A Comparative Analysis of Numerical Methods for Solving Systems of Nonlinear Algebraic Equations

Atle C. Christiansen, John Morud and Sigurd Skogestad
Departement of Chemical Engineering
Norwegian University of Science and Technology
N–7034 Trondheim Norway

Abstract

This paper compares two numerical methods for finding solutions to a system of non–linear algebraic equations (NAE's). We consider *homotopy–continuation* methods and discuss inherent difficulties in using such methods. To prevent potential unboundedness of the homotopy paths we provide some insight into how appropriate *branch–jumping* techniques may be applied. We also present a novel *tear and grid method* based on conventional techniques of partitioning and precedence ordering, with the addition of including a grid of the *tear* variables. Both methods may be used to obtain initial solutions as well as exploring solutions in the parameter space. A comparative analysis of the methods is presented in terms of a few example problems. For simple models consisting of a relatively small number of equations, we find that the grid method offers potential savings in both computer time and implementation effort. However, the perhaps most appealing feature of the tear and grid method lies in the convenient *visualization* of the solution space.

1 Introduction

During the last decades an extensive range of computer–aided methods for both steady state and dynamic simulation has evolved. Driven by an increase in both computer availiability, computing power and efficiency, computer aided tools are now a standard feature in almost all aspects of chemical process engineering. A recent review of availiable numerical methods for process design, optimization and control with emphasis on non–linear analysis is given by Seider and Brengel (1991). However, in this work we limit ourselves to the study of numerical methods for steady state analysis, i.e. procedures for finding one or all solutions to a system of non–linear algebraic equations (NAE's) which we denote by

$$\begin{aligned} f(x,\lambda) &= 0 \\ f : \mathcal{R}^m \times \mathcal{R}^k \to \mathcal{R}^m, &\quad x \in \mathcal{R}^m, \quad \lambda \in \mathcal{R}^k \end{aligned} \tag{1}$$

where x is a m-dimensional vector of state variables and λ a k-dimensional vector of parameters. In an excellent review of numerical methods for solving NAE's that were available at the time the review was written, Sargent (1981) claims that *"there is no method which clearly stands out from all the rest in terms of both reliability and efficiency"*. Even though we today face a bewildering range of methods, and proposals for new methods appear quite frequently in the literature, Sargent's analysis still seems to hold in the general sense. Still, in the future it is to be expected that the engineer will require specific solution methods designed to deal with the particular problem at hand. Being able to choose the *"optimal"* method from a library of different algorithms according to some predefined objective is thus a great advantage. We argue that such objectives for choosing the appropriate numerical methods should be formulated according to the size, complexity and difficulty of the problem at hand, rather than rigour or ability to handle any pathological problem. A rule of thumb should be to avoid shooting sparrows with canons.

Local versus global methods. It is common to distinguish between *local* and *global* methods, depending on the respective domains of attraction for convergence. Among the most common local methods used in solving chemical engineering problem we find the Newton, Quasi–Newton, Secant, Broyden and Deflation methods. However appealing in terms of simplicity and flexibility, local methods usually display poor convergence properties unless good starting guesses are provided, due to dependancy on the function evaluations at the particular point. Although methods exist for enlarging the domain of attraction, local methods often fail to converge. Venkataraman and Lucia (1988) argues that failure is "always due to some physical inconsistency in the model". However, providing good starting points, which is difficult if little knowledge exist about the system, still seems to be the only remedy for avoiding these inconsistencies. Also, local methods can at best find one solution for each given starting guess. Since most chemical engineering models contain non–linear equations, multiple solutions and other complex behaviour arise quite frequently. Knowledge about what kind of behaviour to expect is invaluable in both process design and control, hence methods that are able to detect for example steady state multiplicity should be availiable to the engineer. This calls for reliable *global* methods which are capable of handling most problems. However, these *"methods of last resort"* usually require high computational effort and should be used only if local methods fail. We also stress that there is yet *no* method for guaranteeing global convergence if a solution exist.

Outline of paper. In section 2 we give a brief review on classical approaches to formulation and numerical solution of various homotopy–continuation schemes. Although these methods have been used for quite some time in other disciplines, it is only during the last decade that successfull applications to chemical engineering problems have been reported. We discuss some inherent difficulties in using available homotopies, and in particular we adress the issue of potential unboundedness of the homotopy paths. To restrict the path to some finite domain we apply branch–jumping techniques based on variable transformations and arguments of symmetry. Such techniques have the advantage that no changes in the model equations are required. In section 3 we introduce a novel algorithm called *tear and grid method* in which we utilize conventional theory for partitioning and precedence ordering. We show how *explicit* solution schemes may be found by choosing a suitable set of decision (tear) variables. In conventional algorithms these procedures usually involve some iterative scheme. However, we illustrate that costly iterations may be avoided by making a *grid* in the tear variables. The solutions are finally obtained by visual inspection or interpolation as intersections of the residual *surfaces*. In section 4 we present a comparative analysis of the two methods in terms of a few example problems.

2 Homotopy–Continuation Methods

Due to the lack of global convergence properties for most conventional Newton or quasi–Newton methods, *homotopy-continuation methods* were introduced in solving chemical engineering problems during the late 70's and early 80's. The ideas of continuation methods are however not new, and were according to Ortega and Rheinboldt (1970) introduced in the literature by Lahaye (1934) to solve a single non–linear equation. A comprehensive survey of the use of homotopy continuation in computer aided design is given by Wayburn and Seader (1987). Among more recent applications within chemical engineering we mention a few important works. W. J. Lin (1987) computed multiple solutions of interlinked distillation columns. Kovach and Seider (1987) applied an algorithm with particular emphasis on avoiding limit points for the simulation of an industrial heterogeneous azeotropic distillation tower. A similar approach was used by Chang and Seader (1988) to show how certain design parameters affect a continuous reactive–distillation system. Fidkowski *et al.* (1991) use continuation methods to demonstrate how elementary bifurcation theory may be used in the design of nonideal multicomponent distillation. For reviews on methods and other applications we also refer to the works of Seider and Brengel (1991) or Allgower and Georg (1993). The mathematical principles of the continuation algorithm is thoroughly described in several textbooks (see e.g. Seydel (1988) or Kubiček and Marek (1983)). There also exist some semi–commercial applications, e.g. AUTO (Doedel n.d.) or HOMPACK (Watson *et al.* 1987)).

Problem formulation The underlying idea of homotopy continuation is to embed the model equations $f(x)$ in a *blending* function $H(x,t)$ forming the linear homotopy function

$$H(x,t) = tf(x) + (1-t)\,g(x) = 0 \tag{2}$$

where x denote the model variables, t the homotopy parameter and $g(x)$ a function for which the solution is known or easily obtained. Several choices exist for $g(x)$ each yielding a different homotopy with different behaviour. Alternative homotopies are :

$$\begin{aligned}
\text{Fixed point homotopy :} \quad & H(x,t) = tf(x) + (1-t)(x - x_0) && (3)\\
\text{Newton homotopy :} \quad & H(x,t) = tf(x) + (1-t)(f(x) - f(x_0)) && (4)\\
\textit{Affine homotopy} : \quad & H(x,t) = tf(x) + (1-t)\,A\,(x - x_0) && (5)
\end{aligned}$$

where A denotes a proper weighting matrix to avoid scaling problems, typically chosen as $f'(x_0)$. Yet another alternative is to choose $g(x)$ as a (related) problem which is easier to solve than $f(x)$. The different homotopies form a convex linear homotopy, meaning convexity in the sense that the errors in the components of $f(x)$ decreases linearly from the initial values given by x_0 (Kovach and Seider 1987). The advantage with the often used fixed point homotopy is, besides its simplicity, that any additional multiplicities introduced by adding further *functions* are avoided. Other homotopies may have more appealing numerical properties, but in general there are no definite guidelines for choosing the optimal one. We proceed by discussing briefly how solutions to the homotopy function may be obtained.

Solving the Homotopy function. A common approach to solving the homotopy function (2) is to reformulate the system of NAE's (2) to an initial value problem (IVP), as suggested by Davidenko (1953). By differentiating the homotopy function with respect to t we derive

$$\frac{dH\,(x,t)}{dt} = \frac{\partial H}{\partial x}\frac{dx}{dt} + \frac{\partial H}{\partial t} \tag{6}$$

Given an initial solution, x_0, equation (6) constitutes an IVP which may be integrated by any numerical integration scheme, e.g. Runge Kutta or Gear's method. Most continuation algorithms use some predictor–corrector scheme, typically an Euler predictor and Newton corrector step. The homotopy path is thus defined by the locus of all solutions found by tracking equation (2) from $t = 0$ with a known or easily obtained initial solution $x = x_0$, and ending at $t = 1$ for which $f(x) = 0$. Hence the desired solution to $f(x) = 0$ is obtained iff the homotopy path is tracked up to the point where $t = 1$. Under certain assumptions $H(x,t)$ is continuous such that the path containing $(x,0)$ also contains $(x,1)$ (see Ortega and Rheinboldt (1970)). Multiple solutions may also be detected by allowing the path to extend beyond the limits $0 \leq t \leq 1$. The continuity conditions are however often violated, and obstacles arise that may prevent successfull tracking.

Unboundedness and non–uniqueness of homotopy paths. A detailed analysis of situations under which homotopy–continuation methods may fail is given in the work of Wayburn and Seader (1987). In this work we discuss only the most common causes of failure, of which one is that

i) The Jacobian $H'(x)$ becomes singular at turning points

To avoid problems in tracking the solution curve across turning points, most algorithms proposed in the literature introduce the *arc–length* to define the search directions along the path. We use a slightly modified approach in which orthogonal search directions are found from linear algebra as described in a work by Morud (1995). In addition to difficulties in crossing turning points, problems also arise when

ii) The homotopy path becomes unbounded

Since the homotopy path needs to be finite in order to be trackable, some kind of branch jumping technique or variable transformation must be introduced if the homotopy branches connect in infinity. Seader *et al.* (1990) suggested to use mapped continuation methods, called *toroidal* and *boomerang* mapping, in which variables that extend to $\pm\infty$ are kept bounded through a proper transformation. The authors conjecture that *all* solutions may be traced from any starting point using a fixed point homotopy and allowing all variables to take complex values. Sufficient proof of such global convergence properties is not provided, and whether or not all solutions may be found from *one* starting point is still an open question. An objection to the latter approach is that introducing complex arithmetics considerably increases the computational efforts. Paloschi (1995) suggested instead to use new *bounded homotopies* that on one hand avoids tracking complex paths, and at the same time guarantees solution paths to remain inside a prescribed region. However, as recognized by the authors, more work needs to be done on theoretical aspects of these proposed homotopies. A third potential problem is that

iii) Multiple solutions may exist for $g(x) = 0$

If the added function $g(x)$ has multiple solutions, the homotopy path may return to a second solution of this simpler problem without passing through the desired solutions to $f(x) = 0$. Wayburn and Seader (1987) use the concept of *topological degree* to indicate when multiple solutions of $g(x)$ may cause failure. This problem is easily avoided by using fixed point of Affine homotopies since the residuals $x - x_0$ is simply a vector of scalars. Another problem frequently occurring is that

iv) Variables may exceed the domain on which they are defined

Since thermodynamic functions often involve logarithms or square roots, they become undefined when substituting for example negative values of mole fractions. This situation arise quite frequently since the solutions obtained along the homotopy path are not intrinsically feasible in a physical sense. Wayburn and Seader (1987) suggests to use the absolute value functions to resolve such problems. Finally, convergence may also be prevented by

v) Occurence of *isolated* solutions along the homotopy path

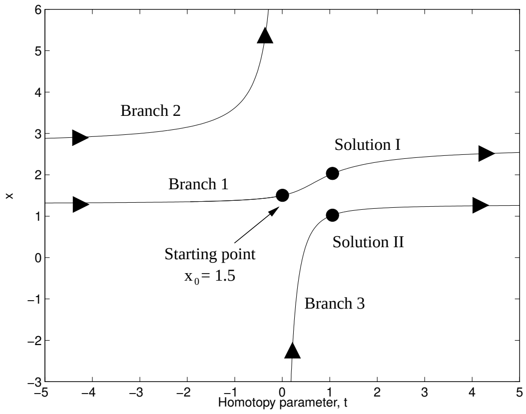
There are at present date no method which deals with overcoming problems of isolas, as noted also by Seader *et al.* (1990), thus we pose this problem as a great challenge for future work.

As an alternative to the approaches suggested by Seader and Paloschi, we will in the next section focus on some simple *branch–jumping* techniques in order to overcome the problems of unbounded paths. By using the simple *inverse mapping* function, we show how arguments of symmetry may be used to predict where solution branches connect across asymptotes. In cases of single and linear asymptotes we found that the inverse mapping works satisfactorily. However, in situations where for example several asymptotes lie arbitrarily close in the variable space (as for the CSTR example to be discussed later), or in cases of non–linear asymptotes, the situation is not that simple and the algorithm displayed poor convergence characteristics. Further work needs to be done on both theoretical and numerical aspects of this issue. Before going into detail on the mathematical issues of continuation and branch jumping, we illustrate the usefullness of applying branch jumping techinques by considering a simple scalar example function previously studied by W. J. Lin (1987) and Seader *et al.* (1990).

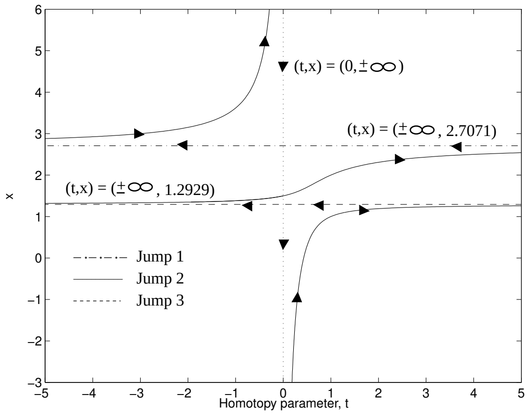
Introductory Example : Scalar Function. Consider the function given by

$$f(x) = x^2 - 3x + 2 \tag{7}$$

for which the analytical solutions $x = \{1, 2\}$ are easily obtained. We applied the fixed point homotopy $H(x,t) = t\left(x^2 - 3x + 2\right) + (1-t)(x - x_0)$ and figure 1 shows the homotopy paths and the two solutions to $f(x) = 0$ denoted I and II at $t = 1$. The path consists of three branches and is traversed in the direction indicated by arrows. Starting at the arbitrary initial point $x_0 = 1.5$ on branch 1, we find *both* solutions by allowing the homotopy path to extend beyond the interval $0 \leq t \leq 1$. We see that the path goes to infinity at the connection points $(t,x) = \{(0, \pm\infty), (\pm\infty, 1.2929), (\pm\infty, 2.7071)\}$. In spite of its unboundedness the homotopy curve is successfully tracked numerically by applying a simple *inverse mapping function* which imposes a jump of finite length in the mapped variable space. In the next



(a)



(b)

Figure 1: Homotopy paths for $H(x) = x^2 - 3x + 2 + (1-t)(x - x_0)$.

section we outline the principles of our continuation algorithm.

2.1 Continuation Algorithm

Given an initial solution to the homotopy function (2), say $(x_0, 0)$, the purpose of *continuation* is to calculate successive solutions

$$(x_1, t_1),\ (x_2, t_2), \ldots,\ (x_n, t_n) \tag{8}$$

until one or more solutions are found for which $t = 1$. The calculation sequence commonly consists of a predictor and corrector step. Our algorithm distinguishes itself somewhat from most other methods found in the literature, in that we introduce the augmented vector of state variables, i.e. $\hat{x} = [\ x \mid t\ ]^T$. This allows us to treat *any* parameter as the continuation parameter, hence we do not distinguish between parameters or variables in the solution procedure. Using the augmented $\hat{x}$ also allows us to find orthogonal search directions for the predictor and corrector step in a simple and straightforward manner. For reasons of simplicity we will in the rest of the presentation denote x as the augmented vector (slight abuse of notation).

Euler Predictor. In the predictor step we start at a known point on the solution curve, x_k, and make a finite step in a direction tangential to the curve. By linearizing the augmented homotopy function $H(x)$ around a point x_k we get

$$\Delta H(x_k) = J\Delta x = 0 \tag{9}$$

We recognize the matrix of partial derivatives $J = \partial H/\partial x$ as the augmented Jacobian matrix, where the $(n+1)$st column is the derivative of H with respect to the homotopy parameter t. We see from equation (9) that any step Δx *along* the homotopy path will be in the *null space* of the augmented Jacobian, $\mathcal{N}(J)$. The Jacobian is computed numerically using central differences. The Euler step thus becomes

$$x_{k+1} = x_k + h\Delta x \tag{10}$$

where h is the steplength in the direction of the nullspace–vector Δx.

Newton Corrector. In the corrector step we apply a Newton Raphson scheme in order to iterate towards a converged solution. By choice we require that the corrector steps should be taken in a direction *orthogonal* to the predictor step. We thus iterate in the row space of the Jacobian $\mathcal{R}(J^T)$, and the search direction is obtained from the pseudo–inverse J^+ evaluated at the previous solution point x_k. Mathematically stated the corrector steps thus becomes

$$x_{k+1}^{n+1} = x_{k+1}^{n} - J^{+}H\left(x_{k+1}^{n}\right) \tag{11}$$

The corrector step is repeated until the error norm is reduced beyond a given tolerance ϵ, say until $\|H(x_{k+1}^{n+1})\| \leq \epsilon$. If convergence problems are encountered one may improve the accuracy by updating the Jacobian matrix in the iteration steps. We may for example use a *rank one update* (Broyden) applied at each new iteration step (see e.g. Westerberg *et al.* (1979)). We apply a simple step–length algorithm where the step–length is updated according to the number of iterations in the corrector step. If a prescribed upper limit on the number of iterations is exceeded, the step–length is halved successively until convergence.

2.2 Branch–jumping Techinques

In order to enable branch–jumping one needs to define a direction for the jump in the solution space. We demonstrate how such directions may be found based on theory from linear algebra and simple arguments of symmetry. We outline the general principles of the methods, but the detailed mathematical issues are described in Appendix A.

Method 1: Aligned Asymptotes. This first method applies when the asymptotes are aligned with the coordinate axis of the original variable space, i.e. when asymptotes occur *either* for a given t or a given x_i. The underlying idea of the method is to impose a finite step in a transformed variable y which is mapped according to the *inverse mapping function*

$$y = \frac{1}{x_i} \tag{12}$$

where x_i denotes the variable for which asymptotes arise. The reader should note that we do not distinguish between variables x or the homotopy parameter t, since both variables may go to infinity along the path. In finding a point on the connecting solution branch we also use a predictor–corrector scheme. A step in the direction of the null-space of the original variable x_i and the corresponding step in the mapped variable y is illustrated in figure 2.
Using arguments of symmetry we seek a step in the mapped variable space so that

$$\frac{1}{x_{new}} = \frac{1}{x_{old}} + \Delta\left(\frac{1}{x}\right) \tag{13}$$

If we take the asymptote for the original variable x_i to constitute the ordinate in the mapped variable space y, we find if we allow x_i to become sufficiently large that the linearized homotopy path is actually symmetric in the vicinity of the *origin* in the mapped space. Difficulties in using this method arise when *several* variables become large simultaneously, or the asymptotes are not aligned with the orthogonal coordinate axes of the variable space $\mathcal{R}^m$. In such cases we wish to define a subspace in which only *one* of the variables are large. Finding such a subspace requires some way of rotating the coordinate axes, an issue that is dealt with in *method 2*.

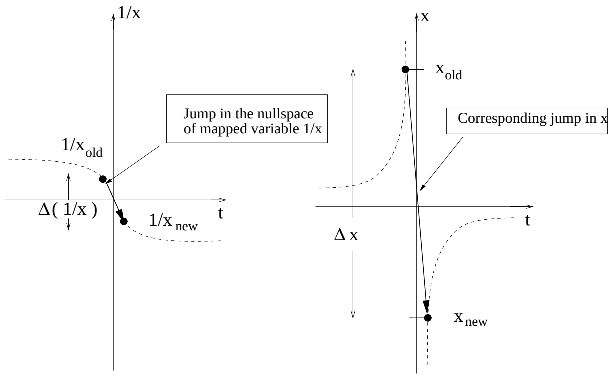


Figure 2: Schematic demonstrating the branch–jumping across asymptotes for *Method 1*

Method 2 : Skew Asymptotes. One way of finding the desired subspace in case of skew asymptotes is to align one of the coordinate axes in the rotated space with the direction of the asymptote of one variable. In order to use arguments of symmetry as described in the last section we aim to find a subset of coordinate axes which are orthogonal to the direction of the asymptote. A basis for such a rotated variable space is formed by a matrix, say T, whose column space is spanned by the bases for the left null-space, $\mathcal{N}(x^T)$, and the column space, $\mathcal{R}(x)$ of x. From linear algebra we know that for a n–dimensional vector $x \in \mathcal{R}^m$, this matrix T spans the whole of $\mathcal{R}^m$. If we denote the new variables by the veactor ζ, we have that

$$x = T\zeta \tag{14}$$

thus x is in the column space of T. To find the direction of the jump in the rotated variable space, $\Delta\zeta$, we simply find the direction of the null–space, $\mathcal{N}(\zeta)$. Hence we have

$$J\Delta x = JT\Delta\zeta = 0 \tag{15}$$

where the matrix product JT defines the Jacobian of the rotated function space, $f(\zeta(\lambda), \zeta)$. We wish to make a small step in $1/\zeta$ in the direction of the null space and thus follow the same steps as demonstrated for *Method 1* to eventually obtain $\Delta x = T\Delta\zeta$. For the simplest case of linear asymptotes we may again use symmetric arguments to predict the value of the mapped variables. However when the asymptotes are non–linear, symmetrical considerations are not at all that obvious. We have applied this method to some simple example problems, and found that problems arise both when asymptotes are non–linear and in cases where several asymptotes appear close in the variable space. Finding robust solution methods that deal with such situations represents a challenge that should be dealt with in future works. In the next section we introduce the features of the alternative *tear and grid approach*.

3 Methods Exploiting Algebraic Structure

A key issue which is often neglected when solving systems of NAE's is the task of decomposing the system into smaller subsets, called *partitions*, until irreducible partitions are obtained. These partitions are then solved in turn, which reduces the computational effort compared to simultaneous solution of the whole system. For large sparse problems Sargent (1981) claims that it is always worth partitioning, but as shown in this work the payback may be considerable also for small problems. The irreducible partitions may however be decomposed even further by guessing some of the variables occurring in the subset, a method known as tearing. If possible one should choose the number of tear variables so that the system of NAE's can be reduced to a sequence of single–variable equations. Since the tear variables are guessed arbitrarily prior to solving the remaining system of *non–teared functions*, a residual corresponds to each teared function, i.e. the functional value is different from zero. Conventional tearing methods thus involve some iterative scheme since the residuals must be adjusted until every teared function attain a value of zero. To avoid costly iterative schemes, we suggest instead to make a *grid* of the tear variables and calculate the residuals for each point in the grid. Depending on the required accuracy solutions may then be obtained either by visual inspection of the solution surfaces or by numerical interpolation between adjacent points in the grid. In addition to finding initial solutions, the method is also well suited for exploring solutions in the parameter space. Bifurcation diagrams are easily obtained by using the bifurcation parameters as the grid variables. In this work the tear (grid) variables are found by inspection, but in the general case an algorithm is required to obtain the optimal set of tear variables. Leigh (1973) propose an implicit scheme to find the *minimum* tear set for complete decomposition.

3.1 Conventional Design of Decomposition Methods

Decomposition methods aim at finding smaller subsets, or partitions, of the system of NAE's that are easy to solve. Partitioning in this sense involves the assignment of which *output* variables to be solved by each of the equations. This choice is not arbitrary since a partition of a system is unique (see e.g. Sargent (1981)). Precedence ordering on the other hand involves finding the order in which the equations are to be solved. We illustrate these ideas by the following simple example. To expose the underlying algebraic structure Himmelblau (1973) suggested to display the occurence (incidence) matrix, where each row corresponds to an equation f_i and each column to a variable x_i. Entries denoted by 1's in row i and column j thus indicate that variable x_j appears explicitly in equation f_i.

$$\begin{array}{cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 \\ f_1 & 1 & 1 & 1 & & 1 \\ f_2 & 1 & 1 & & 1 & 1 \\ f_3 & 1 & 1 & 1 & & \\ f_4 & 1 & 1 & & 1 & \\ f_5 & & & 1 & 1 & 1 \end{array} \tag{16}$$

As suggested by Westerberg *et al.* the best way to obtain the partitioning and precedence ordering is probably to use a *directed graph* where an arrow points from node f_i to f_j iff the assigned output variable for f_i appears in f_j. By assigning each variable x_i to the corresponding equation f_i, we obtain the directed graph shown in figure 3 (a). By making an appropriate choice of *tear*

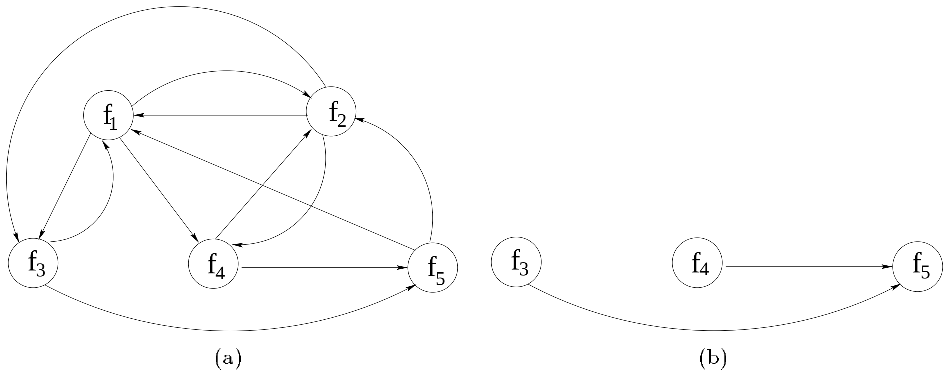


Figure 3: Directed (a) and reduced graph (b) for example (16)

variables we may then find a completely decomposed partition. By inspection of the directed graph in figure 3 (a) one finds that if x_1 and x_2 are assigned as tear–variables, leaving f_1 and f_2 for calculation of the residuals, x_3, x_4 and x_5 may be calculated explicitly from equations f_3–f_5. Figure 3 (b) shows the directed graph that results after choosing the tear variables, i.e. deleting object 1 and 2 and the corresponding arrows in figure 3 (b). In order to avoid computationally expensive iteration on the *teared* functions, we propose in the next section a method based on making a grid of the tear variables.

3.2 Tear and Grid Method

In the following one should note that finding *all* solutions requires that the non–teared functions yield unique solutions. A sufficient condition for uniqueness is that the system can be reduced to single variable equations by algebraic manipulations, i.e. one variable is explicitly calculated from one particular equation. One should note that these variables may appear as nonlinear terms, but in the extreme case a linear subset is obtained from which only matrix invertion is required to obtain the solution. The principles of the method are conveniently described by considering the system (16) discussed in the last section, since solutions may be visualized in 3 dimensional plots.

Finding initial solutions. After assigning the tear variables we find the appropriate partitioning and precedence ordering which yield a completely decomposed subset. We then make a *grid* of the chosen tear variables, in this case x_1 and x_2 which yields two residuals Res_1 and Res_2. By algebraic manipulation we may then calculate x_3–x_5 explicitly from f_3–f_5, provided each function allows for isolation of one of the variables. The residuals are then obtained for the two teared functions f_1 and f_2 for each point in the grid. This scheme leads to solution *surfaces* for Res_1 and $Res2$ in the 3 dimensional space when displayed as functions of the tear variables x_1 and x_2. Since we require that both residuals must be zero, solutions to the system of NAE's (16) are found in the plane defined by $Res_i = 0$ denoted the *zero contour*. The intersections betweeen the residual surfaces and the zero contour yield *lines* in 3 dimensional plane. Finally we obtain *all* solutions within the prescribed grid at the intersections of these two lines, i.e. as *points* in the zero contour. Since visulisation of surfaces is an important feature of the tear and grid method, the method is particularly suitable when the solution procedure requires only two tear variables, although this is by no means restrictive for the *numerical* solutions. We illustrate by some example problems given in the next section that solution surfaces are quite conveniently visualized.

Exploring solutions in parameter space. If the set of NAE's is *non square*, one needs in conventional steady state simulation to specify either a parameter or a variable for each degree of freedom (DOF). Instead of fixing the values of these variables, one may instead use the tear and grid approach to make a grid in the parameters or a convenient subset of the variables corresponding to the number of DOF. Hence one may solve an *underdetermined* set of equations to obtain solutions in the parameter space. By making a proper choice of grid variables and appropriate algebraic manipulations of the model equations, *explicit* solution schemes may be found also in the task of exploring solutions in the parameter space. However, we stress that finding all solutions requires that the variables are *unique* functions of the equations from which they are solved.

Coarseness of grid. Depending on the required accuracy for the solutions, the coarseness of the grid (number of points) may be chosen for convenience. One may for instance apply the grid method to obtain an initial *screening* of the solution surface in order to find in which regions solutions may be found. By reducing the domain of the grid variables step by step and thereby narrowing in on the solutions, one may decrease the computation time considerably compared to making a fine grid of the *whole* solution space. Such screening may be very usefull when little information exist about the process and thus in which regions solutions are likely, which often is the case in the design of new processes.

Interpolation methods. As illustrated in the previous outline, solutions are in the final step detected as intersections of *lines* in the socalled zero contour, which are projections of residual surfaces. Displaying these lines by virtue involves interpolation between the projected grid points. Most available algorithms for interpolation use some polynomial approximation, i.e. Lagrange polynomial or inverse interpolation. The interpolation polynomial aims at connecting the computed data points by selecting a polynome of appropriate order, which is related to the number of grid points. If the chosen grid is too coarse one may either fail to find some of the *true* solutions, or introduce additional solutions due to failure in the interpolation routine. The method applied in this work uses an inverse distance method (The MAthWorks 1992). The solutions obtained by visual inspection are usually approximate, hence if higher accurracy is required one should either resort to interpolation between neighbouring points, or use the approximate solutions as initial guesses for some Newton–based method.

4 Numerical Results

We present in this section numerical results obtained with the proposed methods for two simple examples of physical processes, i.e. an example of two CTSR's in series and a coupled cell reaction. The example problems are primarily chosen so that problems related to complex behaviour of the homotopy path are easily demonstrated. For the CSTR example we show that convergence problems arise if several variables become unbounded simultaneously, since it is difficult to obtain an appropriate direction for the branch jump. Attempting to align the coordinate axis with the asymptotes by rotation (*method 2*) also failed in this case. For the coupled cell example we illustrate that isolas may occur along the homotopy path, thus preventing successfull tracking. For both examples we show that multiple solutions are easily obtained without difficulties by applying the *tear and grid method.*

Example 1 : Two CSTR's in series
We here consider a model of two CSTR's in series (see e.g. Kubiček and Marek (1983) or Seydel (1988) for details regarding the model). The steady state model is comprised by a system of *four* coupled non–linear equations

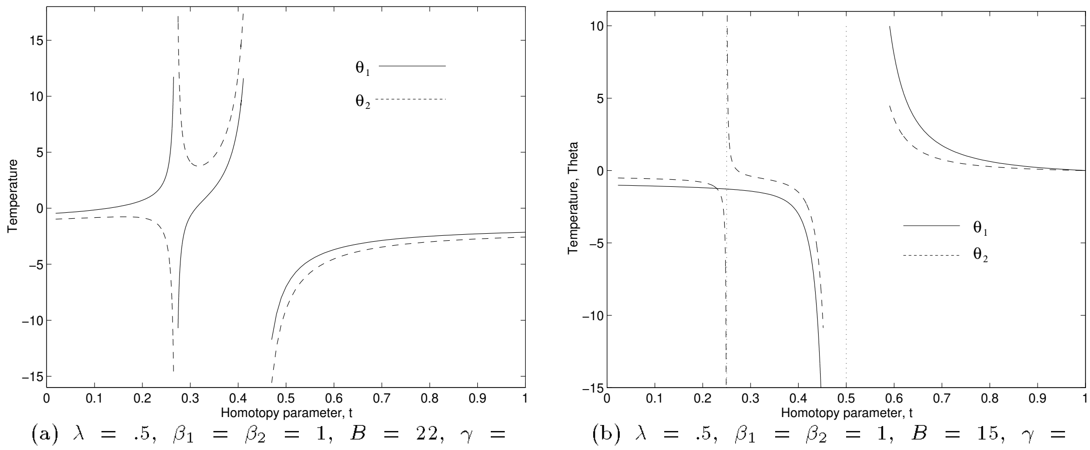
$$(1-\Lambda)\,x_2 + Da_1\,(1-x_1)\,exp\left(\frac{\theta_1}{1+\theta_1/\gamma}\right) = 0 \tag{17}$$

$$(1-\Lambda)\,\theta_2 - \theta_1 + Da_1 B\,(1-x_1)\,exp\left(\frac{\theta_1}{1+\theta_1/\gamma}\right) - \beta_1\,(\theta_1-\theta_{c1}) = 0 \tag{18}$$

$$x_1 - x_2 + Da_2\,(1-x_2)\,exp\left(\frac{\theta_2}{1+\theta_2/\gamma}\right) = 0 \tag{19}$$

$$\theta_1 - \theta_2 + Da_2 B\,(1-x_2)\,exp\left(\frac{\theta_2}{1+\theta_2/\gamma}\right) - \beta_2\,(\theta_2-\theta_{c2}) = 0 \tag{20}$$

Homotopy–continuation. We applied both fixed point and Newton homotopies in order to find a solution with the parameter values fixed as given in figure 4 (a) and (b). As demonstrated in figure 4 the homotopy function displays rather complex behaviour, and convergence problems ocurred due to difficulties in finding a proper direction of the *branch jump*. Numerical instability, in terms of unstable oscillations in the correction step, was encountered due to the Jacobian matrix becoming extremely illconditioned in the vicinity of the asymptotes. This example illus-



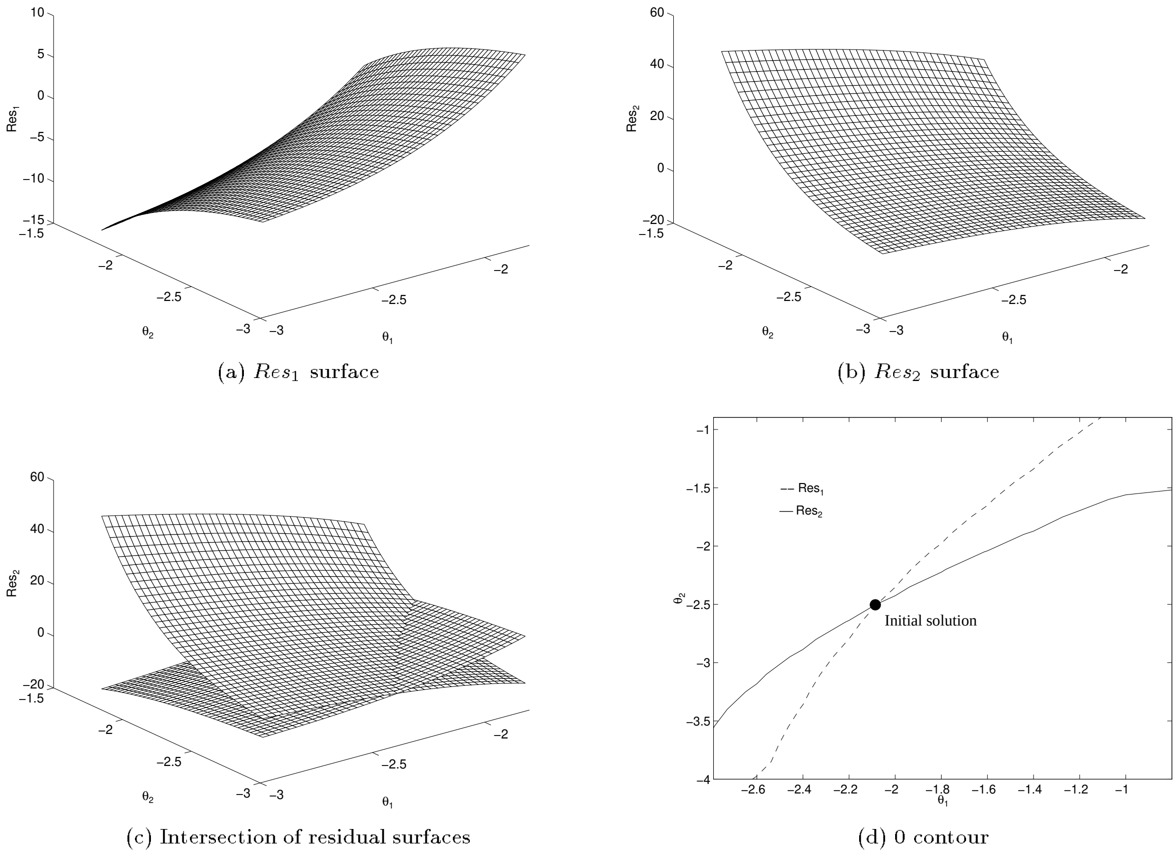
(a) $\lambda = .5,\ \beta_1 = \beta_2 = 1,\ B = 22,\ \gamma = 1000,\ \theta_{c1} = \theta_{c2} = -3,\ Da_1 = Da_2 = 0.5$

(b) $\lambda = .5,\ \beta_1 = \beta_2 = 1,\ B = 15,\ \gamma = 1000,\ \theta_{c1} = \theta_{c2} = -3,\ Da_1 = Da_2 = 0.5$

Figure 4: Fixed point homotopy paths for example of two CSTR's in series

trates one of the problems often encountered when using *branch–jumping* techniques based on arguments of symmetry, namely that an upper limit on the size (norm) of x needs to be chosen *apriori* for when the jump is to be taken. If the upper bound is chosen such that the direction of the null–space vector for the mapped variable, evaluated at $1/x$, does not pass through or rather closely to $-1/x_i$, convergence problems in the corrector step are to be expected. Convergence properties thus relies heavily on the *inverse* being sufficiently close to zero. If not, it becomes difficult to predict at which point the branches connect beyond the asymptote. Another problem related to the choice of an upper bound is that the path may extend to *large* values of x and return without displaying asymptotic behaviour. For the example considered here it may be preferable to apply either *bounded homotopies* as suggested by Paloschi (1995) or allowing continuation in the complex domain (Seader *et al.* 1990). However, there are no proofs that guarantee convergence for neither of these methods. In this case the homotopy path was eventually tracked by applying continuation up to a point in the vicinity of the second (vertical) asymptote. The remaining solution branches was then found by discretizing the homotopy function in t and solving the system of equations by a Newton–method.

Tear and grid approach. Although the homotopy–continuation algorithm exhibited poor convergence properties, initial solutions were quite easily obtained by the grid method. By inspection of the model equations we find that assigning θ_1 and θ_2 as grid variables, allows for explicit calculation of x_1 and x_2 from (18) and (20) respectively. Since x_1 and x_2 occur linearly we also note that the solutions are unique. In figure 5 a) and b) we illustrate the shapes of the residual–surfaces calculated from (17) and (19). The parameter values corresponds to the case given in figure 4 (a). The intersection of the residual surfaces and the 0–contour yields lines in three dimensional space as shown in figure (c), and the initial solution for θ_1 and θ_2 is eventually obtained by the intersection of these lines as shown in (d). As a verification of the method we see



(a) Res_1 surface

(b) Res_2 surface

(c) Intersection of residual surfaces

(d) 0 contour

Figure 5: Visualisation of solutions to $f(x) = 0$ for CSTR example

that the solution corresponds to the one found using the homotopy–continuation method, but in this case solutions were obtained with considerably less effort. In the next example we illustrate that the homotopy functions may display isolas along the homotopy path, thus introducing additional difficulties with respect to branch jumping.

Example 2 : Coupled Cell Reaction
Consider the following system describing a model of a trimolecular reaction (see Seydel (1988) for details concerning the model)

$$\begin{aligned} 2 - 7x_1 + x_1^2 x_2 + \lambda(x_3 - x_1) &= 0 \\ 6x_1 - x_1^2 x_2 + 10\lambda(x_4 - x_2) &= 0 \\ 2 - 7x_3 + x_3^2 x_4 + \lambda(x_1 + x_5 - 2x_3) &= 0 \\ 6x_3 - x_3^2 x_4 + 10\lambda(x_2 + x_6 - 2x_4) &= 0 \\ 2 - 7x_5 + x_5^2 x_6 + \lambda(x_3 - x_5) &= 0 \\ 6x_5 - x_5^2 x_6 + 10\lambda(x_4 - x_6) &= 0 \end{aligned} \tag{21}$$

where the coupling coefficient λ is chosen as the branching parameter.

Homotopy–continuation. We applied a fixed point homotopy function to the model and found (incidentally) that isolated solution branches appear along the homotopy path. Isolas ocurred for a wide selection of starting guesses, and prevented in each case successfull tracking of the homotopy path. This is hardly surprising since the mapping functions described as methods 1 and 2 are in general not suited for branch–jumping from isolas. There are at present date no rigorous way of predicting where the branches connect beyond the isolas. For some homotopies it is however possible to guarantee that isolated solutions will not arise (Paloschi 1995), but this is not an issue dealt with in this work.

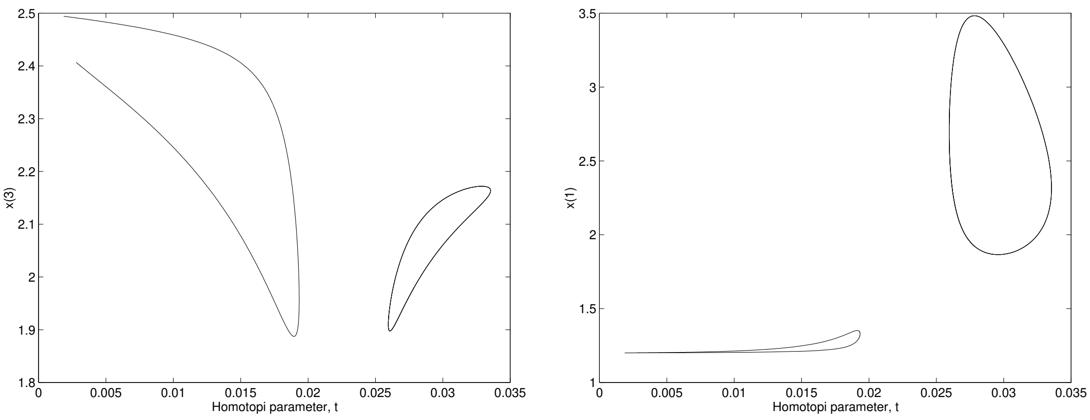


Figure 6: Occurence of isolas along the homotopy path for coupled cell reaction example

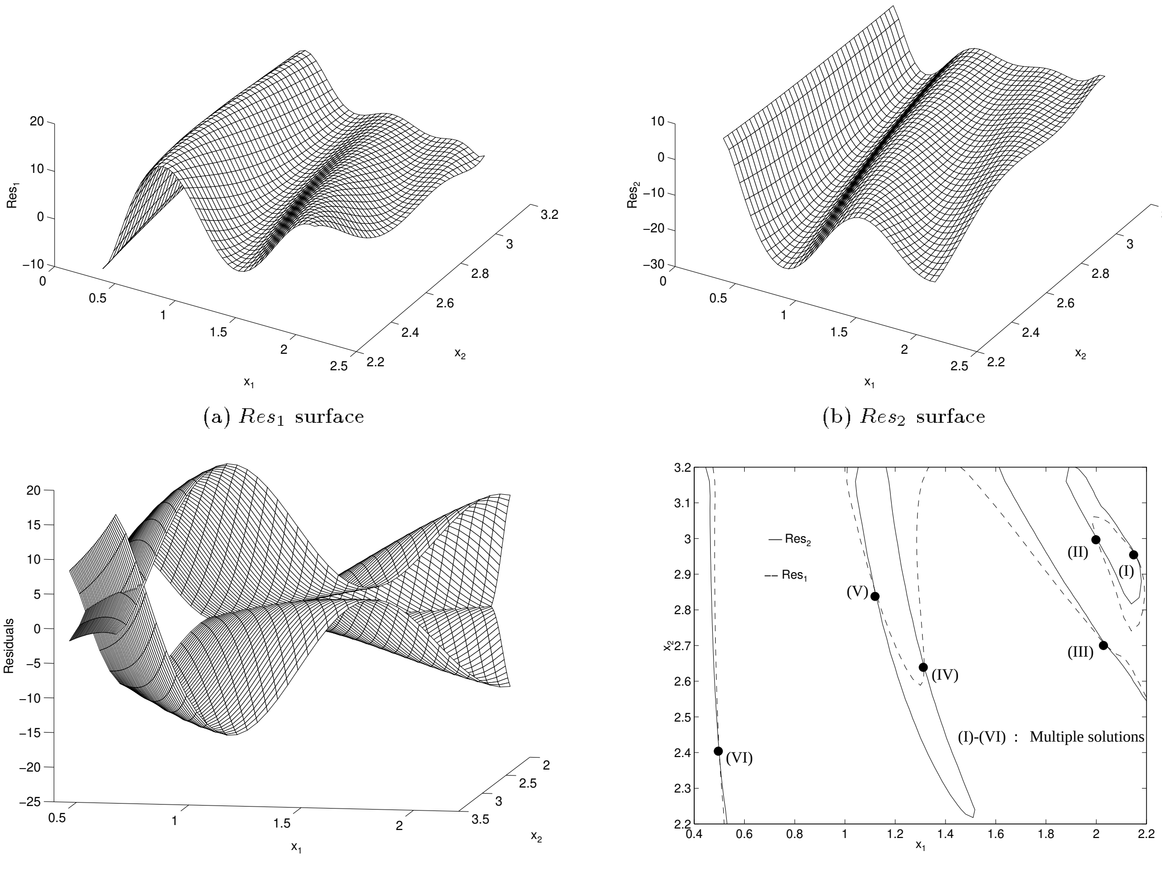
Tear and grid approach. From the occurence matrix displayed in (22) we find that an explicit solution scheme is obtained by choosing for example x_1 and x_2 as tear variables. We note that other sets of tear variables also yield explicit schemes. Deleting the rows for x_1 and x_2 yields the assignment of output variables as indicated by encircled occurrences in (23), which leaves f_5 and f_6 for calculation of the residuals.

$$\begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ f_1 & 1 & 1 & 1 & & & \\ f_2 & 1 & 1 & & 1 & & \\ f_3 & 1 & & 1 & 4 & 1 & \\ f_4 & & 1 & 1 & 1 & & 1 \\ f_5 & & & 1 & & 1 & 1 \\ f_6 & & & & 1 & 1 & 1 \end{array} \tag{22}$$

$$\begin{array}{c|cccc} & x_3 & x_4 & x_5 & x_6 \\ f_1 & 1 & & & \\ f_2 & & 1 & & \\ f_3 & 1 & 1 & 1 & \\ f_4 & 1 & 1 & & 1 \\ f_5 & 1 & & 1 & 1 \\ f_6 & & 1 & 1 & 1 \end{array} \tag{23}$$

Unique solutions are guaranteed since the assigned variables appear as linear terms in the unteared functions. The residual surfaces and solutions are displayed in figure 7, and as shown in d) there are in fact 6 solutions denoted (I)–(VI) for x_1 and x_2 within the prescribed grid for

$\lambda = 1.3$. Our results are in excellent agreement with results given by Seydel (1988), who applied continuation to explore solutions for a range of λ–values.



(a) Res_1 surface

(b) Res_2 surface

(c) Intersection of residual surfaces

(d) 0 contour

Figure 7: Visualisation of solutions to $f(x) = 0$ for coupled cell reaction example

5 Discussion and Conclusions

We have discussed two alternative numerical methods for solving systems of nonlinear algebraic equations (NAE's). We first considered a homotopy continuation scheme, which requires substantial effort both in terms of implementation (code) and computing time. With the exception of a case where isolated solution branches occurred along the homotopy path, all solutions were successfully obtained by using a globally convergent fixed point or Newton homotopy. However, the method may also fail due to potential unboundedness of the homotopy path. To resolve problems arising when variables extend to infinity, we applied two simple *branch jumping* tecniques. By using a simple *inverse mapping* functions we show how search directions for a predictor–corrector scheme may be found by utilizing theory from linear algebra and arguments of symmetry. However, in all cases, also when the homotopy continuation method failed, we showed that a novel *tear and grid* method found all solutions. The *tear and grid method* posesses some appealing properties compared to other methods proposed in the literature. Among these features we recognize straightforward implementation and low computational cost due to non–iterative solution schemes. However, the perhaps most appealing feature is the convenient visualization of solution surfaces, which applies to finding initial solutions as well as exploring solution in the parameter space. Among the drawbacks we recognize that the method may require considerable algebraic manipulations of the equations in order to decompose the system of NAE's into single variable equations. We also stress that the tear and grid method is not suited to deal with large problems since the number of computations becomes excessive. Hence this method is in general appropriate only for relatively small problems, which may still include many problems of great interest. In process analysis one is frequently set to analyse small–sized subsystems of the more complex chemical engineering plant in order to obtain important information regarding process behaviour. Proof of multiple steady states in ideal binary distillation (Jacobsen and Skogestad 1991), explanation of *holes* in some operating regions for integrated distillation columns (Morud 1995) or exhaustive analysis of dynamic behaviour displayed by CSTR's (e.g. van Heerden (1953) and Uppal *et al.* (1974)) are all examples of progressive discoveries obtained by resorting to analysis of simplified problems. For problems of somewhat greater size and complexity one may consider using *partial* tearing. Even though iterations are required in this case within some partitions, considerable savings should be possible since simultaneous solution of the whole problem is avoided. Also, there already exist algorithms for choosing a convenient set of tear variables. Finally we stress that depending on the size, complexity and difficulty of the problem at hand, one should in any case consider the grid approach as a worthy candidate for solving systems of NAE's frequently occurring within chemical engineering, along with the spectre of solution methods that already exist.

Acknowledgements. Constructive comments from Prof. Terje Hertzberg at the Departement of Chemical Engineering is gratefully acknowledged.

References

Allgower, E. and K. Georg (1993). Continuation and Path Following. *Acta Numerica* pp. 1–64.

Chang, Y. A. and J. D. Seader (1988). Simulation of Continuous Reactive Distillation by a Homotopy–Continuation Method. *Comp. Chem. Eng.* **12**(12), 1243–1255.

Davidenko, D. (1953). On a New method of Numerically Integrating a System of Nonlinear Equations. *Dokl. Akad. Nauk USSR* **88**, 601.

Doedel, E. J. (n.d.). AUTO: Software for Continuation and Bifurcation Problems in Ordinary Differential Equations. Technical report.

Fidkowski, Z. T., M. F. Malone and M. F. Doherty (1991). Nonideal Multicomponent Distillation : Use of Bifurcation Theory for Design. *AIChE Journal* **37**(12), 1761–1779.

Himmelblau, D. M. (1973). Morphology of Decomposition. In Decomposition of Large-Scale Problems. North Holland, Amsterdam.

Jacobsen, E. W. and S. Skogestad (1991). Multiple Steady–States in Ideal Two–Product Distillation. *AIChE Journal* **37**(4), 499.

Kovach, J. W. and W.D. Seider (1987). Heterogenous Azeotropic Distillation – Homotopy–Continuation Methods. *Comp. Chem. Eng.* **11**(6), 593–605.

Kubiček, M. and M. Marek (1983). *Computational Methods in Bifurcation Theory and Dissipative Structures.* Springer Series in Computational Physics. Springer–Verlag New York Inc.

Lahaye, E. (1934). Une méthode de résolution d'une catégorie déquations transcendantes. *C. R. Acad. Sci.* **198**, 1840–1842.

Leigh, M. J. (1973). A Computer Flowsheeting Program Incorporating Algebraic Analysis of the Problem Structure. PhD thesis. University of London.

Morud, J. (1995). Studies on the dynamics and operation of integrated plants. PhD thesis. University of Trondheim, The Norwegian Institute of Technology.

Ortega, J. M. and W. C. Rheinboldt (1970). *Iterative Solutions of Nonlinear Equations in Several Variables.* Academic Press. New York.

Paloschi, J. R. (1995). Bounded Homotopies to Solve Systems of Algebraic Nonlinear Equations. *Comp. Chem. Eng.* **19**(12), 1243–1254.

Sargent, R. W. H. (1981). A Review of Methods for Solving Nonlinear Algebraic Equations. In: *Foundations of Computer-aided Process Design* (R. S. H. Mah and W. D. Seider, Eds.). Engineering Foundation. New York.

Seader, J. D., M. Kuno, W. J. Lin, S. A. Johnson, K. Unsworth and J. Wiskin (1990). Mapped Continuation Methods for Computing All Solutions to General Systems of Nonlinear Equations. *Comp. Chem. Eng.* **14**(1), 71.

Seider, W. D. and D. B. Brengel (1991). Nonlinear Analysis in Process Design. *AIChE Journal* **37**(1), 1–37.

Seydel, R. (1988). *From Equilibrium to Chaos, Practical Bifurcation and Stability Analysis.* Elsevier Science Publishing Co., Inc.

The MAthWorks, Inc. (1992). *MATLAB Reference Guide.*

Uppal, A., W. H. ray and A. B. Poore (1974). On the Dynamic Behaviour of Continuous Stirred Tank Reactors. *Chem. Eng. Sci.* **29**, 967–985.

van Heerden, C. (1953). Autothermic Processes. *Ind. Eng. Chem.* **45**(6), 1242–1247.

Venkataraman, S. and A. Lucia (1988). Solving Distillation Problems by Newton–like Methods. *Comp. Chem. Eng.* **12**(1), 55–69.

W. J. Lin, J. D. Seader, T. L. Wayburn (1987). Computing Multiple Solutions to Systems of Interlinked Separation Columns. *AIChE Journal* **33**(6), 886–897.

Watson, L.T., S. C. Billups and A. P. Morgan (1987). Algorithm 652, HOMPACK : A Suite of Codes for Globally Convergent Homotopy Algorithms. *ACM Trans. Math. Software* **11**, 281.

Wayburn, T. L. and J. Seader (1987). Homotopy Continuation Methods for Computer Aided Process Design. *Comp. Chem. Eng.* **11**, 7–25.

Westerberg, A. W., H. P. Hutcison, R. L. Motard and P. Winter (1979). *Process Flowsheeting.* Cambridge University Press.

A Search directions for inverse mapping function

We here show how one may find the step of a desired length in the null–space of the *mapped variable space* which we may denote

$$x' = (x_1, \ldots,\ y, \ldots,\ x_n)\,,\ \in R^m \tag{24}$$

where y denotes the simple inverse mapping $y = 1/x$ of the variable(s) for which the homotopy path becomes unbounded. The elements in the Jacobian matrix for the mapped variable space may be written

$$J' = \frac{\partial f}{\partial x'} = (f_{x_1}|\ldots|f_y|\ldots|f_{x_n}) \tag{25}$$

By partial differentiation of the inverse mapping function we obtain

$$f_y = \frac{\partial f}{\partial x}\cdot\frac{\partial x}{\partial y} = -x_i^2 f_{x_i} \tag{26}$$

To avoid introducing the mapped variables explicitly in the system of equations, we simply multiply the column of the original Jacobian by minus the square of x_i, evaluated at the point from which the jump is to be taken. The direction of the jump in the mapped variable space is then

$$n = (n_1, \ldots, n_y, \ldots, n_n)\,,\quad \in \mathcal{N}(\mathcal{J}') \tag{27}$$

In cases of vertical or linear asymptotes, we may use arguments of symmetry in order to predict new values of the variables beyond the asymptote. For the inverse mapping function we wish to make a finite jump from $1/x_{old}$ to $1/x_{new}$. Due to symmetry around origo in the mapped variable space we assume that $1/x_{new} = -1/x_{old}$. The desired step $\Delta(1/x)$ must satisfy the condition

$$\frac{1}{x_{new}} = \frac{1}{x_{old}} + \Delta\left(\frac{1}{x}\right) \tag{28}$$

Substituting $(1/x_{new})$ by $-(1/x_{old})$ in (28) yields the desired step

$$\Delta\left(\frac{1}{x}\right) = -\frac{2}{x_{old}} \tag{29}$$

In order to obtain a jump of desired length we normalize the vector spanning the null–space by the scaling

$$n' = -\frac{2}{x_i\cdot n_y}n \tag{30}$$

Finally we substitute the n_y element in the null–space vector (27) by $-2x_i$, thus mapping the variable space back to the original $x \in R^m$. Since the curve for the inverse only rarely is *absolutely* symmetrical around origo for the point from which the jump is taken, we need to apply a Newton corrector in order to find a converged solution. Total symmetry is only found for jumps in the vicinity of origo (see figure 2, thus we have to make sure that x is sufficiently large before imposing the step. This method therefore works satisfactorilly only in cases where the homotopy path is symmetric around the asymptote.