Modelling techniques for biological reaction systems 2. Modelling of the steady state case

A E Billing and P L Dold*

Department of Chemical Engineering, University of Cape Town, Private Bag, Rondebosch 7700, South Africa.

Abstract

This paper is the second in a series of three which deals with numeric techniques for biological reaction systems. For the steady state problem, a set of simultaneous non-linear algebraic equations must be solved. Representing the mass balance equations in a matrix format assists in the choice of suitable numerical solution procedures. Five widely-used procedures are evaluated and compared in application to the types of flow sheet encountered in practice. Newton's method, with a finite difference approximation to the Jacobian, was the most successful technique.

Introduction

In a biological reaction system, "steady state" conditions are defined as those where the system operates under conditions of constant input flow rate and load and where the operating conditions are held constant. The problem in modelling is one of predicting the state of the system for different system configurations and operating conditions. That is, under these constant input conditions, the response of each compound in each completely mixed reactor is described by a single concentration value which does not vary with time. It is these concentration values that provide the solution to what is termed the "steady state" problem.

A system which operates under steady state conditions as described above can be characterised by a set of simultaneous mass balance equations which include non-linear terms. Any time-dependent or derivative terms will be zero, and the set of equations will therefore be algebraic. The solution to the system of non-linear algebraic equations cannot be expressed in closed form, so "exact" of direct methods cannot be applied. Instead, iterative procedures must be employed. These require an initial estimate of the solution which is updated via a linear approximation of the relevant mass balance functions. The updating procedure is repeated until convergence is achieved. The main concern is the selection of a solution technique that will guarantee convergence. Additional considerations in the choice of a suitable numerical method would be its computational efficiency, robustness and stability.

A case study: continued

In Part 1 of this series of three papers a case study problem based on a simple biological model and comprising a single completely mixed aerobic reactor plus settling tank was introduced (Fig. 1). The response of the system is described by eight mass balance equations, one for each of the four compounds in the reactor and in the underflow recycle from the settler (Eqs. (14) to (21) of Part 1). Under steady state conditions, any derivative terms in these equations fall away, and the resultant eight steady state mass balances become:

Reactor:

$$Q_r X_{B,r} - (Q_i + Q_r)X_B - b\,X_B\,V + \frac{\hat{\mu}\,S_S\,X_B}{(K_S + S_S)}\,V = - Q_i X_{B,i} \quad (1)$$

$$Q_r X_{E,r} - (Q_i + Q_r)X_E + f\,b\,X_B\,V = - Q_i X_{E,i} \quad (2)$$

$$Q_r X_{S,r} - (Q_i + Q_r)X_S + (1-f)\,b\,X_B\,V - \frac{K_H\,X_S}{(K_X + X_S/X_B)}\,V = - Q_i X_{S,i} \quad (3)$$

$$Q_r S_{S,r} - (Q_i + Q_r)S_S - \frac{\hat{\mu}}{Y}\cdot\frac{S_S\,X_B}{(K_S + S_S)}\,V + \frac{K_H\,X_S}{(K_X + X_S/X_B)}\,V = - Q_i\,S_{S,i} \quad (4)$$

Solids/liquid separator:

$$(Q_i + Q_r - q_w)\,X_B - Q_r\,X_{B,r} = 0 \quad (5)$$

$$(Q_i + Q_r - q_w)\,X_E - Q_r\,X_{E,r} = 0 \quad (6)$$

$$(Q_i + Q_r - q_w)\,X_S - Q_r\,X_{S,r} = 0 \quad (7)$$

$$S_S - S_{S,r} = 0 \quad (8)$$

These equations may be written in the form $\mathbf{f(x)} = 0$ where $\mathbf{x}$ is the vector of state variables:

$$\mathbf{x} = \begin{bmatrix} X_B \\ X_E \\ \vdots \\ X_{S,r} \\ S_{S,r} \end{bmatrix}$$

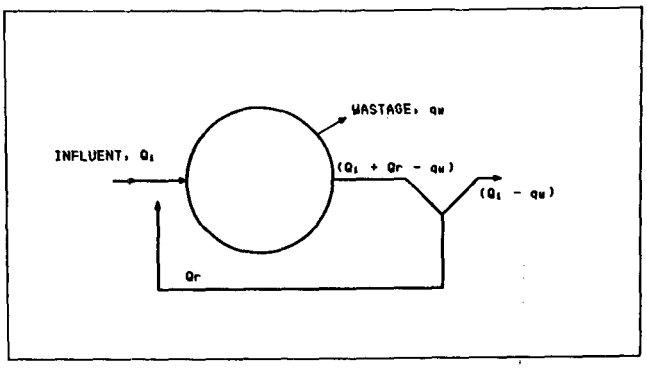


Figure 1
A case study: a single aerobic reactor with settling tank.

* To whom all correspondence should be addressed.
Received 27 October 1987.

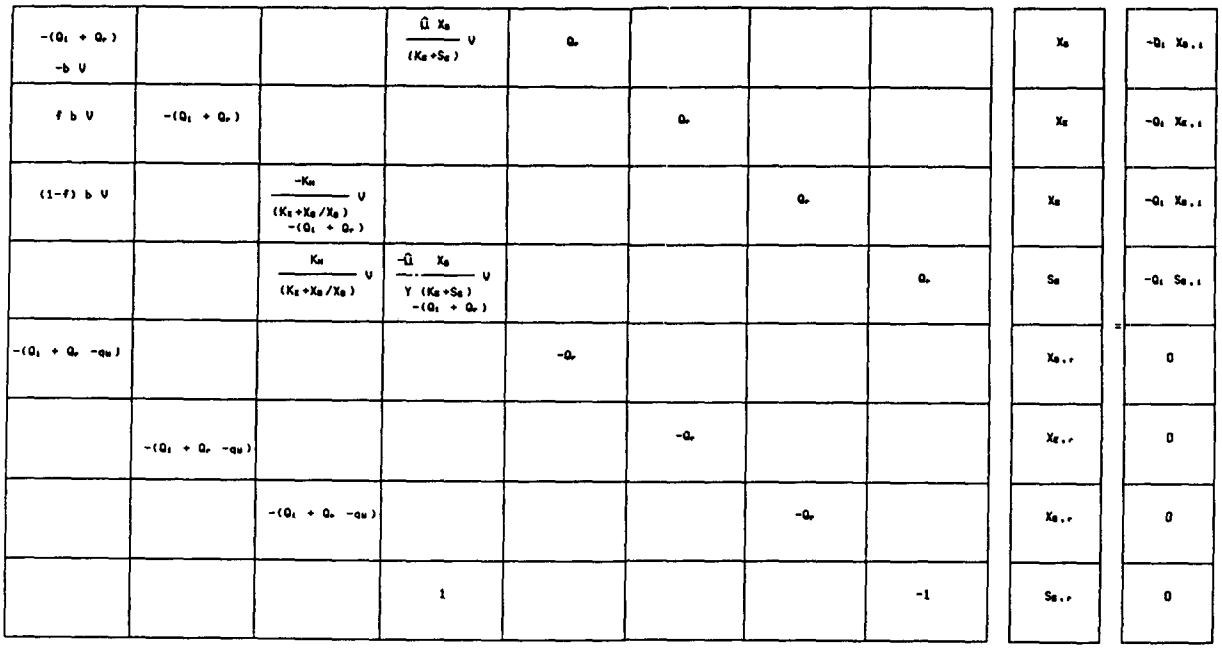


Figure 2
A matrix representation of the steady state mass balance equations for a single reactor and settling tank system. Additional subscripts i and r denote the concentrations in the influent and the underflow recycle respectively.

Some insight into appropriate numerical solution procedures for the steady state problem may be gained by representing the equations in a matrix format.

The steady state matrix

The matrix representation is used here because it gives a concise summary of the steady state problem. It shows, amongst others, features such as feed distribution, the flow links between reactors and the conversion processes, in a "graphical" manner.

Consider how the eight simultaneous steady state mass balance equations (Eqs. (1) to (8)) of the case study are transformed into the matrix format in Fig. 2. The equations are expressed in the form:

$$A\,X = B$$

The X vector: Eqs. (1) to (8) are mass balances for the eight state variables X_B, X_E, ..., $X_{S,r}$ and $S_{S,r}$. These state variables form the X vector, which is the solution to the steady state problem.

The B vector: This is the "feed vector". It contains the elements of the rigth-hand sides of Eqs. (1) to (8). Each term is the influent mass input rate of the corresponding compound into the particular zone (negative value). In this case, the first four values are the influent mass input rates of X_B, X_E, X_S and S_S into the reactor. The last four values are the influent inputs into the settler (zero here).

The A matrix: The A Matrix contains the reaction and flow terms which characterise the particular activated sludge system configuration. It is of interest to note how the non-linear terms are handled. Consider how Eq. (1) is inserted in the top row of the matrix. The linear terms can only be placed in one location. These are $-(Q_i + Q_r)$, Q_r and $-bV$. However, the non-linear term $(\hat{\mu}\, S_S\, X_B\, V/\,(K_S + S_S))$ can be handled in two ways:

$-(\hat{\mu}\, S_S\, V/\,(K_S + S_S))$ in the X_B location
or
$-(\hat{\mu}\, X_B\, V/\,(K_S + S_S))$ in the S_S location

In this case, the second option has been used.

The A matrix is always square and has dimension (number of compounds x (number of reactors + 1)). This system contains four different compounds and one reactor (plus settler). Hence, the size of the A matrix will be eight by eight. Each four by four "block" of the matrix contains specific information about the nature of the system being analysed.

- The upper left "block" contains terms for the reaction processes occurring in the reactor. Also, on the diagonal, flow-related terms appear. These represent the total flow out of the reactor (equal to the sum of the flows into the reactor i.e. $-(Q_i + Q_r)$).
- The lower right block represents the settler. Because no reaction takes place in the settler, only flow-related terms appear. These represent flow out of the settler which is recycled within the system ($-Q_r$ for particulate and -1 for soluble compounds).
- The diagonal vector, Q_r, in the upper right block of the matrix represents the underflow recycle from the settler to the first reactor. Flows directed upstream or "backwards" such as recycles will always lie above the diagonal blocks of the A matrix.

- The diagonal vector ($Q_i + Q_r$), (with + 1 for the soluble compound) in the lower left block of the matrix represents the flow from the reactor into the settling tank. Downstream or "forward" flows will always lie below the diagonal blocks of the A matrix.

Let us now extend the example to a system consisting of n reactors in series, followed by a settling tank. The system can be represented in general matrix format as shown in Fig. 3.

The X vector: The x vector contains the terms $X_{B,1}$, $X_{E,1}$, ..., $X_{S,r}$, $S_{S,r}$. These are the concentrations of the compounds X_B, X_E, X_S and S_S in reactors 1, 2, ..., n and in the underflow recycle from the settler, r. These state variables form the solution to the steady state problem.

The B vector: The B vector contains the feed terms which are the influent input rates of the corresponding compounds into each reactor. In situations where all the feed enters the first reactor, only the first four terms will appear in the vector; all other terms will be zero. If the feed to the system is split, with a portion of the feed entering the k^{th} reactor, then the corresponding locations in the B vector will accordingly be filled with non-zero terms.

The A matrix: This is a square matrix of dimensions ((n + 1) x no. of compounds). The large matrix can be subdivided into (n + 1) by (n + 1) submatrices. Each submatrix is square with dimension equal to the number of compounds.

Consider the k^{th} reactor in the series. The terms representing the conversion processes occurring in the k^{th} reactor will be situated in the k^{th} reactor "block" on the diagonal of the A matrix as indicated in Fig. 3. In addition, the diagonal within the k^{th} reactor block will contain terms representing flow out of the k^{th} reactor. Flow from the k^{th} reactor to the $(k + 1)^{th}$ reactor in the series will be represented by a diagonal vector containing the relevant flow terms in a block situated directly "below" the k^{th} block on the diagonal. That is, the vertical location of the block will be fixed opposite the column representing the k^{th} reactor. The horizontal location of the block will be fixed by the column representing the $(k + 1)^{th}$ reactor.

Recycle flows from one reactor to another in the series are handled in a similar fashion. A recycle from the k^{th} to the i^{th} reactor in the series will be represented by a diagonal vector containing the relevant flow terms in a block situated above the diagonal of the A matrix. The vertical location of the block will be fixed by the column representing the k^{th} reactor and the horizontal location of the block will be fixed by the column representing the i^{th} reactor. In general, the vertical position of the submatrix

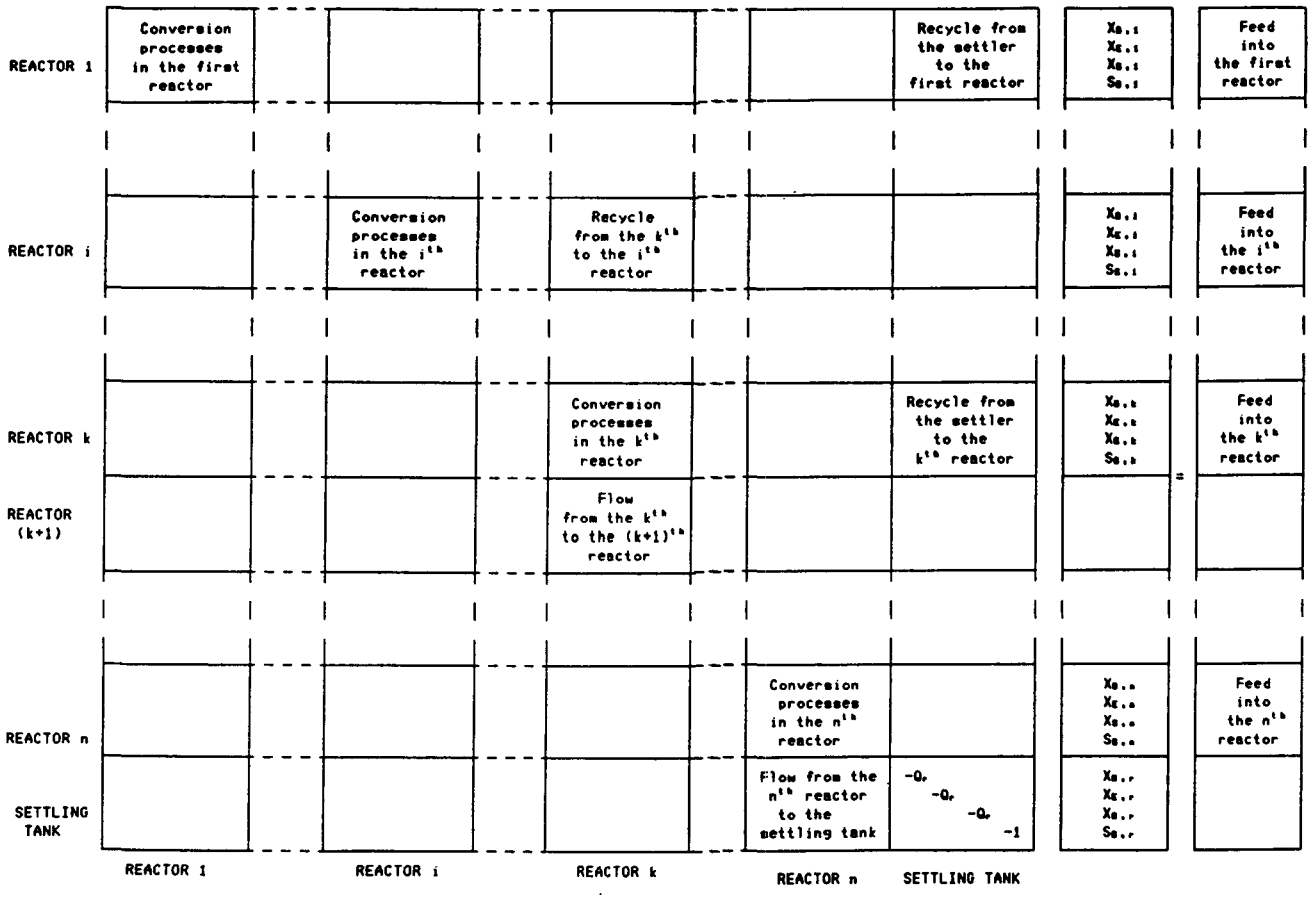


Figure 3
The steady state matrix representation of an n reactor system. Each block in the matrix corresponds to a submatrix of dimension (number of compounds).

represents flow "out of" that reactor. The horizontal position of the submatrix represents flows "into" that reactor.

Solution to the steady state problem

The topography of the steady state matrix provides a graphical illustration of the salient features of the system. It also has specific implications for the nature of a suitable numerical solution procedure. The matrix presentation shows how the numerical problem has a very definite structure. This is dictated by the biological reaction processes as well as the system configuration, particularly the manner in which the series of reactors in the system are interlinked. A significant part of any solution technique is to convert all this structural information into a form in which it can be exploited to reduce computational effort in finding the solution.

The matrices resulting from flow-sheeting problems for systems comprising a number of units are often solved using techniques such as partitioning with precedence ordering and tearing (Westerberg *et al.*, 1979). These techniques involve considering each unit separately, and partitioning the matrix into a number of smaller submatrices (i.e. the diagonal "blocks" in our case) which are then solved individually. The most appropriate sequence in which to solve the individual units can be determined by a process of precedence ordering. In solving the individual units, we may require estimates of the values of the concentrations in streams from other units yet to be solved. Estimation of these concentrations is termed tearing of the system. As a result of this process of estimation, the solution procedure for the complete system of interlinked units is an iterative one. If the recycle flows are not particularly significant, then this approach is a suitable one. However, with biological systems, the recycle terms can be large, exerting a strong and often dominating influence on the system. Therefore, partitioning is not suitable. An appropriate solution procedure should handle the matrix as a single entity.

One of the significant features of the biological flow-sheeting problem is the fact that the steady state matrix is usually sparse. Although many solution methods have been developed which exploit the sparsity of a matrix, most of these rely on the matrix being symmetrical and diagonally dominant, for example, in analysis of structures. In our situation, this is not usually the case, and many of these approches are therefore not suitable.

Five different approaches have been evaluated for computing the solution to the set of non-linear algebraic equations of the form encountered within biological reaction systems. These are the methods generally used in chemical engineering flow-sheeting applications. The five techniques are discussed in detail, with examples, by Billing (1987) where particular consideration is given to their application to biological systems. A more formal and rigorous presentation of the methods can be found in a number of standard texts, for example Reklaitis (1983), Westerberg *et al.* (1979), Dennis and Schnabel (1983) and Johnston (1982). In this paper only a brief overview of the methods is presented. Algorithms for their application can be found in *Appendix B*. With each of these methods, an initial estimate of the state variables must be provided, and the technique is applied iteratively until convergence is reached.

Direct linearisation

One method of solving a set of non-linear equations is by direct linearisation. The complete set of non-linear equations is represented by an equivalent set of linear equations, which are then solved using exact methods. The process of representation requires approximation, and this gives rise to an iterative procedure in which the linear equations become an improved approximation to the non-linear equations as the solution is approached.

The linear representation of a non-linear biological reaction system is demonstrated in Fig. 2. This figure presents the set of eight non-linear equations for the case study problem. If numerical values are assigned to any state variables which appear in the A matrix, then the set of equations has been "linearised directly". By solving the linear problem $AX = B$, a new set of values for the state variables is determined. In the method of direct linearisation the new values are used to update the A matrix, and the procedure is repeated to give an improved approximation to the solution, and so on.

Linear approximations to non-linear terms in the mass balance equations can be formulated in a number of ways. In selecting the appropriate linearisation, a set of linear equations must be chosen which gives rise to a process of iteration that eventually converges. This is not always possible; some of the possibilities may actually diverge. In the situation where more than one set converges, it is the different rates of convergence from a range of starting values that will determine the selection. It is difficult to generalise about the rate of convergence, or about the region from which convergence will be possible. Generally, however, it is possible to construct some form of linear approximation which, from a starting point sufficiently close to the solution, will eventually converge to that solution.

Although the method of direct linearisation as described above was successfully applied to a variety of biological system configurations, particular drawbacks to its application should be noted:

- Non-linear terms in the equations must be linearised. This can require extensive mathematical manipulation before the method can be implemented.
- Some linear approximations lead to systems of equations which do not converge. Therefore, a certain amount of skill, and perhaps trial and error, is necessary in selecting suitable linearisations (see *Appendix A*).
- A set of linear equations must be set up for each system configuration and each biological model. Any changes to the model or the configuration will necessitate a complete reworking of the equations.

Successive substitution

Successive substitution is a fixed point iteration method which requires the rearrangement of the non-linear equations $f_n(x_n) = 0$ in the form $x_n = g_n(x_n)$. The current estimate of the solution is substituted into the functions $g_n(x_n)$ to provide updated values. Although the method of successive substitution has the advangage of being simple and straightforward in its application, certain drawbacks are apparent:

- A certain amount of mathematical manipulation is necessary before the method can be applied, as the equations need to be rearranged in a form suitable for the fixed point iteration.
- The convergence behaviour of the method depends on the form of rearrangement. Functions that display sensitivity to any of the state variables could become unstable and prevent the system from converging. Also, the rate of convergence is slow.
- Careful consideration needs to be given to the selection of starting values. The initial estimates of the state variables

often need to be very close to the solution in order to ensure convergence.

- The set of equations $x = g(x)$ is specific to both the biological model and the reactor/recycle configuration. Any changes to the model or configuration would necessitate a complete reworking of the equations.

The secant method of Wegstein

A drawback of the successive substitution method is that its rate of convergence is only linear. A number of "acceleration procedures" have been proposed in order to improve this rate. The most widely used is Aitken's (1925) "δ^2 acceleration" method, which uses linear extrapolation through two points generated initially by a successive substitution formula. The same idea was later "rediscovered" by Steffensen (1933) and even later by Wegstein (1958), and hence it is known under these various names (Sargent, 1981). The method is a one-dimensional acceleration method in which each variable is treated separately by driving it with a uniquely associated function. Interactions with other variables are consequently ignored at each iteration.

Reklaitis (1983) notes that Wegstein's method may encounter difficulties if the slope for any variables does not dominate the slopes associated with the other variables which have been neglected in deriving the method. In practice, testing the validity of this assumption would require the evaluation of all of the partial derivatives of the functions $g_i(x)$. Westerberg *et al.* (1979) comments that this method could suffer from instability in a multidimensional environment, since large acceleration factors are encountered in most problems. He suggests using the bounded Wegstein method with delay, which could involve applying the acceleration function only every few iterations.

In spite of the shortcomings of Wegstein's method, it remains a commonly used algorithm, and has been accepted as the "best" one-dimensional method available (Westerberg *et al.*, 1979).

Newton's method

Newton's method is a more sophisticated root-finding technique which overcomes the problems of the relatively slow and often unpredictable convergence properties of the successive substitution and Wegstein methods. It has a much improved rate of convergence, although this is at the computational expense of requiring values of the partial derivatives of the functions.

The method is based on the idea of approximating a set of non-linear functions of the form $f(x) = 0$ by local linear approximations with slopes given by the derivatives of the functions. These functions are then used in an iterative procedure that generates new, and hopefully better, approximations to the solution.

Newton's method requires the evaluation of the partial derivatives of each of the n functions with respect to each of the n variables in order to evaluate the Jacobian matrix. This means that, if Newton's method is used to solve the biological system equations, any changes to the model or to the process configuration would require the re-evaluation of all of the partial derivatives. To avoid this problem, a finite difference approximation of the Jacobian may be used. Each term is evaluated as follows:

$$\frac{\partial f_i(x)}{\partial x_j} = \frac{f_i(x_1, x_2, \ldots, x_j + \Delta x_j, \ldots, x_n) - f_i(x_1, x_2, \ldots, x_n)}{\Delta x_j} \quad (10)$$

where Δx_j = a small perturbation to x_j
$\approx 10^{-6} \cdot x_j$

Dennis and Schnabel (1983) show that, when the analytical Jacobian in Newton's method is replaced by a finite difference approximation, the quadratic convergence properties of Newton's method can be retained provided the functions are not too non-linear. In fact, for most problems, Newton's method using analytical derivatives and Newton's method using properly chosen finite differences are virtually indistinguishable.

A finite difference approach would not save on the major expense involved in evaluating the $n \times n$ partial derivative matrix — in fact this can be a more costly process than when analytical derivatives are used. It does, however, render a simulation program more generally applicable because of not requiring further analysis when changes occur in the functions as a result of adjustments to the biological model or the system configuration.

Newton's method is generally superior to the successive substitution method and the secant method of Wegstein. The major advantages and disadvantages of the method may be summarised as follows:

Advantages:

- it exhibits quadratic convergence properties;
- it has been found to be extremely efficient for problems that are near linear (Johnston, 1982); and
- when the finite difference approximation to the Jacobian is used, the method has a very general applicability. Any changes to the biological model or system configuration can thus be easily incorporated into a computer program without having to re-evaluate the partial derivatives.

Disadvantages

- in regions where the Jacobian is nearly singular, the method can behave erratically; and
- implementation of the method is a costly exercise, as both the functions and the Jacobian matrix need to be recalculated at every iteration.

Broyden's method

Broyden's algorithm (Broyden, 1965; 1969) is a modification of Newton's method that was designed specifically to reduce the number of function evaluations necessary in finding a solution to a set of non-linear simultaneous algebraic equations. It is one of a whole class of methods which may be termed "quasi-Newton methods". These are techniques based on the idea of approximating the Jacobian in order to avoid the computational effort required to evaluate it fully. For a one-dimensional environment, the secant method fills this role since it is based on approximating the derivative, $f'(x)$ of a single function $f(x) = 0$. Hence, any quasi-Newton method may be regarded as an n dimensional extension of the secant method. For any of these techniques, it is only the method for approximating the Jacobian matrix that will be different, the rest of the Newton algorithm remains unchanged.

Broyden's method is particularly suited to flow-sheeting type problems and has been analysed widely in the chemical engineering literature. The major advantage of the method, and

TABLE 1
KINETIC AND STOICHIOMETRIC PARAMETERS USED IN THE CASE STUDIES. THE BIOLOGICAL MODEL IS PRESENTED IN PART 1.

Symbol	Value	Units
Kinetic parameters:		
$\hat{\mu}$	4,0	d^{-1}
K_S	5,0	g COD m^{-3}
b	0,62	d^{-1}
K_H	2,2	g COD(g cell COD)$^{-1}$ d^{-1}
K_X	0,15	g COD (g COD)$^{-1}$
Stoichiometric parameters:		
Y_H	0,666	g COD cell yield (g COD utilised)$^{-1}$
f	0,08	

indeed, the reason for its development, is that it preserves many of the positive characteristics of Newton's method whilst only requiring roughly half the computational effort with respect to the Jacobian evaluation.

Certain potential drawbacks to the method should be noted:

- The convergence rate of Broyden's method is superlinear but not of the same order as Newton's method. Therefore, more iterations will be required than for Newton's method.
- A good approximation to the Jacobian matrix is necessary to seed the method, otherwise it may fail to converge.
- The method can behave erratically in regions where the partial derivative matrix is nearly singular.
- In many flow-sheeting applications, (for example, the biological system) the Jacobian matrix is sparse. In updating the approximation to the Jacobian using Broyden's method, non-zero terms (of very small magnitude) may be introduced into the approximating matrix, $\mathbf{B}_{(p)}$, at points where the true Jacobian, $\mathbf{J}_{(p)}$, would contain zeros. This has certain implications because part of Broyden's method involves solving a set of linear equations incorporating $\mathbf{B}_{(p)}$ (see Step 3 of the algorithm in *Appendix B*). Solution methods such as Gaussian elimination with pivotal rearrangement will now require more computation at this step because the matrix has become less sparse. This will partially negate the benefit of fewer function evaluations required to set up the Jacobian.

Selecting case studies for analysis

The five numerical techniques are now evaluated and compared in application to solution of a range of specific steady state biological system problems. Before evaluating the numerical techniques, two aspects must be specified. The first is that a biological model must be selected. The second is the selection of a range of reactor configurations to be considered in case studies. These configurations should incorporate the characteristics of the various types of flow sheet encountered in practice.

Selection of a biological model

Considerations that are involved in the selection of a biological model have been referred to in Part 1. The model selected for the purpose of evaluating the various numerical techniques in this study is a restricted version of the IAWPRC Task Group model for the activated sludge process. Only aerobic heterotrophic growth phenomena have been included, as shown in the model matrix of Table 2 of the Part 1 paper.

The values for the kinetic and stoichiometric parameters that have been used in the simulations are in line with those selected by the IAWPRC Task Group and described by Dold and Marais (1985); Table 1 summarises these parameters.

Selection of the reactor configurations and operating conditions

An evaluation of the suitability of the different numerical methods needs to be carried out in the context of the types of situation that the methods will encounter in practice. For example, in a waste-water treatment application, a numerical technique would need to handle problems stemming from a wide variety of system configurations and operating conditions. These may range from a simple single reactor process operated at short sludge age to a more complex system incorporating numerous reactors in series linked by both forward and recycle flows and operated at a long sludge age.

Five configurations were selected as case studies for evaluating the numerical methods. These specific configurations with associated recycles and operating conditions were chosen as they incorporate facets of a spectrum of systems encountered in biological waste-water treatment. Although specific to activated sludge systems, the configurations include certain features general to most biological reaction systems. Table 2 summarises the details of the system configurations and operating conditions for the five test cases. The configurations are shown diagrammatically in Fig.4. Because a limited biological model was used for the study, no provision is made for the usual phenomena encountered with unaerated reactors e.g. denitrification. Hence, all the reactors in the test case configurations are aerated even though unaerated reactors would usually be incorporated in certain of the configurations; for example, the UCT process (Case 5). Aspects particular to the five selected configurations are as follows:

Case Study 1: The simplest configuration that could be encountered in an activated sludge process. It consists of an aerated reactor and a settling tank. The underflow from the settling tank is recycled to the reactor. The configuration is the same as that introduced in Part 1.

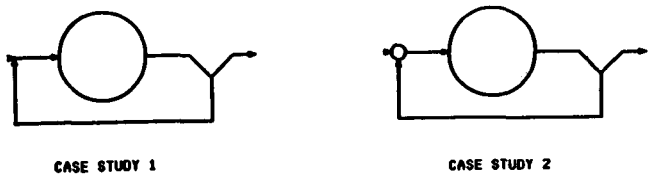
Case Study 2: A "selector reactor" configuration utilised in the control of sludge bulking. It consists of two aerobic reactors in series, the first reactor being very much smaller than the second (volume ratio 1:32). All the feed enters the first reactor, as does the underflow from the settling tank.

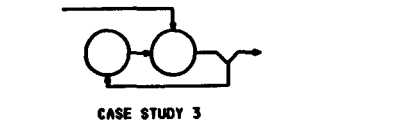
Case Study 3: A "contact stabilisation" process, in which all the feed enters the "contact" reactor, which is the second of two aerobic reactors in series. The underflow from the settling tank is recycled to the first reactor.

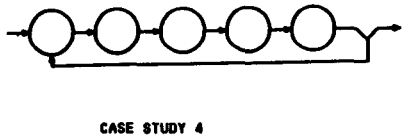
Case Study 4: Five aerated reactors in series, with all the feed entering the first reactor. Underflow from the settling tank is recycled to the first reactor in the series.

TABLE 2
SUMMARY OF SYSTEM CONFIGURATIONS AND OPERATING CONDITIONS FOR THE CASE STUDIES SHOWN IN FIG. 4.

		Case 1	Case 2	Case 3	Case 4	Case 5
Configuration		Single reactor	Selector reactor	Contact stabilisation	Series reactors	UCT process
Reactor volumes (ℓ)	Reactor 1	8	0,25	12	1,5	2
	Reactor 2		8	2	1,5	3
	Reactor 3				1,5	6
	Reactor 4				1,5	
	Reactor 5				1,5	
Sludge age (d)		3	3	6	5	20
Feed rate (ℓd^{-1})		20	20	36	20	10
RAS recycle rate (ℓd^{-1})		20	20	72	20	10
A recycle	From reactor					3
	To reactor					2
	Rate (ℓd^{-1})					40
B recycle	From reactor					2
	To reactor					1
	Rate (ℓd^{-1})					10
Influent COD		500 $g.m^{-3}$ [S_S = 100 $gCOD.m^{-3}$; X_S = 400 $gCOD.m^{-3}$]				







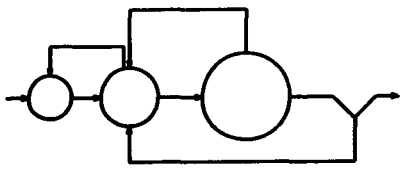


Figure 4
The case study reactor and recycle configurations.

Case Study 5: A "UCT process" with three reactors in series. The distinguishing feature of the configuration is the arrangement of recycles between reactors. Mixed liquor recycles are taken from the third to the second and from the second to the first reactors. Underflow from the settling tank is recycled to the second reactor in the series. All the feed enters the first reactor.

Criteria for evaluating numerical methods

The general characteristics, advantages and disadvantages of the five selected numerical methods have been outlined earlier. In attempting to select a numerical method appropriate to a particular application, the main criteria that need to be satisfied are:

- the method must offer a reasonable guarantee of convergence to a solution from the specified initial values; and
- it should converge as "efficiently" as possible.

The "efficiency" of a method is a measure of how much computational effort is required to calculate a reasonable approximation to a solution. Two aspects need to be considered here:

- the number of iterations required before a method converges; and
- the amount of computation required to perform each iteration.

In general, when comparing numerical methods, it has been found that those that are superior with respect to guarantee of convergence will usually be slow. Conversely, the faster numerical methods are more likely to diverge (Johnston, 1982). Consequently, in choosing a numerical method, a decision has to be made as to which qualities are more important at any one time. For example, in situations where the location of the solution is completely unknown, a slow method, but one which is unlikely

to diverge despite crude initial estimates, will be preferred.

Implementation of the numerical methods

A computer program was written to test the different numerical techniques. The simulation program was written in Turbo Pascal (Borland, 1985), a language which was found to be suitable for use with an IBM PC or compatible machine. The program was specific to the selected biological model but allowed flexibility in the choice of system configuration and operating conditions. Each numerical technique was written as a module, which was then inserted in its entirety into the simulation program. This was done in an effort to eliminate any bias that might be introduced by different programming codes affecting the relative efficiencies of any of the methods. That is, computer codes for setting up and evaluating reaction rates, reactor input and output terms, etc., were common to all the methods.

The solution to the steady state problem is reached when the set of mass balance equations, $\mathbf{f(x)} = 0$, is satisfied within a specified convergence criterion. In converging to the solution, a measure of the accuracy of the current values at each iteration is given by the magnitude of the functions. To have some global measure which will embrace all the state variables, the convergence criterion was formulated in terms of

$$\Sigma[f_i(x)]^2 \qquad (11)$$

It was assumed that a solution had been reached when this summation was less than a certain error tolerance. In choosing the magnitude of this tolerance, a balance between efficiency and reliability should be maintained. The convergence tolerance must be reasonably small in order to prevent early termination. Choosing too small a value, however, can delay termination unnecessarily. The selection of 10^{-3} as the convergence tolerance was found through practice to result in accurate solutions. At the same time, it is not so stringent that the numerical methods take unacceptably long to satisfy it.

Two features incorporated in the computer program which are not specific to discussion of the numerical methods, but which may be of interest are:

- calculation of sludge wastage rate in accordance with a specified sludge age: and
- the initial estimates of the solution.

These aspects are discussed in *Appendix C.*

Case study results and discussion

The five numerical methods discussed were applied to each test case. Each method was allowed to run until convergence was achieved, and subsequently assessed in terms of:

- how long it took to reach an acceptable solution from a standard set of starting values; and
- how many iterations were required for the given convergence criterion. (All the results were obtained using Turbo Pascal Version 3.0 running on a standard IBM PC operating at 4,77 MHz. The configuration did not include an 8087 maths co-processor).

The results for each method and test case are presented in Table 3. Certain overall aspects are apparent from the results. These are discussed below. In addition, a more detailed comparison of some of the numerical methods was carried out to assess the actual manner in which different techniques approached the solution. For certain of the techniques, this evaluation involved examination of potential instability problems. For others, an assessment was made regarding exactly how much computational energy was expended at each point in an iteration loop. This was to develop a greater understanding of the behaviour of each method in its practical implementation, and to establish a qualitative feel for more than just the convergence properties of a particular technique. The more detailed comparison of methods is discussed in the following sections.

TABLE 3
TEST CASE RESULTS

	METHOD									
	Direct linearisation		Successive substitution		Wegstein's method		Newton's method		Broyden's method	
	Its.	Time	Its.	Time	Its.	Time	Its.	Time	Its.	Time
CASE 1 Single reactor	16	12,9	108	41,8	133	54,1	4	5,5	5	7,0
CASE 2 Selector reactor	16	24,8	256	143,5	258	175,5	4	12,2	10	32,0
CASE 3 Contact stabilisation	12	18,3	619	347,0	576	391,4	4	12,2	8	25,5
CASE 4 Five-in-series	14	62,6	1 663	2 945,2	1 605	3 004,9	3	36,4	7	49,6
CASE 5 UCT process	10	25,1	606	501,5	615	605,4	4	24,1	8	49,2

where Its. = number of iterations
Time = time in seconds

General comments

- All the methods converged to the same solution for all the test cases. However, it should be remembered that for the direct linearisation approach, successive substitution and Wegstein's method, the set of equations had to be arranged in particular ways in order for convergence to be attained. Some forms of rearrangement of the equations did not converge from the specified initial conditions.
- The test case results bear out a generally expected trend of convergence characteristics. The successive substitution and Wegstein methods, exhibiting only linear convergence rates, needed significantly more iterations to converge to a solution. Newton's method, with a quadratic rate of convergence, requires very few iterations to attain convergence. Broyden's method, which has a convergence rate that is superlinear, although not quadratic, required approximately twice as many iterations to converge as did Newton's method.
- For all the case studies, Newton's method was always the fastest to converge. Despite the fact that each iteration in this method involves a complete re-computation of the Jacobian matrix, the computational time expended per iteration is not excessive. In addition, the case studies verify the advantage of the quadratic convergence rate, as Newton's method requires significantly fewer iterations than any of the other methods to reach a solution. This seems to be irrespective of the complexity of the configurations, as the method consistently required only three or four iterations to converge.
- Broyden's method generally required approximately twice as many iterations as Newton's method. This is in agreement with the general convergence characteristics of quasi-Newton methods i.e. those using an approximation to the Jacobian matrix. However, the time taken to reach convergence by the two methods should then be approximately equal, given that Broyden's method requires only half the number of function evaluations to estimate the Jacobian. Examination of Table 3 shows that, in practice, this does not occur. In fact, Broyden's method consistently required longer than Newton's method to converge. This aspect is discussed in more detail later.
- Both the methods of Wegstein and successive substitution were found to perform consistently poorly for all the test cases. This was not entirely unexpected. The fact that both are simple to implement and require very little computational effort per iteration is counterbalanced by inferior rates of convergence.
- Successive substitution and Wegstein's method may appear to perform disproportionately poorly for the five-in-series reactor configuration of Case 4. On consideration, however, this result is to be expected. Both these methods involve fixed point iteration in which each state variable is modified without regard for the simultaneous changes in other variables at each iteration. In contrast, Newton's method, for example, accounts for this "simultaneity" via the partial derivatives in the Jacobian. Therefore, when successive substitution or Wegstein's method is applied to a long train of reactors with no internal recycles such as Case 4, inaccuracies in the initial estimates "work through" the system slowly. The performance of these methods is improved relatively when internal recycles are included in the configuration as in Case 5 for example. These recycle links in effect partially account for the interaction between the state variables which is not directly considered with successive substitution or Wegstein's method. To explain this, consider a certain compound in a two-reactor system where the respective concentrations are denoted by x_1 and x_2. If there is no recycle from reactor 2 to reactor 1, then the mass balance equation for x_1 does not contain the variable x_2. As a result, in the fixed point iteration step for x_1 the influence of the variable x_2 is disregarded. In contrast, if there is a recycle from reactor 2 to 1, then the variable x_2 is incorporated in the mass balance equation for x_1. In this case, cognisance is given to x_2 when iterating for x_1.
- The performance of Wegstein's acceleration method compared to that of successive substitution was surprisingly poor. The lack of improvement over successive substitution indicates that Wegstein's method is not an appropriate acceleration technique for these types of functions. A more detailed examination of the relative merits of successive substitution and Wegstein's method is presented later.
- The direct linearisation method produced very favourable results for all the test cases. Accurate solutions were achieved, and convergence was both rapid and efficient. In fact, for Case 4, its performance is almost comparable to that of Newton's method. The efficiency of the method also seems to be relatively independent of the complexity of the system configuration and operating conditions. In fact, fewer iterations and computational effort were required to reach a solution in Case 5 — the most complex configuration — that in Case 1 — the simplest case study. The reason for the success of the direct linearisation method is that the functions for the biological system under consideration are not particularly non-linear in the regions of interest, and thus the linearised functions give a good approximation to the non-linear equations. However, as noted earlier, a severe drawback of the method is the prior skill and mathematical manipulation that are necessary before the method can be implemented.

Comparison of the Wegstein and successive substitution methods

These two numerical methods both reached convergence for all situations, although in Case 4 many iterations were required before the tolerance was eventually satisfied. The amount of computational time expended per iteration for both techniques is near equal, although Wegstein's method generally takes slightly longer than successive substitution for each loop. This is to be expected, as the methods are identical except for the relatively inexpensive additional calculation of acceleration factors and checks on these that are introduced with Wegstein's method.

The number of iterations required by each method in order to attain convergence was found to be near equal, although Wegstein's method consistently required a few more iterations than did successive substitution. This is contrary to what was expected, as Wegstein's method was originally implemented to accelerate the rate of convergence of successive substitution. This result demanded further investigation.

To examine the phenomenon more fully, various modifications of Case Study 1 were considered. In one of these modifications, the sludge age was changed to 30 d instead of 3 d (all other parameters were maintained as before) and both methods were re-tested. Fig. 5 shows the trend observed in the sum of the squares of the function values, which was used as the stopping criterion for both methods. As expected, Wegstein's acceleration method moves to the region of solution more rapidly than does successive substitution. What is surprising, however, is that ultimately Wegstein's method requires more iterations to reduce the error to within the specified tolerance. On closer examination it was apparent that the reason for this behaviour was slight in-

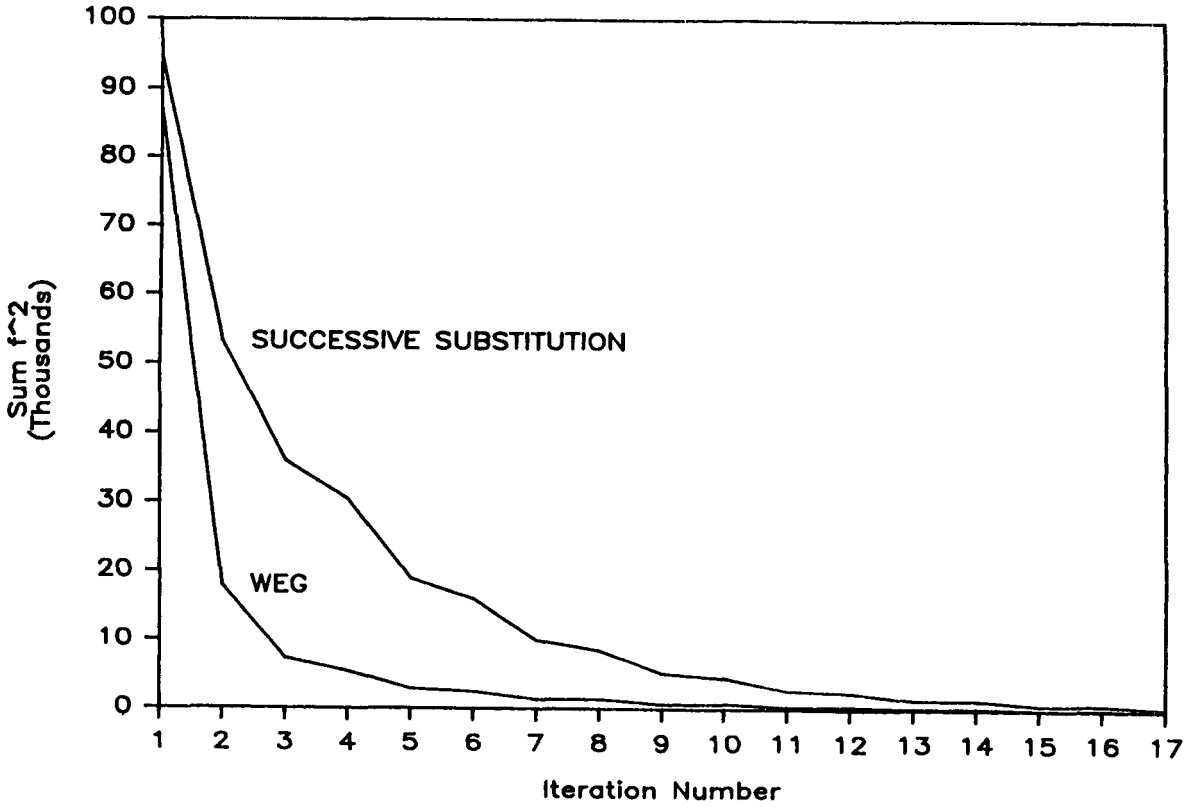


Figure 5
Comparison of Wegstein and successive substitution methods for Case Study 1 (Sludge age = 30 d).

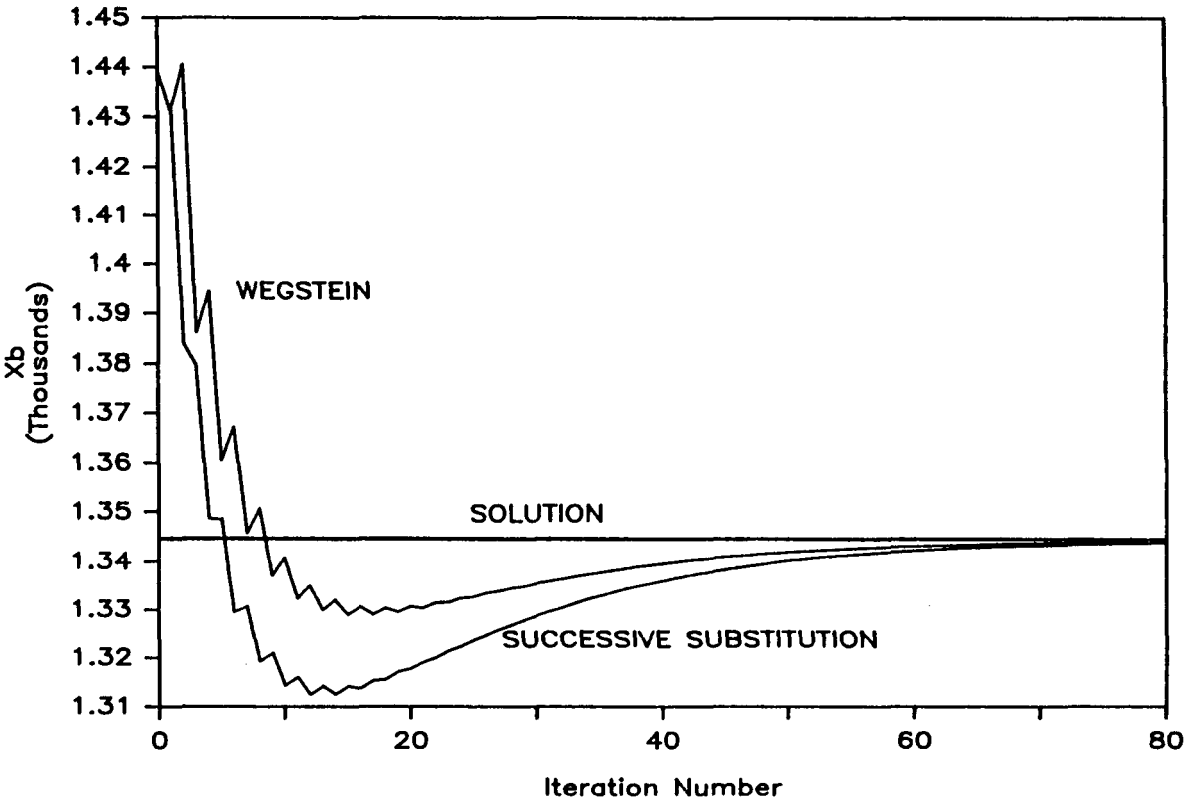


Figure 6
Comparison of X_B values for Wegstein and successive substitution methods for Case Study 1.

stability introduced by Wegstein's method. This is not readily noticeable in the plot of Fig. 5.

Fig. 6 shows the path followed by the concentration of particulate biomass, X_B, in approaching the solution for Case Study 1. Again, with Wegstein's method, the value of X_B initially converges more rapidly to the solution, with less overshoot, as would be expected. However, although the general trend introduced by the acceleration is towards a more "damped" path, the individual points on the curve have more of a tendency to oscillate than those generated by the successive substitution technique. When the solution is approached, this instability prevents the convergence criterion from being satisfied.

It appears from the results that there would perhaps be some merit in using the approach suggested by Westerberg *et al.* (1979); that is, applying Wegstein's method at intervals. This would presumably accelerate the successive substitution whilst avoiding the instabilities associated with Wegstein's method.

Comparison of Broyden's and Newton's methods

The relative convergence rates of these two methods bear out the expected trends: Broyden's method does not converge as rapidly or as efficiently as Newton's method. However, the fact that Broyden's method is so computationally expensive merits further investigation.

Case Study 2 was used to examine the details of how the computational energy for each iteration in the methods was distributed. Table 4 shows this "division of effort" for the second, third and fourth iterations. Both techniques took approximately 3 s to complete each iteration. The major components are the time required to set up the Jacobian (or its approximation) and the time required to solve the resulting set of linear equations.

- Previous discussion indicated that the major advantage of Broyden's method is that, to set up the Jacobian approximation, it requires fewer function evaluations at each iteration than Newton's method and thus should require less time per iteration. An examination of the results in Table 4 shows that the time spent by Broyden's method in updating the approximation to the Jacobian is roughly half that spent by Newton's method in re-evaluating the complete matrix of partial derivatives (0,98 s versus 2,02 s). This is to be expected as half the number of function evaluations are required when using Broyden's method.

- The major expense in Broyden's method is the disproportionate time spent in solving the resulting system of linear equations by the Gauss elimination procedure used here. In Broyden's method, the Gauss elimination takes more than twice as long to implement as it does in Newton's method (1,72 s versus 0,72 s). The reason for this is that small non-zero terms are introduced into the matrix by Broyden's updating formula in locations that would usually contain zeros in the Jacobian. This effectively reduces the sparsity of the Broyden matrix and severely hampers the operation of the Gaussian technique, which relies on pivotal rearrangement for its efficiency.

From the results above, it is apparent that, if the saving in the number of function evaluations in Broyden's method is to be exploited, then attention should be paid to the method used to solve the linear equations. Perhaps this could be improved by using some specialised matrix technique, rather than the Gaussian elimination used here.

TABLE 4
COMPARISON OF TIME PER ITERATION AS EXPENDED BY BROYDEN'S AND NEWTON'S METHODS FOR ITERATIONS 2 TO 4 IN CASE STUDY 2.

Case study		Time (s)	
Iteration number		Broyden's method	Newton's method
2	Gauss	1,71	0,72
	matrix	0,98	2,04
3	Gauss	1,76	0,71
	matrix	0,99	1,98
4	Gauss	1,70	0,72
	matrix	0,98	2,03

where Gauss = time spent solving linear equations
Matrix = time spent updating the matrix

General conclusions

- The method of direct linearisation, although performing relatively efficiently for these test cases, is not a suitable technique for general use in a simulation program. Thus, although the preliminary analysis seems to have paid off in the satisfactory performance of the method, the requirements of the program that it be as generally applicable as possible eliminates direct linearisation from the possibilities that can seriously be considered.
- The methods of Wegstein and successive substitution, although simple and robust, are inappropriate due to their slow rates of convergence. Instability problems may be encountered in their implementation and, as a result, convergence cannot always be guaranteed.
- The poor performance of Wegstein's method in comparison to that of successive substitution could perhaps be eliminated by applying Wegstein only at selected intervals, as suggested by Westerberg *et al.* (1979).
- Broyden's method converges to a solution in comparatively few iterations. The computational effort required to set up the Jacobian approximation is roughly half that required by Newton's method in setting up the true Jacobian. However, the method as implemented here requires an excessive amount of computational effort per iteration. The major portion of this effort is concentrated in the solution of the system of linear equations. Perhaps this bottleneck could be removed by employing a specialised sparse matrix technique.
- Of the methods evaluated, Newton's appears to be the most favourable. In addition, the use of a finite difference approximation to the Jacobian matrix renders it a generally suitable technique for the biological flow-sheeting systems under consideration.

Appendix A

Multiple possibilities for direct linearisation.

$$f(X,Y) = X \times Y \qquad (A.1)$$

To create a linear approximation, the first two terms of a Taylor's expansion about the point (X_0, Y_0) can be formulated. The point (X_0, Y_0) should lie in the region of interest. This will yield:

$$f(X,Y) = f(X,Y)_{(X0,Y0)} + \frac{\partial f}{\partial X}\Big|_{(X0,Y0)} \Delta X + \frac{\partial f}{\partial Y}\Big|_{(X0,Y0)} \Delta Y \qquad (A.2)$$

with $\Delta X = X - X_0$ and $\Delta Y = Y - Y_0$.

This simplifies to:

$$f(X,Y) = X_0 Y + X Y_0 - X_0 Y_0 \qquad (A.3)$$

Eq. (A.3) represents one possible linear approximation to Eq. (A.1). There are, however, other possible linear equations that could be used to approximate the non-linear equation. Other options can usually be developed from a further examination of Eq. (A.2) as well as utilising additional information as regards the point (X_0, Y_0). If, for example, $\Delta X \approx \Delta Y, X_0 \approx 1\,000$ and $Y_0 \approx 1$, then the differing orders of magnitude of the two terms could be used to make an important simplifying assumption to Eq. (A.2). In this case, consider the contribution of the terms

$$\frac{\partial f}{\partial X}\Big|_{(X0,Y0)} \Delta X = Y_0 \Delta X \approx 1.\Delta X$$

and

$$\frac{\partial f}{\partial Y}\Big|_{(X0,Y0)} \Delta Y = X_0 \Delta Y \approx 1\,000 \Delta Y$$

When $\Delta X \approx \Delta Y$, the first term will be negligible in comparison to the second. Therefore, Eq. (A.2) could be reduced to:

$$f(X,Y) = f(X,Y)_{(X0,Y0)} + \frac{\partial f}{\partial Y}\Big|_{(X0,Y0)} \Delta Y \qquad (A.4)$$

which yields the linear approximation

$$\begin{aligned} f(X,Y) &= X_0 Y_0 + X_0 (Y - Y_0) \\ &= X_0 Y \end{aligned} \qquad (A.5)$$

The approach leading the Eq. (A.5) is one that can often be used successfully in the direct linearisation method for biological systems. This simplification is possible because these systems often incorporate particulate compounds at high concentrations and soluble compounds at low concentrations. The reason for interest in this approach is that it often leads to simpler equations (compare Eqs. (A.3) and (A.5)).

To illustrate applying this simplification in biological systems, let us return to the case study of the single reactor plus settler. Consider the non-linear term in Eq. (1):

$$\frac{\hat{\mu} S_S X_B}{(K_S + S_S)} V \qquad (A.6)$$

A linear approximation to this term can be created in a number of ways. These include, amongst others, the following two possibilities:

- A complete Taylor's expansion about the point (X_{B0}, S_{S0}).
- The same Taylor's expansion could be employed, but the resulting linearisation could be further simplified by using the fact that we have additional information as regards the nature of certain of the terms in the equation.

For the second option, we note that, for every unit of soluble substrate (S_S), utilised, Y units of biomass (X_B) are created. Because $Y \approx 0{,}66$, we can assume that ΔX_B and ΔS_S are of similar magnitude i.e.

$$\Delta X_B \approx \Delta S_S \qquad (A.7)$$

In addition, in the situations encountered in practice, the concentration of S_S is generally low (≈ 1) and the concentration of X_B is generally high (≈ 1000). Thus, the non-linear term of Eq. (1) could be linearised using the simplifying assumptions outlined for Eq. (A.5) in the region (X_{B0}, S_{S0}) as follows:

$$f(X_B, S_S) = \frac{\hat{\mu} X_{B0} V}{(K_S + S_{S0})} S_S \qquad (A.8)$$

This is the approach that has been used in the method of direct linearisation employed in the simulation program here. Similar simplifying assumptions may be applied to all the non-linear terms in the mass balance equations.

Appendix B:

Algorithms for the numerical techniques.

The *direct linearisation* algorithm

Step 1: Set up linear approximations for all the non-linear terms in the n mass balance equations $\mathbf{f(x)} = 0$.

Step 2: Arrange the linearised equations into the form $AX = B$.

Step 3: Initialise the A matrix with seed values of the state variables.

Step 4: Find new values of the state variables, X, by establishing the solution to the matrix problem
$A X = B$
using Gaussian elimination.

Step 5: Test for convergence.
If the convergence criterion is satisfied, then terminate the iteration. Otherwise, insert the new values of the state variables into the A matrix and return to *Step 4*.

The successive substitution algorithm

Step 1: Arrange the n equations $\mathbf{f(x)} = 0$ in the form $\mathbf{x} = \mathbf{g(x)}$.

Step 2: Select an initial estimate for the state variables, $\mathbf{x}_{(0)}$, and a suitable convergence criterion.

Step 3: Calculate
$\mathbf{x}_{(p+1)} = \mathbf{g}(\mathbf{x}_{(p)})$

Step 4: Test for convergence
If $\Sigma |\mathbf{g}(\mathbf{x}_{(p)}) - \mathbf{x}_{(p+1)}|^2 <$ convergence tolerance
then terminate the iteration.
Otherwise, replace $\mathbf{x}_{(p)}$ by $\mathbf{x}_{(p+1)}$ and return to *Step 3*.

The Wegstein algorithm

Step 1: Arrange the n equations $f(x) = 0$ in the form $x = g(x) = 0$.

Step 2: Select an initial estimate for the state variables, $x_{(0)}$, and a suitable convergence criterion, and upper and lower bounds for t. ($|t_{upper}| = |t_{lower}| = t_{max}$.)

Step 3: Calculate

$$x_{(1)} = g(x)_{(0)}$$

Step 4: Calculate the slopes

$$m = \frac{g(x)_{(p)} - g(x)_{(p-1)}}{x_{(p)} - x_{(p-1)}}$$

Step 5: Calculate

$$t = \frac{1}{(1-m)}$$

If $|t_i| > t_{max}$ then $t_i = t_{max}$

Step 6: Calculate

$$x_{(p+1)} = (1-t) \cdot x_{(p)} + t \cdot g(x)_{(p)}$$

Step 7: Test for convergence

If $\Sigma|g_i(x_{(p)}) - x_{i(p+1)}|^2 <$ convergence tolerance then terminate the iteration.

Otherwise, replace $x_{(p)}$ by $x_{(p+1)}$ and return to *Step 4*.

The Newton algorithm

Step 1: Express the non-linear functions in the form $f(x) = 0$. Select initial estimates for the roots $x_{(0)}$ and a suitable convergence criterion.

Step 2: Evaluate $J(x)_{(p)}$

Step 3: Calculate $x_{(p+1)} = x_{(p)} - [J(x)_{(p)}]^{-1} \cdot f(x)_{(p)}$ as follows:

(i) Solve the set of linear equations:

$$J(x)_{(p)} \cdot h_{(p)} = -f(x)_{(p)}$$

for $h_{(p)}$

(ii) $x_{(p+1)} = x_{(p)} + h_{(p)}$

Step 4: Test for convergence

If $\Sigma|f_i(x_{(p)})|^2 <$ convergence tolerance then terminate the iteration.

Otherwise, replace $x_{(p)}$ by $x_{(p+1)}$ and return to *Step 2*.

The Broyden algorithm

Step 1: Express the non-linear functions in the $f(x) = 0$. Select initial estimates for the roots $x_{(0)}$, an initial approximation to the Jacobian matrix, $B_{(0)}$ and a suitable convergence criterion.

Step 2: Solve the set of linear equations

$$B_{(p)} \cdot s_{(p)} = -f(x)_{(p)} \text{ for } s_{(p)}$$

Step 3: Calculate

$$x_{(p+1))} = x_{(p)} + s_{(p)}$$
$$y_{(p)} = f(x)_{(p)} - f(x)_{(p-1)}$$

Step 4: Calculate

$$B_{(p)} = B_{(p-1)} + \frac{1}{s^T_{(p)} s_{(p)}}(y_{(p)} - B_{(p-1)} \cdot s^T_{(p)})\, s_{(p)}$$

Step 5: Test for convergence

If $\Sigma|f_i(x_{(p)})|^2 <$ convergence tolerance then terminate the iteration.

Otherwise, replace $x_{(p)}$ by $x_{(p+1)}$ and return to *Step 2*.

Appendix C:

Calculation of the wastage rate, q_w

In setting up the simulation problem, sludge age (solids retention time, SRT) is specified as an operating parameter. This is defined as:

$$\text{Sludge age} = \frac{\text{Mass of sludge in the system}}{\text{Mass of sludge wasted per day}}$$

$$= \frac{\text{Mass of sludge in the system}}{q_w \cdot C_n} \qquad \text{(C.1)}$$

where C_n = concentration of solids in the n^{th} reactor.

In this study, it is assumed that sludge wastage always comes from the last reactor (n^{th}) *in the series i.e. hydraulic control of sludge* age. If all the feed enters the first reactor, then the settling tank underflow is recycled to the first reactor and the concentration of sludge from reactor to reactor is more or less constant. In this case, the required sludge wastage rate, q_w, to maintain a specified sludge age is given by:

$$q_w = \frac{\text{Total volume of system}}{\text{Sludge age}} \qquad \text{(C.2)}$$

When specifying sludge age as an operating parameter, a problem in calculating the required wastage rate occurs where the concentration of sludge varies from reactor to reactor. This will be encountered when the feed enters, for example, the second reactor in the contact stabilisation process (Case 3) or where the settler underflow is not recycled to the first reactor as in the UCT process (Case 5). The problem arises because the wastage rate can only be determined once the distribution of sludge between the reactors and particularly the concentration in the last reactor is known. However, this concentration is influenced by the wastage rate itself. To overcome this problem, the following iterative procedure was employed for calculating wastage rate once the reactor configuration and feed and recycle rates had been specified:

Step 1: Assume that a particulate inert tracer is introduced into the influent at some constant concentration. This fixes the mass of inert tracer in the system for a given sludge age. Mass of tracer = Daily inflow × concentration of tracer in the influent × sludge age.

Step 2: Provide an initial estimate of the wastage rate from Eq. (C.2).

Step 3: For the selected q_w, solve the set of mass balance equations describing the concentration of tracer in each reactor and in the underflow recycle. This is a set of linear equations.

Step 4: Recalculate the wastage rate from Eq. (C.1).

Step 5: Test for convergence.

If convergence is achieved, then terminate the iteration. Otherwise, return to *Step 3*.

Initial estimates of the solution

To initiate any of the iterative numerical procedures, an estimate of the solution is required. If these estimates are not accurate, it is possible that the numerical method will not converge to the correct solution. Also, the less accurate the initial estimate, the greater the number of iterations that will be required to attain convergence.

In the computer program, initial estimates of the state variables are based on steady state waste-water treatment theory (WRC, 1984) and on empirical estimates. The simulation program estimates the masses of the active organism (X_B) and endogenous residue (X_E) fractions from this theory, based on the effective steady state endogenous respiration rate. The masses of these particulate materials, biomass, X_B, and endogenous residue, X_E, are distributed amongst the reactors in accordance with the distribution of the inert particulate tracer as discussed above. The initial concentration of particulate substrate, X_S, in each reactor is assumed to always be 10 per cent of X_B, and the initial estimate of the soluble substrate, S_S, is always taken as 1,5 g COD m^{-3}.

References

AITKEN, A.C. (1925) On Bernoulli's numerical solution of algebraic equations. *Proc. Roy. Soc. Edinburgh* **46** 289.

BILLING, A.E. (1987) Modelling techniques for biological systems. M.Sc. Thesis, Departement of Chemical Engineering, University of Cape Town.

BORLAND INTERNATIONAL INC. (1985) *Turbo Pascal Version 3.0.* Borland International Inc. Scotts Valley. CA.

BROYDEN, C.G. (1965) A class of methods for solving non-linear simultaneous equations. *Math Comp.* **19** 577-593.

BROYDEN, C.G. (1969) A new double rank minimisation algorithm. *AMS Notices* **16** 670.

DENNIS, J.R. (Jnr) and SCHNABEL, R.B. (1983) *Numerical Methods for Unconstrained Optimisation and Non-linear Equations.* Prentice Hall Inc. Englewood Cliffs. New Jersey.

DOLD, P.L. and MARAIS, G.v.R. (1985) Evaluation of the general activated sludge model of the IAWPRC task group. *Wat. Sci. Tech.* **18**(6) 63-89.

JOHNSTON, R.L. (1982) *Numerical Methods, a Software Approach.* John Wiley and Sons. Canada.

REKLAITIS, G.V. (1983) *Introduction to Material and Energy Balances.* John Wiley and Sons. New York.

SARGENT, R.W.H. (1981) A review of methods for solving non-linear algebraic equations. In R.S.H. Mah and W.D. Sieder (Editors). *Foundations of Computer-Aided Chemical Process Design.* Engineering Foundation. New York. 1 22-76.

STEFFENSEN, J.F. (1933) Remarks on iteration. *Skand. Aktuar. Tidskr* **16** 64-72.

WEGSTEIN, J.H. (1958) Accelerating convergence of iterative processes. *Comm. ACM* 1 (6) 9.

WESTERBERG. A.W., HUTCHINSON, H.P., MOTARD. R.L. and WINTER, P. (1979) Process flow-sheeting. Cambridge University Press.

WRC (1984) *Theory, Design and Operation of Biological Nutrient Removal Activated Sludge Processes.* Water Research Commission, P.O. Box 824, Pretoria 0001, South Africa.