J}^T\mathbf{J} = \mathbf{R}^T\mathbf{R}$ is symmetric positive definite.

Buzzi-Ferraris and Tronconi (1986) showed a new methodology for the modification of the Jacobian matrix. If the Jacobian is ill-conditioned or singular then some equations in system (35) are linearly dependent. Consequently, it is possible to eliminate those linearly dependent rows. Since the resulting system becomes underdimensioned, it is appropriate to adopt the **LQ** factorization that produces the solution with minimum Euclidean norm for vector $\mathbf{d}_i$. Thus, it is possible to avoid an excessively large correction on such a vector. The numerical solution satisfies not only the subsystem but also the equations that were removed, since, if compatible, they are almost a linear combination of the others. This criterion is often efficient and is preferable to the previous one since it is not influenced by the merit function. At the same time, it does not produce a false solution. By the term false we mean a solution of the minimum problem that is not the solution of the NLS.

1.6.2 The Convergence Problem

As mentioned, the Newton's estimate is not satisfactory whenever point $\mathbf{x}_{i+1}$ does not meet conditions (47) and (48). In such a case, it is possible to adopt the following strategies:

- A monodimensional search is performed in the same direction as the Newton's one.
- The region where the functions are linearized is reduced.
- An alternative algorithm to the Newton's method is adopted.

Before addressing these points, the following feature should be emphasized: an NLS may not have a solution. Moreover, if a solution exists, we are not sure that it will be possible to determine it. A numerical program should warn the user about its incapability of solving the problem.

1.6.2.1 Monodimensional Search

Usually, since the monodimensional optimization is both not time-consuming and quite efficient, it is adopted in all-purpose solvers. Normally, the monodimensional search algorithm is not pushed to the extreme. Actually, the optimization is intended to identify a new point where Newton's method might easily converge. The merit function that is usually adopted is the weighted one (12). At the outset, the following data are known:

- the value of Φ_w at point $\mathbf{x}_i$;
- the gradient $\mathbf{g}_w$ at point $\mathbf{x}_i$ and that in the direction $\mathbf{d}_i$, $\mathbf{g}_w^T\mathbf{d}_i$;
- the value of Φ_w at point $\mathbf{x}_i + \mathbf{d}_i$.

Since there are three data in the direction $\mathbf{d}_i$, the merit function Φ_w may be approximated by the parabola:

$$y(t) = y(0) + ty'(0) + t^2\left[y(1) - y(0) - y'(0)\right] \tag{52}$$

The minimum of the parabola is:

$$u = \frac{-y'(0)}{2\left[y(1) - y(0) - y'(0)\right]} \tag{53}$$

In the following, we will assume to have a good estimate of the Jacobian matrix. Consequently, the following equation may be adopted:

$$y'(0) = \mathbf{g}_w^T\mathbf{d}_i = -\mathbf{f}_i^T\mathbf{W}_i^2\mathbf{f}_i \tag{54}$$

and the minimum of the parabola becomes:

$$u = \frac{\mathbf{f}_i^T\mathbf{W}_i^2\mathbf{f}_i}{\mathbf{f}_{i+1}^T\mathbf{W}_i^2\mathbf{f}_{i+1} + \mathbf{f}_i^T\mathbf{W}_i^2\mathbf{f}_i} \tag{55}$$

It is recommended to check that u is not too small by imposing:

$$u > 0.1 \tag{56}$$

Since point $\mathbf{x}_{i+1}$ does not satisfy Eq. (47), an upper limit for u is automatically set.

If the minimization is not successful then a new artifice should be exploited, otherwise the program must stop with a warning.

1.6.2.2
Reduction of the Search Zone

The Levenberg-Marquardt method may be considered from three distinct perspectives:

- as an artifice to avoid the ill-conditioned problem of the Jacobian matrix;
- as an algorithm that couples Newton's method to the gradient one;
- as a method exploiting either a reduced step or a confidence region.

The third point is the most interesting when a reduction of the search zone is concerned. In this case, it is required to identify the correction $\mathbf{d}_i$ that minimizes the auxiliary function (49) with the constraint:

$$\|\mathbf{d}_i\|_2 = \delta \tag{57}$$

where δ has a specified value.

A valid alternative to the Levenberg-Marquardt method is represented by the dog leg method, also known as Powell's hybrid method (1970). Once again, such a method couples Newton's method to the gradient one. The original version of Powell's method was close to the concept of either confidence region or reduced step. Powell proposed a strategy for the modification of parameter δ subject to both the successes and failures of the procedure.

1.6.2.3
Alternative Methods

Whenever Newton's method fails, it is necessary to switch to an alternative method. For this reason, the most commonly used method is the gradient of a merit function.

There are several alternatives. It is possible to perform a monodimensional search along the gradient direction. Even better, the two methods may be coupled as it happens with the Levenberg-Marquardt algorithm or the dog leg method. Another choice is to perform a bidirectional optimization on the plane defined by both search directions.

Unfortunately, there are no heuristic methods for the solution of NLSs. Only for very specific problems can a substitution method be expressly tailored and coupled to Newton's method. As an example, in the field of chemical engineering, the boiling point (BP) method may be implemented and applied to distillation columns. In the following, we will introduce the continuation methods. Such methods transform the functions of the NLS and solve an equivalent and dynamically easier problem.

1.7
Quasi-Newton Methods

Let $\mathbf{B}_i$ be an approximation of the Jacobian matrix at point $\mathbf{x}_i$. As mentioned before, matrix $\mathbf{B}_i$ must be a good approximation of the Jacobian. Consequently, also in the case of quasi-Newton methods, it is necessary to evaluate either analytically or numerically the Jacobian matrix. If during the search of the solution the rate of convergence should decrease, reevaluating the Jacobian is recommended. Therefore, it is not possible to implement a quasi-Newton method without a modified Newton's method.

During the solution procedure the values of functions $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$ are known in the points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Such points must not necessarily correspond to previous Newton's method estimates. Given:

$$\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i \tag{58}$$

if the distance between the two points is not significant, it is possible to link the function values $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$ through a Taylor expansion:

$$\mathbf{f}_{i+1} = \mathbf{f}_i + \mathbf{B}\Delta\mathbf{x}_i \tag{59}$$

where the Jacobian, **B**, is evaluated in a suitable point between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. Specifically, it is possible to impose that the Jacobian satisfies the following condition:

$$\mathbf{f}_{i+1} = \mathbf{f}_i + \mathbf{B}_{i+1}\Delta\mathbf{x}_i \tag{60}$$

Equation (60) does not allow univocal evaluation of all components of the Jacobian when the number of equations $N > 1$. In this case, $N - 1$ more conditions are necessary. In 1965 Broyden proposed to choose the conditions to be added to Eq. (60) in order to keep invariant the product between the Jacobian, evaluated in $\mathbf{x}_i$ and in $\mathbf{x}_{i+1}$, and an orthogonal vector to Δx_i. Generally, for any given vector, $\mathbf{q}_i$, *with:*

$$\mathbf{q}_i^T \Delta\mathbf{x}_i = 0 \tag{61}$$

it must result that:

$$\mathbf{B}_{i+1}\mathbf{q}_i = \mathbf{B}_i\mathbf{q}_i \tag{62}$$

This condition is reasonable if we consider Eq. (59). Actually, it is possible to modify the Jacobian in the direction Δx_i so as to satisfy condition (60). On the contrary, in a direction orthogonal to the previous one, there is no additional information and the behavior of the Jacobian, with respect to a Taylor expansion in that direction, should be invariant.

By coupling conditions (62) and (60), it is possible to univocally identify the Jacobian in $\mathbf{x}_{i+1}$:

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{(\mathbf{f}_{i+1} - \mathbf{f}_i - \mathbf{B}_i\Delta\mathbf{x}_i)\Delta\mathbf{x}_i^T}{\Delta\mathbf{x}_i^T\Delta\mathbf{x}_i} \tag{63}$$

1.8 Large and Sparse Systems

When the number of equations and variables is quite large, often each equation depends on a reduced set of variables. As far as the Newton's and quasi-Newton methods are concerned, it is necessary to exploit the sparsity of the Jacobian matrix so as to reduce the memory allocation while saving CPU time. In particular, the following expedients are essential:

- The solution of system (35) should be made by the method that best exploits the Jacobian sparsity and structure.
- If the Jacobian has no specific structure that can be directly exploited, it is worthwhile rearranging both the variables and equations so as to reduce the CPU effort and memory allocation required by the factorization of the Jacobian matrix.
- The null Jacobian components should not be evaluated. This happens automatically if the Jacobian is evaluated analytically. Conversely, whenever the Jacobian matrix is approximated numerically, the following computations:

$$J_{ik} = \frac{f_j(\mathbf{x}_i + h_k \mathbf{e}_k) - f_j(\mathbf{x}_i)}{h_k} \tag{64}$$

should be avoided if it is *a priori* known that: $f_j(\mathbf{x}_i + h_k\mathbf{e}_k) = f_j(\mathbf{x}_i)$.

- It is possible to exploit some formulas to update the Jacobian, which are able to preserve its sparsity. At the same time, if some elements are constant, they should not be updated by those formulas. Schubert (1970) proposed a modification of the Broyden formula (1965), while Buzzi-Ferraris and Mazzotti (1984) proposed a modification of the Barnes formula (1965). These formulas take into account the coefficients that are known and do not modify them. The update is performed only on the coefficients that are unknown.
- If the Jacobian is evaluated numerically, it is not convenient to increment a variable one at a time and to perform a call to the nonlinear system. This point must be emphasized. If Eq. (64) is adopted to evaluate a Jacobian matrix that is supposed to be full, then vector $\mathbf{e}_k$ is the null array except for position k, where the element is equal to 1. In this case, system (1) is called N times to evaluate the derivatives of the functions with respect to the N variables. Let us now consider the following sparse Jacobian matrix, where the symbol × represents a nonzero element (see Fig. 3.1).

It can be observed that when the system is called to evaluate the derivatives with respect to variable x_1, the only functions to be modified are f_1 and f_8. If at the same time variable x_2 were modified, it could be possible to evaluate the derivatives with respect to this variable since it only influences functions f_2 and f_7. Going on with the reasoning, it is possible to show that only three calls to the system of Fig. 3.1 are sufficient to evaluate the whole Jacobian matrix. In fact, with the first call it is possible to increment variables $x_1 x_2 x_3 x_4 x_6 x_9$. With the second call we increment variables $x_5 x_8$. Finally, with the third call we increment variables $x_7 x_{10}$. When the system is sparse, the total number of calls necessary for the evaluation of the Jacobian matrix can be drastically reduced. It is not easy to identify the sequence of variable groupings that minimizes the number of calls to the nonlinear system. Curtis, Powell and Reid (1972) proposed a heuristic algorithm that is often optimal and can be easily described. We start with the first variable and identify the functions that depend on it. We then check if the second variable does not interfere with the functions with which the first variable interacts. If this happens, we go on to the third variable. Any new variable introduced in the sequence also increases the number of functions involved. When no additional variables can be added to the list, this means that the first group has been identified and we can go on with the next group until all N variables of the system have been collected. It is evident that the matrix structure of the Jacobian must be known for this procedure to be applied. This means that the user must identify the Boolean of the Jacobian, i.e., the matrix that contains the dependencies of each function from the system variables (see Fig. 3.1).

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
f1	×									×
f2		×								×
f3			×				×			
f4				×	×					×
f5					×		×			
f6						×		×		
f7		×					×	×		
f8	×							×		
f9									×	
f10			×							×

Figure 3.1 The Boolean matrix describes the Jacobian structure and the function dependency from the variables of the nonlinear system.

1.9 Stop Criteria

When the problem is supposed to be solved or when there are insurmountable problems, there are some tests to bring the iterations to an end:

- It is advisable to implement a limitation on the maximum number of iterations.
- If the weighted function (12) is lower than an assigned value, there is a good chance a solution has been reached.
- The procedure is stopped if the estimate of Newton's method, $\mathbf{d}_i$, has all components reasonably small. With multidimensional optimization it is not sufficient to check whether a norm of vector $\mathbf{d}_i$ is lower than an assigned value. On the contrary, it is advisable also to to check the following relative:

$$\max_{1 \le i \le N} \left| \frac{d_i}{\max\{x_i, z_i\}} \right| \le \varepsilon \tag{65}$$

Even if a quasi-Newton method is used, a good approximation of the Jacobian is known. Consequently, this criterion is adequately reliable. Nonetheless, it should be emphasized that this test is correct only when the difference between two consecutive iterations, $\mathbf{d}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, comes from a Newton-like method and the Jacobian is not singular.

1.10 Bounds, Constraints, and Discontinuities

Some problems have solutions that are not acceptable since they belong to unfeasible regions. In these situations, it can be worthy assigning some bounds to the variables in order to avoid the solution from falling in those unfeasible regions. This issue requires the adoption of specifically tailored numerical algorithms that are able to depart from the unfeasible attractor while moving towards the feasible region. Similar considerations apply when discontinuities are involved. In this case, the numerical algorithm should be able to work across the discontinuity while avoiding a crisis due to its presence. The discontinuity can be either in the function itself or in its derivatives (Sacham and Brauner 2002).

1.11 Continuation Methods

Let us suppose we have a nonlinear system of N equations whose solution is quite difficult. For some reason that will be explained in the following we suppose we have a vector of adjoint parameters $\mathbf{z}$, of M elements, in the equations of the system. Therefore, the NLS can be rewritten as:

$$\mathbf{f}(\mathbf{x}, \mathbf{z}) = 0 \tag{66}$$

The system must be solved with respect to the N unknowns, $\mathbf{x}$, given a specific value, $\mathbf{z} = \mathbf{z}_F$, of the parameter vector.

Let us now suppose we know another system, $\mathbf{q}(\mathbf{x}, \mathbf{z}) = \mathbf{0}$, in some way related to the previous one, whose solution, for a given value of parameters $\mathbf{z}$, is quite easy. In this case it is possible to write a new system that is a linear combination of the previous two:

$$\mathbf{h}(\mathbf{x}, \mathbf{z}, t) = t\mathbf{f}(\mathbf{x}, \mathbf{z}) + (1 - t)\mathbf{q}(\mathbf{x}, \mathbf{z}) = 0 \tag{67}$$

The parameter t in Eq. (67) is called the homotopy parameter. When the parameter t varies in the interval 0, ... 1, system $\mathbf{h}$ is solved for a value of $\mathbf{x}$ that satisfies both systems $\mathbf{q}$ and $\mathbf{f}$. The parameters $\mathbf{z}$ can be a function of parameter t in any way, provided that for $t = 1$ we have $\mathbf{z} = \mathbf{z}_F$. The most straightforward functional dependency between t and $\mathbf{z}$ is the linear one:

$$\mathbf{z} = \mathbf{z}_0 + (\mathbf{z}_F - \mathbf{z}_0)t \tag{68}$$

where $\mathbf{z}_0$ corresponds to the initial value of the parameters.

Another functional dependency is the following one:

$$z_i = z_{0i}\left(\frac{z_{Fi}}{z_{0i}}\right)^t \tag{69}$$

There are several alternatives for the auxiliary system $\mathbf{q}(\mathbf{x}, \mathbf{z})$. The common characteristic is that for $t = 0$ the solution of system $\mathbf{q}(\mathbf{x}_0, \mathbf{z}_0) = \mathbf{0}$ should be effortless.

The following are some choices:

- Fixed point homotopy: $\mathbf{q}(\mathbf{x}, \mathbf{z}) = \mathbf{x} - \mathbf{x}_0 + \mathbf{z} - \mathbf{z}_0$

$$\mathbf{h}(\mathbf{x},\mathbf{z},t) = t\mathbf{f}(\mathbf{x},\mathbf{z}) + (1-t)\left[(\mathbf{x}-\mathbf{x}_0) + (\mathbf{z}-\mathbf{z}_0)\right] = \mathbf{0} \tag{70}$$

- Newton or global homotopy: $\mathbf{q}(\mathbf{x},\mathbf{z}) = \mathbf{f}(\mathbf{x},\mathbf{z}) - \mathbf{f}(\mathbf{x}_0,\mathbf{z}_0)$

$$\mathbf{h}(\mathbf{x},\mathbf{z},t) = \mathbf{f}(\mathbf{x},\mathbf{z}) - (1-t)\mathbf{f}(\mathbf{x}_0,\mathbf{z}_0) = \mathbf{0} \tag{71}$$

- Homotopy with scale invariance: $\mathbf{q}(\mathbf{x},\mathbf{z}) = \mathbf{J}(\mathbf{x}_0,\mathbf{z}_0)\left[(\mathbf{x}-\mathbf{x}_0) + (\mathbf{z}-\mathbf{z}_0)\right]$

$$\mathbf{h}(\mathbf{x},\mathbf{z},t) = t\mathbf{f}(\mathbf{x},\mathbf{z}) + (1-t)\mathbf{J}(\mathbf{x}_0,\mathbf{z}_0)\left[(\mathbf{x}-\mathbf{x}_0) + (\mathbf{z}-\mathbf{z}_0)\right] = \mathbf{0} \tag{72}$$

- Parametric continuation method: $\mathbf{q}(\mathbf{x},\mathbf{z}) = \mathbf{0}$

$$\mathbf{h}(\mathbf{x},\mathbf{z},t) = \mathbf{f}(\mathbf{x},\mathbf{z}) = \mathbf{0} \tag{73}$$

The fourth criterion (73) deserves some explanation since one could think that the original problem has not been modified. In many practical cases, a problem may have a simple solution in correspondence to a value $\mathbf{z}_0$ of the parameters, while there are numerical difficulties for $\mathbf{z}_F$. In this case the system is solved by setting $t = 0$ and $\mathbf{z} = \mathbf{z}_0$. We then get a first solution $\mathbf{x}_0$ that satisfies:

$$\mathbf{h}(\mathbf{x}_0,\mathbf{z}_0,0) = \mathbf{f}(\mathbf{x}_0,\mathbf{z}_0) = \mathbf{0} \tag{74}$$

Successively, we change t from 0 to 1 in order to modify continuously the parameters from $\mathbf{z}_0$ to $\mathbf{z}_F$. By doing so, several intermediate problems are solved through a step-by-step procedure.

It is worth highlighting two cases:

1. The parameters, $\mathbf{z}$, correspond to some specifications that should be satisfied. Often, the problem can be easily solved if the specifications are mild, while it becomes hard when the requirements are tight. A typical example is represented by a distillation column. The continuation parameter can be the product purity. If the product purity is quite high, there can be some problems concerning the numerical solution. In this case, it is recommended to start with a lax specification. Once the solution has been evaluated, the problem is slightly modified by tightening the specification. A new solution is performed by adopting as a first guess the previous solution. By *continuing* the procedure, it is possible to reach the final product purity.
2. The problem can be solved easily by introducing some simplifications. The continuation method modifies continuously the simplified hypotheses, carrying the system towards the detailed model. In the case of separation units, one of the problems may be the evaluation of the liquid-vapor equilibrium constants, **k**. If the **k** vector strongly depends on the compositions then it can be difficult to identify the first guess values that make the Newton's method converge. In this case, it is convenient to consider the system ideal. By solving the simplified problem under the hypothesis of ideal **k** values, a solution is easily obtained. Such a solution becomes the first guess for the continuation procedure that takes the system towards the hypothesis of nonideal liquid-vapor equilibria. The parameter vector **z** comprises the **k** values as follows:

$$k = k_{id}\left(\frac{k}{k_{id}}\right)^t \tag{75}$$

Initially, when $t = 0$, all parameters are equal to the ideal **k** values. The homotopy parameter, t, evolves from 0 to 1. By doing so the **k** values continuously change from the ideal to the real hypotheses. The same reasoning can be applied to the enthalpies of the mixture.

The main advantage obtained by the continuation method is that the intermediate problems have a physical implication. Consequently, each intermediate problem has a solution where the variables take up reasonable values.

Another approach to the solution of the continuation problem is to implement an ODE system that integrates the **x** variables and the **z** parameters from an initial condition (easy problem) to a final *time* (difficult problem). Seader and coauthors (Kuno and Seader 1988; Seader et al. 1990; Jalali and Seader 1999; Gritton et al. 2001) have worked extensively on this approach.

References

1 *Barnes J. C. P.* An Algorithm for Solving Nonlinear Equations Based on the Secant Method. Comput. J. 8 (1965) p. 66

2 *Broyden C. G.* A Class of Methods for Solving Nonlinear Simultaneous Equations. Math. Comput. 21 (1965) p. 368

3 *Broyden C. G.* A New Method of Solving Nonlinear Simultaneous Equations. Comput. J. 12 (1969) p. 94

4 *Buzzi-Ferraris G. Mazzotti M.* Orthogonal Procedure for the Updating of Sparse Jacobian Matrices. Comput. Chem. Eng. 8 (1984) p. 389

5 *Gritton K. S. Seader J. D. Lin W. J.* Global Homotopy Continuation Procedures for Seeking all Roots of a Nonlinear Equation. Comput. Chem. Eng. 25 (2001) p. 1003

6 *Buzzi-Ferraris G. Tronconi E.* BUNLSI – A Fortran Program for Solution of Systems of Nonlinear Algebraic Equations. Comput. Chem. Eng. 10 (1986) p. 129

7 *Buzzi-Ferraris G. Tronconi E.* An Improved Convergence Criterion in the Solution of Nonlinear Algebraic Equations. Comput. Chem. Eng. 17 (1993) p. 419

8 *Curtis A. R. Powell M. J. D. Reid J. K.* On estimation of Sparse Jacobian Matrices. Report TP476 AERE Harwell 1972

9 *Jalali F. Seader J. D.* Homotopy Continuation Method in Multi-phase Multi-reaction Equilibrium Systems. Comput. Chem. Eng. 23 (1999) p. 1319

10 *Powell M. J. D.* A Hybrid Method for Nonlinear Equations. In: Numerical Methods for Nonlinear Algebraic Equations. Ed.: Rabinowitz G. Breach London, 1970

11 *Schubert, L. K.* Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian. Math. Comput. 24 (1970) p. 27

Further Reading

1 *Allgower E. L. Georg K.* Homotopy Method of Approximating Several Solutions to Nonlinear Systems of Equations. In: Numerical Solution of Highly Nonlinear Problems, Foster, Amsterdam, 1980

2 *Davidenko D.* On a New Method of Numerically Integrating a System of Nonlinear Equations, Doklady Akademii Nauk USSR 88 (1953) p. 601

3 *Eberhart J. G.* Solving Equations by Successive Substitution – The Problems of Divergence and Slow Convergence. J. Chem. Edu. 63 (1986) p. 576

4 *Gupta Y. P.* Bracketing Method for On-line Solution of Low-Dimensional Nonlinear Algebraic Equations. Ind. Eng. Chem. Res. 15 (1995) p. 239

5 *Jalali F.* Process Simulation Using Continuation Method in Complex Domain. Comput. Chem. Eng. 22 (1998) p. S943

6 *Jsun Y. W.* Multiple-step Method for Solving Nonlinear Systems of Equations. Comput. Appl. Eng. Edu. 5 (1997) p. 121

7 *Karr C. L. Weck B. Freeman L. M.* Solutions to Systems of Nonlinear Equations via a Genetic Algorithm. Eng. App. Artif. Intell. 11 (1998) p. 369

8 *Kuno M. Seader J. D.* Computing all Real Solutions to Systems of Nonlinear Equations with a Global Fixed-Point Homotopy. Ind. Eng. Chem. Res. 27 (1988) p. 1320

9 *Neumaier A.* Interval Methods for Systems of Equations. Cambridge University Press, Cambridge 1990

10 *Paloschi J. R.* Bounded Homotopies to Solve Systems of Algebraic Nonlinear Equations. Comput. Chem. Eng. 19 (1995) p. 1243

11 *Paterson W. R.* A New Method for Solving a Class of Nonlinear Equations. Chem. Eng. Sci. 41 (1986) p. 135

12 *Rice R. J.* Numerical Methods, Software and Analysis. Academic Press, London, 1993

13 *Seader J. D. Kuno M. Lin W. J. Johnson S. A. Unsworth K. Wiskin J. W.* Mapped Continuation Methods for Computing all Solutions to General Systems of Nonlinear Equations. Comput. Chem. Eng. 14 (1990) p. 71

14 *Shacham M. Kehat E.* Converging Interval Methods for the Iterative Solution of a Nonlinear Equation. Chem. Eng. Sci. 28 (1973) p. 2187

15 *Shacham M. Brauner N.* Numerical Solution of Non-linear Algebraic Equations with Discontinuities. Comput. Chem. Eng. 26 (2002) p. 1449

16 *Shacham M. Brauner N. Cutlip M. B.* A Web-based Library for Testing Performance of Numerical Software for Solving Nonlinear Algebraic Equations. Comput. Chem. Eng. 26 (2002) p. 547

17 *Schnepper C. A. Stadtherr M. A.* Application of Parallel Interval Newton/Generalized Bisection Algorithm to Equation-Based Chemical Process Flowsheeting. Interval Comput. 4 (1993) p. 40

18 *Sundar S. Bhagavan B. K. Prasad S.* Newton-preconditioned Krylov Subspace Solvers for System of Nonlinear Equations: a Numerical Experiment. Appl. Math. Lett. 14 (2001) p. 195

19 *Wayburn T. L. Seader J. D.* Homotopy Continuation Methods for Computer-Aided Process Design. Comput. Chem. Eng. 11 (1987) p. 7

20 *Wilhelm C. E. Swaney R. E.* Robust Solution of Algebraic Process Modeling Equations. Comput. Chem. Eng 18 (1994) p. 511