LITERATURE CITED

Baldwin, R. R. and D. Brattan, "Homogeneous Gas-Phase Decomposition of Hydrogen Peroxide," 8th Symposium (International) on Combustion, 110 (1960).

Bamford, C. H., "The Reaction between Nitric Oxide and some Nitrogenous Free Radicals," *Trans.* Faraday Soc., **35**, 568 (1939).

Bradley, J. N., R. N. Butlin, and Lewis, "Oxidation of Ammonia in Shock Waves," *Trans.* Faraday Soc., **64**, 61 (1968).

Baulch, M., B. Drysdale, D. G. Horne, and A. C. Lloyd, *Evaluated Kinetic Data for High Temperature Reactions*, London, Butterworths.

Bull,I D. C., "A Shock Tube Study of the Oxidation of Ammonia," *Comb. and Flame*, **12**, 603 (1968).

Cox, R. A. and R. G. Derwent, "Kinetics of the Reaction of HO_2 with Nitric Oxide and Nitrogen Dioxide," *J. Photochemistry*, **4**, 139 (1975).

Cullis, C. F. and E. J. Newitt, "The Production of Hydrogen Peroxide via Combustion: Conditions for the Synthesis and Isolation in High Yield from the Combustion of Aliphatic Alcohols," 6th Symposium (International) on Combustion, 827 (1958).

Gehring, M., K. Hoyermann, H. Schacke, and J. Wolfrum, "Direct Studies of Some Elementary Steps for the Formation and Destruction of Nitric Oxide in the H—N—O System," 14th Symposium (International) on Combustion, 99 (1974).

Hussain, D. and R. G. W. Narrish, "The Explosive Oxidation of Ammonia and Hydrazine Studied by Kinetic Spectroscopy," Proc. Roy. Soc., **A273**, 145 (1963).

Lyon, R. K., "Method for Reducing of the Concentration of NO in Combustion Effluents Using Ammonia," US Patent 3,900,554 (1975).

Maclean, D. I. and H. Gg. Wagner, "The Structure of the Reaction Zones of Ammonia-Oxygen and Hydrazine-Decomposition Flames," 11th Symposium (International) on Combustion, 871 (1967).

Meyer, E., H. A. Olschewski, J. Troe, and H. Gg. Wagner, "Investigation of N_2H_4 and H_2O_2 Decomposition in Low and High Pressure Shock Waves," 12th Symposium (International) on Combustion, 345 (1968).

Manuscript received January 4, 1980; revision received January 22, and accepted February 10, 1981.

Solution of Stiff Ordinary Differential Equations by Decomposition and Orthogonal Collocation

A fast and accurate method was developed for the integration of large sparse systems of stiff initial value ordinary differential equations. The system is ordered, decoupled and, if necessary, torn into subsystems (also called blocks) which are then solved by orthogonal collocation on finite elements. The size of these elements, or steps, is different for each subsystem and is a function of the stiffness of the set of equations constituting the subsystem. The steps are overlapped for maximum computational efficiency.

M. K. BURKA

Department of Chemical and Nuclear Engineering
University of Maryland
College Park, MD 20742

SCOPE

Mathematical modeling of physical systems encountered in kinetics, process design, control, etc., often involve analysis of large sets of stiff initial value ordinary differential equations (ODE's). Such problems may occur, for example, in the kinetics of large multicomponent reactions (Farrow and Edelson, 1974) where some of the reaction rate constants are substantially larger than others, in the analysis of transient phase change distributions in the separations processes, or in circuit theory (Gear, 1971; Hachtel, 1971).

Over the past few years, many special solution techniques have been developed to handle stiff systems (Cash, 1976; Michelsen, 1976; Bui and Bui, 1979; Bui, 1979) but most of these are effective only for small problems, those consisting of approximately ten equations or less. The methods published for large systems (Gear, 1971; Hachtel, 1971; Hofer, 1976) have invariably involved some form of decomposition.

The decomposition procedure used in this method consists of selection of an output variable for each equation in the system, ordering the whole set of equations (putting them in the sequence in which they have to be solved), partitioning the equations into single equations plus the smallest cycles obtainable without tearing, and finally tearing some of these cycles by removing a variable (or variables) in a systematic manner to yield even smaller blocks and/or single equations.

The decomposed system is solved by orthogonal collocation on finite elements. Orthogonal collocation solutions are approximations in the form of the sums of orthogonal polynomials. The calculations involve substituting the sums for the unknown variables and evaluating the individual polynomials' coefficients by setting the residuals equal to zero at the collocation points. A series of ODE's is thus reduced to a series of algebraic equations.

The results obtained in this manner are valid only over a limited range of the independent variable, requiring the full region of interest to be divided into sections (also called finite elements, steps, or subintervals), and separate solutions are then calculated for all dependent variables for each section. A modified full step/half step method is used to determine the optimal size of the subintervals used for each block.

0001-1541-82-5051-0011-$2.00

CONCLUSIONS AND SIGNIFICANCE

For the first step in the decomposition sequence the variables in the full set of equations are weighted to encourage the selection of linear output variables. The ordering and partitioning algorithms of Steward (1965), Christensen and Rudd (1969), Upadhye and Grens (1972, 1975), or any of the many others that are applicable may then be applied. If cyclic subsystems are found, these have to be further analysed.

It was found that loops of five or more equations should always be torn while equations constituting smaller loops should be solved simultaneously. Within each loop, the tearing algorithm first assigns a set of weights (different from those used before) such that stiff variables (variables which change much more rapidly as a function of time, than the remaining variables) are weighted very heavily, lighter weights are assigned to nonlinear nonstiff variables and the smallest weights to linear nonstiff variables. Selection of the torn set is based on the minimization of the sum of all of these weights for the whole system.

The series of initial value ODE's were then solved by orthogonal collocation using Legendre polynomials and third order approximations were found to be most efficient. Orthogonal collocation in this context transforms a set of n ODE's into 3n algebraic equations. The zeros of the shifted Legendre polynomials were used as collocation points.

The individual characteristics of each equation or block at any value of the independent variable determine the associated subinterval sizes used; for example small steps are used for blocks containing stiff elements. A block or subsystem that contains the derivative of one or more stiff variables will be called a stiff subsystem. All step sizes vary over the full interval of interest and are determined using a full step/half step method which includes a new technique of overlapping the subintervals of contiguous steps. In the full step/half step sequence, the half steps are overlapped such that the initial values of the dependent variables for each new half step are set equal to their values calculated at the last collocation point of the previous half step. These are used as final results while solutions obtained using the full step are used only for comparison purposes, i.e., to test for accuracy.

The final set of algebraic equations are solved by the Newton-Raphson method, if non-linear or, by direct inversion or LU factorization, if linear. Tearing a nonlinear loop in this sense results in an effective combination of the Newton-Raphson method and direct substitution (which by itself was found to be unstable for all problems tested).

The overall method was found to be simple and efficient and gave results to any desired accuracy. As with all numerical methods, increased accuracy could be obtained with additional expenditures in computer time. For those familiar with orthogonal collocation, different trial functions may be substituted in order to solve other problems such as two-point boundary value problems.

INTRODUCTION

The differential equations

$$\frac{d\underline{y}(x)}{dx} = \underline{f}[\underline{y}(x),x] \tag{1}$$

are stiff when the eigenvalues λ of the Jacobian

$$\underline{J} = \frac{\partial \underline{f}}{\partial \underline{y}} \tag{2}$$

are widely separated, i.e., some of the variables change much more rapidly than others, often only over a small subdomain of x. The solution techniques developed by Michelsen (1976), Cash (1975), and Bui and Bui (1979), for example, cannot be used for the solution of large systems because the calculations take too long and the consequent cost of computer time is prohibitive. The purpose of this work was therefore to develop an efficient method for the solution of large sets of stiff ODE's, a technique that would take advantage of the sparsity of the system. The original concept was to decompose the system so that the variables with the fast transients are decoupled from the others and the resultant series of subsystems would then be solved by appropriate numerical methods.

Decomposition algorithms analyze the structure of the equations and rearrange them to permit sequential, instead of the more time consuming simultaneous solution. This is easy to implement for algebraic systems because constant values are sought for the unknown variables. For ODE's the dependent variables are unknown over some range $a \leq x \leq b$ of the independent variable. The decomposition scheme would therefore be a function of not only the overall structure of the system, but also of the method of integration used (which for optimal efficiency should be allowed to vary over the range $a \leq x \leq b$). Numerical step methods used to solve initial value ODE's calculate an approximate solution $\bar{\underline{y}}(x_{i+1})$ (the exact solution will be denoted as $\underline{y}(x_{i+1})$) by taking steps of length h where

$$x_{i+1} = x_i + h \tag{3}$$

Depending on the specific method used, $\bar{\underline{y}}(x_{i+1})$ may be a function of $\underline{f}[\bar{\underline{y}}(x_i), x_i]$, $\bar{\underline{y}}_i$, x_i, $\bar{\underline{y}}_{i-1}$, x_{i+1}, $\bar{\underline{y}}_{i+1}$, x_{i+1}, etc. The results obtained by these methods are affected by the two areas of difficulty for stiff problems: stability and/or accuracy. Accuracy is a measure of how well the calculated solution $\bar{\underline{y}}(x_{i+1})$ approximates the true solution $\underline{y}(x_{i+1})$ over the interval $a \leq x \leq b$. Stability refers to whether or not the error at step i

$$\underline{E}_i = \bar{\underline{y}}(x_i) - \underline{y}(x_i) \tag{4}$$

is bounded for large i.

Michelsen (1976) and Cash (1975) developed such step by step methods for stiff ODE's which are effective in terms of the accuracy/stability dilemma, but have a serious drawback for use on decomposed systems in that they calculate $\bar{\underline{y}}$ at only discrete points x_i. Because the step sizes used reflect the local stiffness of a system, subsystems containing stiff variables should be integrated using small steps, while the rest are solved for using large steps. The problem with this approach is that values of some variables in non-stiff subsystems, e.g., $\bar{\underline{y}}_A$, are usually needed in the integration of some stiff subsystems, e.g., $\bar{\underline{y}}_B$, requiring interpolation between the calculated values at $\bar{\underline{y}}_A(x_i)$ and $\bar{\underline{y}}_A(x_{i+1})$, i.e., calculated values of $\bar{\underline{y}}_A$ at x_i and $\bar{\underline{y}}_A$ at x_{x+1} where the large step for $x_{j+1} = x_j + k$ uses $k > h$. This is time consuming and partially defeats the purpose of the decomposition. The use of functional approximations incorporating the large step/small step procedure suggested above is more logical.

Studies using orthogonal collocation had shown it to be a promising technique for the solution of small stiff systems (Villadsen, 1970). Using approximations in a form

$$\bar{\underline{y}}(x) = \underline{y}(x_o) + \sum_{i=1}^{n} \underline{a}_i P_i \tag{5}$$

where the P_i were a series of orthogonal polynomials, the N differential Eq. 1 were transformed into $N \times n$ algebraic equations in the unknowns a_i. Orthogonal collocation is an implicit A-stable method, i.e., the approximation Eq. 5 is valid over a subdomain of x of length σ, and there are no preset limits on the size of the product $|\sigma\lambda|$ for $\mathrm{Re}(\lambda) < 0$. A-stable methods thus have no stability problems but may be inaccurate; the variables with the fast transients (large λ's) may be poorly approximated (Seinfeld, Lapidus, and Hwang, 1970; Lapidus and Seinfeld, 1971, p. 140; Bui, 1979).

The technique used to solve for a_i in Eq. 5 is a function of whether or not there are any nonlinearities in the system. If all equations are linear, any Gauss-Jordan method or LU factorization can be used to invert the whole matrix. If there is even one nonlinearity, the whole system has to be solved by some iterative technique, and in this case the Newton-Raphson method was the only one found to converge for stiff systems. Therefore orthogonal collocation is impractical for large systems ($N > 10$) if they are to be solved simultaneously because the repeated inversions of $N \times n$ matrices for linear problems and even larger matrices for nonlinear problems are too costly. Combination of deposition with orthogonal collocation is proposed because orthogonal collocation provides the functional approximations that would permit implementation of the decomposition scheme outlined above.

Numerical experiments were run on the following three test examples:

I. A set of two simultaneous equations, one linear one nonlinear. This is the mathematical model of three irreversible reactions involving four chemical species.
II. A set of four simultaneous linear equations representing the transient mass balances around a mixer, reactor and separator.
III. A set of thirty two simultaneous ODE's (nine nonlinear, twenty-three linear) which represent a modified version of a flowsheet presented by Christensen and Rudd (1969).

An extended description of these three test problems are given in the Appendix.

The objectives of the simulations were to determine:

1. The best approximating polynomial to use for initial value problems.
2. The optimal n; i.e., order of approximation.
3. Which collocation points to use for maximum efficiency without loss of accuracy.
4. How to splice together contiguous approximate solutions.
5. The effect of stiffness on the solution technique.
6. How the method compares with others presented in the literature for stiff problems.
7. The best approach for large problems, specifically: (a) benefits of tearing as a function of loop size; and (b) the effectiveness of varying subinterval size for each subset.
8. How the combination would affect each of its parts; i.e., the methods of decomposition and orthogonal collocation as previously developed by others.

Overall, the goal was to test the efficiency, accuracy and general applicability of the method.

ORTHOGONAL COLLOCATION

Collocation methods give accurate and efficient solutions to differential equations (Finlayson, 1972; Villadsen, 1970). For the series of initial value ODE's of interest here (Eq. 1), Finlayson and Villadsen recommend solutions of the form

$$\bar{\underline{y}}(x) = \underline{y}(x_o) + x \sum_{i=1}^{n} \underline{c}_i P_{i-1}(x) \tag{6}$$

where

x_o = initial value of the independent variable x
P_i = Legendre polynomials
$\underline{c}_i$ = parameters, to be determined

The solution is embodied in the coefficients $\underline{c}_i$ obtained by setting the residual defined as

$$\underline{R}(x) = \frac{d\bar{\underline{y}}(x)}{dx} - \underline{f}\,[\bar{\underline{y}}(x), x] \tag{7}$$

equal to zero at the "collocation points," which are the zeros of the nth order Legendre polynomial.

Successful application of collocation methods to specific problems comes with experience. The general nature of initial value problems is that, although $a = x_o$ is some finite number, the end of the region of interest, b, may be a finite number or ∞. For example, in control, b might correspond to the time at which a system arrives at the desired new steady state or equilibrium. This, at first glance, indicated the use of orthogonal polynomials, valid over the range 0 to ∞, specifically the most general Laguerre polynomials. When tried on test examples I and II extremely inaccurate results were obtained for the following reasons.

Orthogonal collocation is a curve fitting technique and as such the accuracy of the approximation is a function of the behavior, i.e., the shape of the curves of $\underline{y}$ vs. x. These approximations are incorrect unless every inflection is accounted for by sampling in its vicinity. Stiff systems are characterized by irregularly shaped curves particularly in stiff regions where there may be numerous peaks and valleys. The many sampling points required in these regions are not provided by the zeros of the Laguerre polynomials.

In fact, no approximating function of this form can be used over the entire region $a \le x \le b$ for stiff systems if $[a, b]$ is large and encompasses both stiff and non-stiff areas because it does not provide the cluster of sample points needed in the stiff section. Instead, it is necessary to divide $[a, b]$ into two or more subintervals and thus splice a series of solutions together combining the attributes of orthogonal collocation and finite differencing methods. The procedure, called orthogonal collocation on finite elements, provides solutions to any desired accuracy.

The Legendre polynomials

$$P_n(x) = \frac{1}{2^n\, n!} \frac{d^n}{dx^n} (x^2 - 1)^n \tag{8}$$

are orthogonal to each other over the interval $-1 \le x \le +1$,

$$\int_{-1}^{+1} P_k(x) P_l(x) dx = 0 \; k \neq l \tag{9}$$

[the weighting function $w(x) = 1$]. The collocation points are the n roots of P_n shifted to the interval [0, 1]. For initial value problems, Finlayson and Villadsen recommend using an altered form of Eq. 6:

$$\bar{\underline{y}}(X) = \underline{y}(0) + X \sum_{i=1}^{n+1} \underline{d}_i P_{i-1}(X) \tag{10}$$

where

$$X = \frac{x - a}{b - a} \quad 0 \le X \le 1.0$$

and the collocation points are the shifted zeros of the Legendre polynomial plus $X = 1$. The $\underline{d}_i$'s are calculated by solving the series of simultaneous algebraic equations formed when the residuals are set equal to zero at these collocation points

$$\underline{R}(X_m) = \frac{d\bar{\underline{y}}(X_m)}{dX} - f[\bar{\underline{y}}(X_m), X_m] = 0 \tag{11}$$

The actual solution procedure is easy to implement. For example, the solution of the linear equation

$$\frac{dy}{dx} = y \tag{12}$$

using a third order approximation is

$$y = y_o + d_1 X + d_2 X^2 + d_3 \left(\frac{3X^3 - X}{2}\right). \tag{13}$$

Therefore,

$$\frac{dy}{dX} = d_1 + 2d_2 X + d_3\left(\frac{9X^2 - 1}{2}\right) \tag{14}$$

giving three simultaneous equations

$$d_1(1 - X_m) + d_2(2X_m - X_m{}^2) + d_3\left[\left(\frac{9X_m{}^2 - 1}{2}\right) - \left(\frac{3X_m{}^2 - X_m}{2}\right)\right] = y_o; \qquad m = 1, 2, \text{and } 3 \tag{15}$$

which can be solved for the d_i's by direct matrix inversion or LU factorization. The solution of the nonlinear equation

$$\frac{dy}{dX} = y^2 \tag{16}$$

gives three simultaneous equations:

$$d_1 + 2d_2X_m + d_3\left(\frac{9X_m{}^2 - 1}{2}\right) - \left[y_o + d_1X_m + d_2X_m{}^2 + d_3\left(\frac{3X_m{}^3 - X_m}{2}\right)\right]^2 = 0; \qquad m = 1, 2, \text{and } 3 \tag{17}$$

which require an iterative numerical solution. The latter problem is aggravated by the fact that the d_i's have no physical significance and most assumed initial values are therefore inherently poor guesses. Successive substitutions and the Reguli-Falsi method diverged for every example tested and, therefore, the more time consuming Newton-Raphson method had to be used.

Optimal Collocation Order

The approximating equation, Eq. 10, can be used with any desired value of n. With higher order approximations larger subintervals can be used but the size of the final matrix to be inverted to solve for the d_i's also increases. Tests were run on Examples II and III (Appendix) to determine which, if any, value of n is better than the others. The results showed that the optimal solutions were obtained when third order approximations were used.

Examples II and III were solved using n values ranging from 2 through 6. It was found that the amount of computer times used per subinterval in relative proportions was:

n	2	3	4	5	6
Computer Time Used Relative to $n = 2$	1	2	4	8	10

At the same time the solution of Example III showed that at least twice as many subintervals were necessary for $n = 2$ as for $n = 3$ and approximately the same number for $n = 4$ as for $n = 3$:

n	2	3	4	5	6
Total Steps	9	4	4	No Results*	No Results*

Complete results were obtained for the small problem, Example II, with all values of n, and gave the following relative execution times:

n	2	3	4	5	6
Execution Time	1.28	1	1.88	2.15	2.65

Tests thus showed that third order approximations give the best results for these stiff systems. Further work needs to be done, using a variety of test problems, to fully verify these results.

Overlapping Subintervals

One aspect of this problem is finding an optimal splicing technique. As outlined above, orthogonal collocation converts a differential equation into a series of simultaneous equations. If the original ODE's are linear, direct matrix inversion or LU factorization is used to solve for the $\underline{d}_i$'s. For sets of nonlinear ODE's the Newton-Raphson method is used necessitating repeated matrix manipulations. In either case the method depends on some form of matrix manipulation which consumes the largest portion of the solution time. Therefore, the most efficient solution is inherently one that minimizes the size of these matrices.

Elimination of the point $X = 1$ was recommended by Villadsen for stiff problems because it adds little to the overall approximating solution $\underline{y}$. This was tried but gave poor results because the value of $\underline{y}$ at $X = 1$, which is an extrapolated number, is used as the initial value for the next subinterval and the error is thus ultimately greatly magnified. This problem is eliminated if the subintervals are overlapped, such that the initial value for each subinterval is equated to the last zero of the Legendre polynomial for the previous subinterval. For example, if a third order approximation is to be used, for each subinterval j, $\bar{\underline{y}}^{(j)}$ is of the form

$$\bar{\underline{y}}^{(j)} = \underline{y}^{(j)}(0) + \underline{d}_1XP_o + \underline{d}_2XP_1 + \underline{d}_3XP_2 = \underline{y}^{(j)}(0) + \underline{d}_1X + \underline{d}_2X^2 + \frac{\underline{d}_3}{2}(3X^3 - X) \tag{18}$$

The residuals are set equal to zero at

$$X_1 = 0.112702$$

$$X_2 = 0.5$$

$$X_3 = 0.887298$$

The set of simultaneous equations are solved for the $\underline{d}_i$'s which are in turn used to evaluate $\underline{y}^{(j)}$ at $X_3 = 0.887298$. This value is then used in the following subinterval for $\underline{y}^{(j+1)}(0)$, i.e.,

$$\underline{y}^{(j+1)}(0) = \bar{\underline{y}}^{(j)}(0.887298)$$

Tests of accuracy are made by using full step/half step comparisons. This is combined with overlapping as follows. An initial subinterval of size $\sigma^{(1)}$

$$\sigma^{(1)} = x^{(1)} - x_{\text{initial}} \tag{19}$$

is chosen based on any available *a priori* knowledge of the system. A better criterion could be established, such as limits on $|\sigma\lambda|$ as a function of n, the order of the approximation, where λ is the largest eigenvalue. This approach was rejected because calculation of the eigenvalues for the large systems, expecially nonlinear ones, is much more time consuming than the calculations involved in the trial and subsequent modification according to Eq. 22 of any "guessed" value of $\sigma^{(1)}$. The series of equations are solved yielding a set of $\underline{d}_i$'s and thus estimated functions $\bar{\underline{y}}^{(1)}(X)$. Because

$$X = \frac{x - x_{\text{initial}}}{x^{(1)} - x_{\text{initial}}}$$

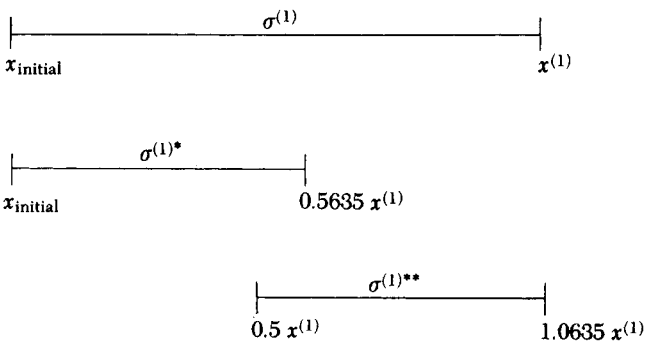
then

$$x = X(x^{(1)} - x_{\text{initial}}) + x_{\text{initial}} = X\sigma^{(1)} + x_{\text{initial}}$$

$\bar{\underline{y}}^{(1)}(x)$ can be easily found from $\bar{\underline{y}}^{(1)}(X)$. Assuming, a third order solution is used, a second set of approximations using two $\sigma^{(1)*}$s each

$$\sigma^{(1)*} = 0.5635\ \sigma^{(1)} = \sigma^{(1)**}$$

is used overlapped as outlined above such that:



* No complete results were obtained because the prescribed time limit was exceeded before completion of the problem. For $n = 5$ this occurred after two steps and for $n = 6$ after just one step.

giving another set of estimated functions $\bar{y}^{(1)*}(x)$ and $\bar{y}^{(1)**}(x)$. The relative errors $\underline{\varepsilon}^{(1)}$ (consisting of $\underline{\varepsilon}^{(1)*}$ and $\underline{\varepsilon}^{(1)**}$):

$$\underline{\varepsilon}^{(1)*}(x) = \bar{\underline{y}}^{(1)*}(x) - \bar{\underline{y}}^{(1)}(x) \quad x_{\text{initial}} \le x \le 0.5x^{(1)} \tag{20}$$

$$\underline{\varepsilon}^{(1)**}(x) = \bar{\underline{y}}^{(1)**}(x) - \bar{\underline{y}}^{(1)}(x) \quad 0.5\,x^{(1)} \le x \le x^{(1)} \tag{21}$$

are both functions of x. The maximum value of the error can be found by either spanning $[x_{\text{initial}}, x^{(1)}]$ at set intervals such as $0.1\sigma^{(1)}$ or by examining $\underline{\varepsilon}^{(1)}$ at values of x found by solving

$$\frac{\partial \underline{\varepsilon}^{(1)}}{\partial x} = 0,$$

at the boundaries $x = x_{\text{initial}}$ and $x^{(1)}$, and at the discontinuity $x = 0.5x^{(1)}$. The desired accuracy determines which of the two approaches to use. For the examples tested, the former method proved adequate.

If $\underline{\varepsilon}^{(1)}$ is greater than 0.01 (chosen as an "acceptable" error bound for most engineering calculations) within the range $x_{\text{initial}} \le x \le x^{(1)}$, the results are rejected. The procedure is repeated using the step size σ^* as the new full step.

If $\underline{\varepsilon}^{(1)}$ is less than 0.01 everywhere over the full range $x_{\text{initial}} \le x \le x^{(1)}$ the results are accepted. The numerical solutions obtained with the two smaller, overlapped steps $\sigma^{(1)*}$ and $\sigma^{(1)**}$ are recorded. The next step size $\sigma^{(2)}$ is determined based on the maximum of $\underline{\varepsilon}^{(1)}$ such that

$$\sigma^{(2)} = \sigma^{(1)} \cdot \text{Min}\,[(4q)^{-0.25}, 3]$$

where

$$q = \text{Max}\left|\frac{\underline{\varepsilon}^{(1)}}{0.01}\right| \tag{22}$$

and

$$\underline{y}^{(2)}(0) = \bar{\underline{y}}^{(1)}(1.0) = \bar{\underline{y}}^{(1)**}(0.887298). \tag{23}$$

This is the acceleration scheme used by Michelsen (1976).

Results

The two small test examples, Examples I and II (Appendix) were solved using both this proposed method and Michelsen's (1976) method. The total execution times (in seconds) are:

	Proposed Method	Michelsen's Method
Example I	3.12	18.91
Example II	8.61	22.52

Example I consists of two nonlinear ODE's while Example II consists of four linear ODE's.

The results showed that orthogonal collocation on finite overlapped elements using third order approximations is an excellent method for solving stiff initial value ODE's. Furthermore, the previous table shows that its effectiveness increases as the size of the system decreases.

DECOMPOSITION

The analysis of chemical processes often requires simultaneous solution of large systems of ODE's. For sparse systems, obtained for example when dealing with transient material and energy balances, these calculations can be simplified by reorganization so that sequential solution is possible. Such decomposition may consist of partitioning and ordering of the equations into cyclic subsystems which may be further reduced by tearing. The partitioning and tearing algorithms in the literature, developed for systems of algebraic equations, were modified to handle systems of ODE's.

Selection of Output Variables

Orthogonal collocation changes a set of simultaneous ODE's in the dependent variables $\underline{y}$ (Eq. 1) into a larger set of simultaneous algebraic equations in the variables $\underline{d}_i$ (Eq. 10). While the time to solve a single algebraic equation is affected by the structure of the equation, this effect is accentuated for an ODE. For example, solving

$$x + y^2 = 0 \tag{24}$$

for y (i.e., y is the "output" variable) is more time consuming than solving it for x. The solution of Eqs. 12 through 17 show that it is substantially more time consuming and risky (the iterative technique may diverge) to solve for the nonlinear variable in ODE's. Several schemes have been used for selecting the output variables for algebraic systems (Soylemez and Seider, 1972; Lee, Christensen, and Rudd, 1966); however, the only factor of importance for dynamic problems is whether or not the resultant system is linear. It was found that linear variables should be assigned a weight of one and nonlinear variables a weight of five because on the average five iterations were necessary for convergence.

Partitioning and Ordering

Partitioning involves subdividing the system into smaller subsystems that may be solved simultaneously. The ordering is setting up the sequence in which these subsystems are to be solved. For example, elimination of stream 5 in Example II results in the information in Figure 1 flow diagram with corresponding equations

$$\underline{v}_2 = f_1(\underline{v}_1)$$

$$\underline{v}_3 = f_2(\underline{v}_2, \underline{v}_6)$$

$$\underline{v}_6 = f_3(\underline{v}_3)$$

$$\underline{v}_4 = f_4(\underline{v}_3)$$

where $\underline{v}_i$ are the variables in stream i. This can be partitioned into three subsystems: (1) f_1, (2) f_2 and f_3 (which have to be solved simultaneously), (3) f_4. The order of the solution will proceed by the steps

(1) solve for the variables $\underline{v}_2$ using f_1
(2) solve for the variables $\underline{v}_3$ and $\underline{v}_6$ using f_2 and f_3
(3) solve for the variables $\underline{v}_4$ using f_4.

Numerous methods have been devised for partitioning and ordering large algebraic systems (Lee, Christensen, and Rudd, 1966; Steward, 1965; Christensen and Rudd, 1969) any of which can be applied without modification to dynamic problems.

Tearing

The partitioning algorithms yield a sequence of blocks which cannot be further reduced without tearing. Algebraic systems are torn by assuming values for the torn variables and updating these values by solving the resulting sequence of equations iteratively to some prescribed accuracy. Tearing dynamic systems, for which the integration technique is orthogonal collocation as outlined above, involves similar iteration, in this case starting with some assumed values for the coefficients d_{ji} of the torn variable y_j. In the example above, see Figure 1, f_2 and f_3 form a cycle which can be opened by tearing either at stream 3 or stream 6 and sequentially

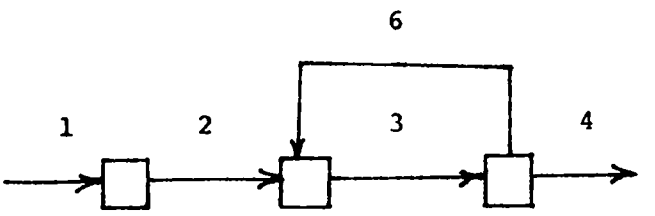
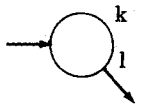


Figure 1. Elimination of stream 5 in example II.

solving the nontorn equations, then the other, repeatedly until y_3 and y_6 converge.

Various algorithms aimed at finding the optimum torn stream or streams as well as the least time consuming method for finding them, have been developed (Batstone, Fenton, and Prince, 1972; Upadhye and Grens, 1972; Genna and Motard 1975). Generally weights were attached to streams or variables and the sum of all weights were minimized. Any of these methods can be adopted for dynamic problems as long as the objective is minimizing the total number of variables torn.

The dynamic programming technique developed by Upadhye and Grens (1972) was used with good results for the examples tested. The states of the system obtained when one or more of the streams had been torn are depicted as nodes and the streams torn in going from one state to the next are represented by arcs joining the nodes. For each node there are two parameters, k and l where



k = sum of weight w_i of variables torn to reach the node
l = streams torn to reach the node

Once a cycle is torn no other stream in that cycle is considered for tearing because the cycle is eliminated. For the example of Figure 4, using the following weightings, for example,

Stream	w
2	8
3	7
5	3
6	3

we could arrive at Figure 2.

The crucial point is the weights to be used. Of highest priority is that the minimum number of variables be torn followed by the premise that with other things equal, given the choice between a stiff and a slowly changing variable, always tear the latter. The next lower priority is the clustering of stiff variables in as few subsystems possible. As an illustration, take the problem depicted in Figure 3. If variables 1 and 5 are stiff, the possible tear variables and resultant solution sequence are as shown in Table 1. It would be more efficient to tear either 3, 4 or 7 than either 2 to 6 since this will group the stiff variables 5 and 1 together, Table 1.

There is one major difference between tearing dynamic vs. algebraic systems: it is not always clearcut whether a loop consisting of dynamic equations should be torn while the goal is to tear all cycles in an algebraic problem. The determining factors are the size of the dynamic loop and its linearity. The results can be understood by realizing that all calculations are based on matrix inversion which involve approximately $n^3/3$ multiplications for an n by n matrix.

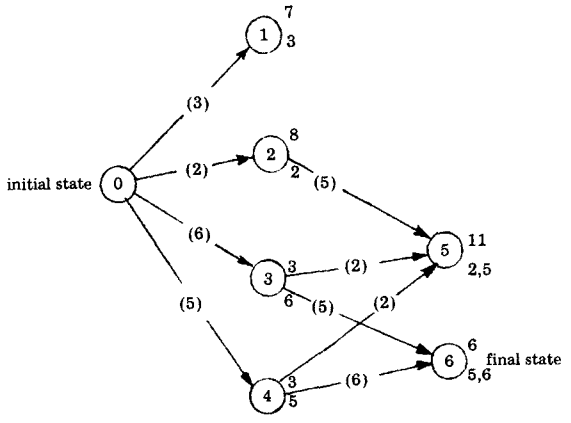


Figure 2. Dynamic programing solution for example 2.

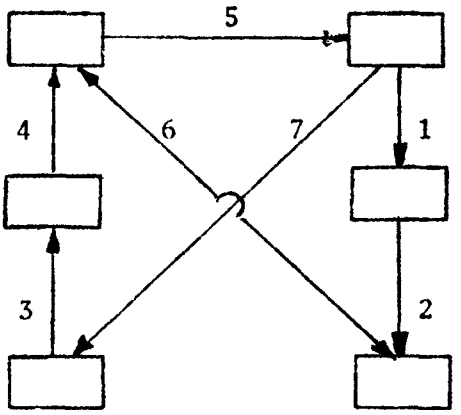


Figure 3. Clustering of stiff variables.

For a loop consisting of five linear equations, simultaneous solution (using $n = 3$) requires the inversion of a 15 by 15 matrix or, approximately $(15^3)/3 = 1125$ multiplications. When this block is torn, five 3 by 3 matrices are solved repeatedly which is $5 \times (3^3)/3 = 45$ multiplications per iteration. For Example III this was examined, and depending on the particular value of k_2 used, two through eleven iterations were necessary or 90 to 495 multiplications (most averaged around five iterations or 225 multiplications). Even including all the other calculations necessary to solve blocks in series, the time to invert the large 15 by 15 matrix could not be compensated for.

If the block contains even one nonlinearity, this effect multiplies since then the 15 by 15 matrix has to be repeatedly inverted for simultaneous solution. As the loop gets bigger, the effect is also more pronounced, a loop of 6 ODE's means inversions of an 18 by 18 matrix, etc. Therefore, loops of five or more ODE's should always be torn.

For smaller cycles, the decision is not that obvious. Even though the actual number of multiplications is less for series solution of four equations, both linear or nonlinear, the other calculations involved in the tearing plus the risk of divergence indicate that simultaneous solution is to be preferred.

Numerical experiments were performed on Examples II and III to determine the validity of the above theory. When stream 3 (which contains the stiff variables) is torn in Example II, the system

TABLE 1. POSSIBLE TEAR VARIABLES AND SOLUTIONS

Tear Variable	Solution Sequence
2	6
	3,4,5,7 simultaneously
	1
6	3,4,5,7 simultaneously
	1
	2
3	4
	5,1,2,6 simultaneously
	7
4	5,1,2,6 simultaneously
	7
	3
7	3
	4
	5,1,2,6 simultaneously

becomes acyclic. It took three times as long to solve the torn set as solving the system of four equations simultaneously (8.74 seconds vs. 24.52 seconds). On the other hand, Example III gave evidence that cyclic subsystems of five or more equations should be torn. LOOP 2, consisting of Eqs. III-18, III-24, III-26, III-30, and III-31 contains all linear elements, whereas LOOP 1, consisting of Eqs. III-9, III-10, III-13, III-14, and III-20 contains a nonlinear element. LOOP 3, is the overall large loop containing all the equations. To span $0 \leq t \leq 10$, the following step sizes were found to give the most efficient solution for the problem with all loops opened.

σ	t	Number of Steps
0.2	$0 \leq t \leq 0.6$	3
1.0	$0.6 \leq t \leq 2.6$	2
4.0	$2.6 \leq t \leq 10.6$	2
		Total = 7 Steps

and the total solution time was 96.21 s. When the problem was solved with LOOP 2 not torn, after 120 s, (the upper limit set on all problems) the solution had only progressed to $t = 6.6$. With LOOP 1 intact, the program only reached $t = 0.6$. Other step sizes were tried, which gave less efficient solutions for the totally acyclic system, in an effort to speed up the problem containing the cyclic elements but the results were identical to the above.

Varying Step Size According to Stiffness

The greatest strength of the method is that stiff subsystems can be removed and treated separately to allow faster solution of the rest of the system and to provide the most efficient results. Because step sizes are tried and tested for each subsystem independent of the others, little *a priori* knowledge of the problem is necessary.

Example III was run with k_2 values of 50 (stiff) and 1000 (very stiff) and results were compared for solutions where the step sizes for the whole system were identical (call this SOLUTION A) vs those solutions where the step sizes varied with the stiffness of each subsystem (call this SOLUTION B), where a subsystem may contain one or more simultaneous ODE.

For $k = 50$ SOLUTION B provided results up to $t = 1.95$ in the allotted 120 seconds. It was found that the initial step size for the stiffest subset was $\sigma = 0.025$. SOLUTION A, when tried with this step size diverged. The problem was caused by the coefficients of variables X_{27} and X_{30} which have to be solved for simultaneously. In the region $0 \leq t \leq 0.025$, X_3 stays almost constant whereas X_{27} changes markedly. Therefore, the coefficients, d_i, of X_{27} were orders of magnitude greater than those of X_{30}. This was a kind of stiffness in the coefficient matrices and prevented the small parameters from converging to a final solution. Smaller step sizes were not tested as they would inherently give a less efficient solution.

Results were obtained for both SOLUTION A and B for $k_2 = 1000$. The initial step size in this case was small enough so that the differences between the slopes of X_{27} and X_{32} were sufficiently reduced to eliminate the above complications. In the initial very stiff region ($0 \leq t \leq 0.23$), SOLUTION A took 2.84 times as long as SOLUTION B (68.61 seconds vs. 24.14 seconds). For $t > 0.23$ the differences between the two solutions were negligible.

Summary

Any ordering and partitioning method can be applied as long as initially weights are assigned to force the selection of linear output variables. The equations are solved in the sequence determined by the ordering algorithm. At the beginning of each set of calculations, the step-size to be used for the subsystems with the slow transients are assumed based on the previous step-size and step-acceleration (Eqs. 22 and 23). For the very first step, some initial value is chosen, as discussed following Eq. 19, then the iterations begin and the step sizes are adjusted automatically. Loops of five or more ODE's should always be opened by tearing the minimum number of variables. Optimal results are obtained when step size is allowed to float with the stiffness of each final subsystem as well as with each step σ, over the full range $a \leq x \leq b$ of the independent variable. This permits effective treatment of all stiff problems irrespective of where the stiffness occurs.

ACKNOWLEDGMENT

The author is greatly indebted to the late Professor L. Lapidus of Princeton University, who initiated this project, for his invaluable help during a large part of this work. Thanks also to Professors E. F. Johnson and M. Kostin also of Princeton University, for their helpful hints and suggestions.

NOTATION

$\underline{c}, \underline{d}$ = parameters in approximate solution
$\underline{f}[y(x), x]$ = set of ODE's
J = Jacobian $\dfrac{\partial f}{\partial y}$
n = order of the approximation
N = number of equations in system
P_i = orthogonal polynomials
$\underline{R}$ = residuals
X = independent variable
X_m = collocation points
$\underline{y}$ = exact solution of ODE's
$\underline{y}$ = approximate solution of ODE's

Greek Letters

$\underline{\epsilon}$ = relative error
$\underline{\lambda}$ = eigenvalues
σ = step size

Subscripts

o = initial value
i = any coefficient or polynomial
m = refers to mth collocation point

Superscripts

(j) = jth subinterval
$*, **$ = half steps

LITERATURE CITED

Batstone, D. B., G. Fenton, and R. G. H. Prince, "The Steady State Simulation of Chemical Plants of Arbitrary Configuration," *IFAC* Reprint (1972).

Bui, T. D., "Some A-Stable and L-Stable Methods for the Numerical Integration of Stiff Ordinary Differential Equations," *J. Assoc. Comput. Mach.*, **26**, 483 (1979).

Bui, T. D., and T. R. Bui, "Numerical Methods for Extremely Stiff Systems of Ordinary Differential Equations," *Appl. Math. Modelling*, **3**, 355 (1979).

Cash, J. R., "Semi-Implicit Runge-Kutta Procedures with Error Estimates for the Numerical Integration of Stiff Systems of Ordinary Differential Equations," *J. Assoc. Comput. Mach.*, **23**, 455 (1976).

Cash, J. R., "A Class of Implicit Runge-Kutta Methods for Numerical Integration of Stiff Ordinary Differential Equations," *J. Assoc. Comput. Mach.*, **22**, 504 (1975).

Christensen, J. H. and D. F. Rudd, "Structuring Design Computations," *AIChE J.*, **15**, 94 (1969).

Farrow, L. A. and D. Edelson, "The Steady-State Approximation: Fact or Fiction?" *Int. J. Chem. Kinetics*, **6**, 787 (1974).

Finlayson, B. A., *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York (1972).

Gear, C. W., "Simultaneous Numerical Solution of Differential Algebraic Equations," *IEEE Trans. on Circuit Theory*, **18**, 89 (1971).

Genna, P. L. and R. L. Motard, "Optimal Decomposition of Process Networks," *AIChE J.*, **21**, 656 (1975).
Hachtel, G. D., R. K. Brayton, and F. G. Gustavson, "The Sparse Tableau Approach to Network Analysis and Design," *IEEE Trans. on Circuit Theory*, **18**, 101 (1971).
Hofer, E., "A Partially Implicit Method for Large Stiff Systems of Ordinary Differential Equations with only Few Equations Introducing Small Time-Constants," *SIAM J. Numer. Anal.*, **13**, 645 (1976).
Lapidus, L. and J. H. Seinfeld, *Numerical Solution of Ordinary Differential Equations*, Academic Press, New York (1971).
Lee, W., J. H. Christensen, and D. F. Rudd, "Design Variable Selection to Simplify Process Calculations," *AIChE J.*, **12**, 1104 (1966).
Michelsen, M. L., "An Efficient General Purpose Method for the Integration of Stiff Ordinary Differential Equations," *AIChE J.*, **22**, 549 (1976).
Schneider, D. R., N. R. Amundson and R. Aris, "On a Mechanism for Autocatalysis," *Chem. Eng. Sci.*, **27**, 895 (1972).
Seinfeld, J. H., L. Lapidus, and M. Hwang, "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations," *I & EC Fundamentals*, **9**, 266 (1970).
Sena, M. and L. Kershenbaum, "Mechanism Discrimination in Free-Radical Reactions," Presented at 66th Annual Meeting of AIChE in Symp. "Recent Advances in Kinetics and Catalysis" (1970).
Soylemez, S. and W. D. Seider, "A New Technique for Precedence-Ordering Chemical Process Equation Sets," 71st National Meeting of the AIChE, Dallas, Texas (1972).
Steward, D. V., "Partitioning and Tearing Systems of Equations," *SIAM J. Numer. Anal.*, **B2**, 345 (1965).
Upadhye, R. S. and E. A. Grens II, "Selection of Decompositions for Chemical Process Simulation," *AIChE J.*, **21**, 136 (1975).
Upadhye, R. S. and E. A. Grens II, "An Efficient Algorithm for Optimum Decomposition of Recycle Systems," *AIChE J.*, **18**, 533 (1972).
Villadsen, J., *Selected Approximation Methods for Chemical Engineering Problems*, Reproset, Copenhagen (1970).

Manuscript received August 29, 1980; revision received January 21, and accepted February 10, 1981.

APPENDIX: Test Examples Used

I. The series of irreversible reactions

$$A \xrightarrow{k_1} B + C$$

$$A + B \xrightarrow{k_2} AB$$

$$AB \xrightarrow{k_3} 2B + C$$

has been studied by, among others, Sena and Kershenbaum (1970) and Schneider, Amundson, and Aris (1972). The corresponding ODE's are

$$\frac{d[A]}{dT} = -k_1[A] - k_2[A][B]$$

$$\frac{d[B]}{dT} = k_1[A] - k_2[A][B] + 2k_3[A]$$

$$\frac{d[C]}{dT} = k_1[A] + k_3[AB]$$

with initial conditions

$$[A] = [A]_o$$

$$[B]_o = [C]_o = [AB]_o = 0$$

the reaction rate constants k_1, k_2 and k_3 determine the stiffness of the system.

Schneider et al. used conservation of mass:

$$[A] - [B] + 2[C] = [A]_o$$

$$[A] + [AB] + [C] = [A]_o$$

to eliminate the equations for $[B]$ and $[AB]$. The two final dimensionless equations to be solved are then:

TABLE 2. EIGENVALUES FOR EXAMPLE 1 AS FUNCTIONS OF TIME AND ϵ.

ϵ	$t = 0.0$	$t = 1.0$	$t = 2.0$
0.02	0.3875	0.3727	0.1259
	−51.487	−37.603	−13.151
0.002	0.3987	0.3969	0.0944
	−501.5	−349.9	−40.4
0.0002	0.39999	0.3997	0.2377
	−5001.5	−3471.1	−142.8
0.00002	0.4	0.4	0.3189
	−50001.5	−34682.9	−727.0
0.000002	0.4	0.4	0.3891
	−500001.5	−346800.5	−6023.5
0.0000002	0.4	0.4	0.3988
	−5000001.5	−3467971.5	−58831.5

$$\epsilon \frac{dy}{dt} = -y(x + y) - \epsilon\lambda y \qquad \text{(I-1)}$$

$$\frac{dx}{dt} = 1.0 - x - 2(1.0 - \lambda)y \qquad \text{(I-2)}$$

where

$$x = 2\frac{[C]}{[A]_o} - 1.0 \quad y = \frac{[A]}{[A]_o}$$

$$\epsilon = \frac{k_3}{k_2[A]_o} \quad \lambda = \frac{k_1}{k_3}$$

$$t = k_3 T$$

$$x(o) = -1.0 \quad y(o) = +1.0$$

The eigenvalues of these nonlinear equations vary with time t and the parameters λ and ϵ. The latter (ϵ) is the measure of stiffness for this problem. Table 2 lists the two eigenvalues as functions of time and ϵ for $\lambda = 0.1$. Unless otherwise stated, the results cited in this paper are for $\lambda = 0.1$ and $\epsilon = 0.2$.

II. The flow diagram of a typical engineering process

Of a typical chemical engineering process, Figure 4, was used to generate a small system of ODE's as follows.

Initially all streams contain an inert substance and compound A in concentration $[A_o]$. At time $t = 0$, the reaction

$$A \underset{k_2}{\overset{k_1}{\rightleftarrows}} B \qquad \text{(II-1)}$$

is triggered in the reactor and the problem then consists of solving for $[A]$ and $[B]$ throughout the system as functions of time.

Both the mixer and the reactor were modeled as continuous stirred tank reactors. The separator is characterized by a separation ratio S which is a linear function of time

$$S = \frac{[A]_5}{[A]_4} = c_1 + c_2 t \qquad \text{(II-2)}$$

Constant flowrates are used, consistent with the relationships

$$q_2 = q_1 + q_5$$

$$q_3 = q_2 + q_6 \qquad \text{(II-3)}$$

$$q_3 = q_4 + q_5 + q_6$$

$$q_1 = q_4$$

$$S = 1.0 + 10.0t \quad 0 \le t \le 0.34$$

$$S = 4.4 \quad t \le 0.34$$

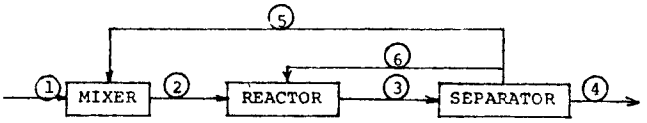


Figure 4. Typical chemical engineering process.

TABLE 3. EIGENVALUES FOR EXAMPLE 2 AS FUNCTIONS OF TIME AND ρ AND γ.

Time	λ's
$t = 0.0$	-7.648
	-5.0
	-3.452
	-1.654×10^{-7}
$t = 0.1$	-7.524
	-5.0
	-3.243
	-1.143×10^{-6}
$t \leq 0.5$	-7.423
	-5.0
	-3.048
	-1.486×10^{-6}

based on a number of systems, gave the final equations:

$$\frac{da_2}{dt} = 30 + \frac{a_3}{\rho} - 4a_2 \tag{II-13}$$

$$\frac{da_3}{dt} = 2a_2 + 0.1\,b_3 - \left(8 - \frac{1}{\rho}\right)a_3 \tag{II-14}$$

$$\frac{db_2}{dt} = -10.0 + 2b_3 + \frac{a_3}{\gamma} - 4b_2 \tag{II-15}$$

$$\frac{db_3}{dt} = -10.0 + 2b_2 + \left(5 + \frac{1}{\gamma}\right)a_3 - 1.1\,b_3 \tag{II-16}$$

where

$$\rho = \frac{0.5}{S} + 0.5$$

$$\gamma = 0.5 + 0.5S$$

The eigenvalues of this system are given in Table 3. They are functions of time because ρ and γ vary with time.

The final set of dimensionless equations for the system are:

$$V_M \frac{da_2}{dt} = q_1a_1 + q_5a_5 - q_2a_2 \tag{II-4}$$

$$V_M \frac{da_3}{dt} = q_2a_2 + q_6a_6 - q_3a_3 + V_Rk_2b_3 - V_Rk_1a_3 \tag{II-5}$$

$$S = a_5/a_4 = C_1 + C_2t \tag{II-6}$$

$$a_5 = a_6 \tag{II-7}$$

$$q_3a_3 = q_4a_4 + q_5a_5 + q_6a_6 \tag{II-8}$$

$$V_M \frac{db_2}{dt} = q_5b_5 - q_2b_2 \tag{II-9}$$

$$V_M \frac{db_3}{dt} = q_2b_2 + q_6b_6 - q_3b_3 + V_Rk_1a_3 - V_Rk_2b_3 \tag{II-10}$$

$$b_5 = b_6 \tag{II-11}$$

$$q_3b_3 = q_4b_4 + q_5b_5 + q_6b_6 \tag{II-12}$$

where

V_M = volume of the mixer
V_R = volume of the reactor
a_i = concentration of A in stream i
b_i = concentration of B in stream i

The following constant values

$V_M = 1$ $\quad V_R = 2$
$q_1 = 3$ $\quad q_2 = 4$ $\quad q_3 = 6$
$q_4 = 3$ $\quad q_5 = 1$ $\quad q_6 = 2$
$k_1 = 5.0$ $\quad k_2 = 0.1$
$[A_o]_i = 10.0$ $\quad [B_o]_i = 0.0$

were used for test purposes.

III. Information flow diagram

An information flow diagram that was decomposed by Christensen and Rudd (1969, p. 98) was adapted and stream numbers were added as shown in Figure 5. It was used to generate the following coupled ODE's

$$\frac{dx_1}{dt} = -2x_1 + 4x_{15}x_{22} + 4x_{27}x_{32} \tag{III-1}$$

$$\frac{dx_2}{dt} = -k_2x_2^2 + 2x_1 \tag{III-2}$$

$$\frac{dx_3}{dt} = -4x_3 + 0.5k_2x_2^2 - 2x_3 \tag{III-3}$$

$$\frac{dx_4}{dt} = -2x_4 + 4x_3 \tag{III-4}$$

$$\frac{dx_5}{dt} = -3x_5 + 2x_4 \tag{III-5}$$

$$\frac{dx_6}{dt} = -4x_6 + 3x_5 \tag{III-6}$$

$$\frac{dx_7}{dt} = -2x_7 + 4x_6 \tag{III-7}$$

$$\frac{dx_8}{dt} = -8x_8 + 2x_7 \tag{III-8}$$

$$\frac{dx_9}{dt} = -4x_9^2 + 3x_8 + 4x_{13} \tag{III-9}$$

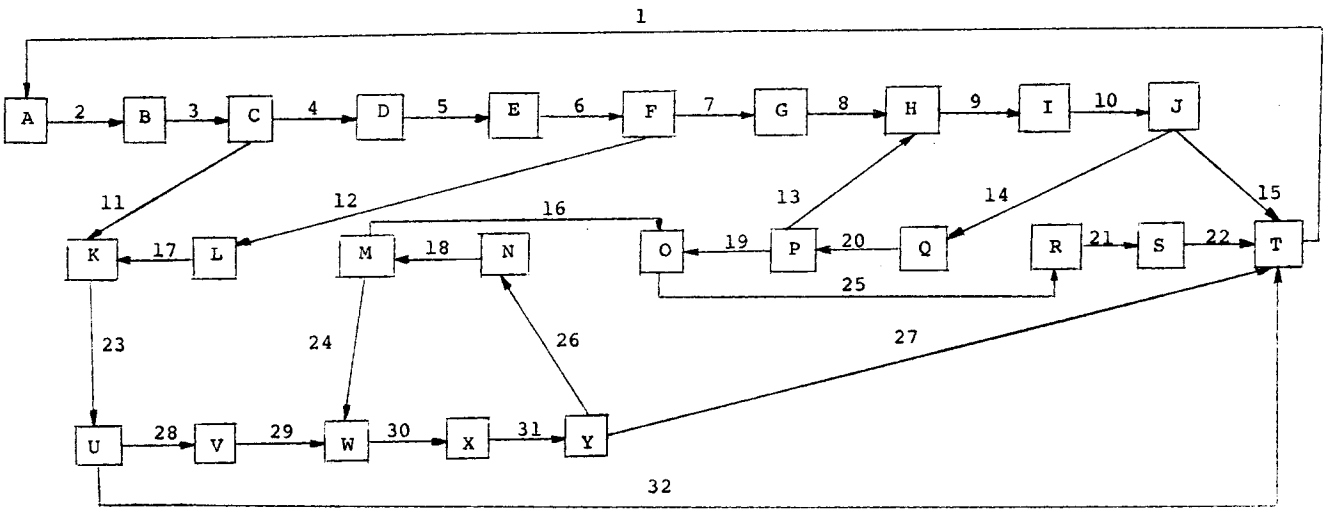


Figure 5. Information flow diagram of example III.

$$\frac{dx_{10}}{dt} = -2x_{10} + 2x_9^2 \quad \text{(III-10)}$$

$$\frac{dx_{11}}{dt} = -3x_{11} + 2x_3 \quad \text{(III-11)}$$

$$\frac{dx_{12}}{dt} = -2x_{12} + 4x_6 \quad \text{(III-12)}$$

$$\frac{dx_{13}}{dt} = -2x_{13} + 4x_6 \quad \text{(III-13)}$$

$$\frac{dx_{14}}{dt} = -3x_{14} + 2x_{10} \quad \text{(III-14)}$$

$$\frac{dx_{15}}{dt} = -4x_{15}x_{22} + 2x_{10} \quad \text{(III-15)}$$

$$\frac{dx_{16}}{dt} = -2k_{16} + 4x_{18} \quad \text{(III-16)}$$

$$\frac{dx_{17}}{dt} = -3x_{17} + 2x_{12} \quad \text{(III-17)}$$

$$\frac{dx_{18}}{dt} = -4x_{18} + 3x_{26} \quad \text{(III-18)}$$

$$\frac{dx_{19}}{dt} = -2x_{19} + 3x_{20} \quad \text{(III-19)}$$

$$\frac{dx_{20}}{dt} = -3x_{20} + 3x_{14} \quad \text{(III-20)}$$

$$\frac{dx_{21}}{dt} = -4x_{21} + 2x_{25} \quad \text{(III-21)}$$

$$\frac{dx_{22}}{dt} = -4x_{15}x_{22} + 4x_{21} \quad \text{(III-22)}$$

$$\frac{dx_{23}}{dt} = -3x_{23} - 4x_{23} + 3x_{11} + 3x_{17} \quad \text{(III-23)}$$

$$\frac{dx_{24}}{dt} = -4x_{24} + 4x_{18} \quad \text{(III-24)}$$

$$\frac{dx_{25}}{dt} = -2x_{25} + 2x_{16} + 2x_{19} \quad \text{(III-25)}$$

$$\frac{dx_{26}}{dt} = -3x_{26} - 2x_{31} \quad \text{(III-26)}$$

$$\frac{dx_{27}}{dt} = -4x_{27}x_{32} + 3x_{31} \quad \text{(III-27)}$$

$$\frac{dx_{28}}{dt} = -2x_{28} + 3x_{23} \quad \text{(III-28)}$$

$$\frac{dx_{29}}{dt} = -3x_{29} + 2x_{28} \quad \text{(III-29)}$$

$$\frac{dx_{30}}{dt} = -4x_{30} + 4x_{24} + 3x_{29} \quad \text{(III-30)}$$

$$\frac{dx_{31}}{dt} = -2x_{31} - 3x_{31} + 4x_{30} \quad \text{(III-31)}$$

$$\frac{dx_{32}}{dt} = -4x_{27}x_{32} + 4x_{23} \quad \text{(III-32)}$$

The system was first partitioned, giving two small loops and one large loop. LOOP 1 consisted of Eqs. III-9, III-10, III-13, III-14 and III-20 in which Eqs. III-9 and III-10 are nonlinear. LOOP 2 was made up of Eqs. III-18, III-24, III-26, III-30 and III-31 all of which are linear. Once these blocks were isolated they were collapsed into single nodes leaving one large cycle, LOOP 3, consisting of all of the remaining equations.

The constant k_2 was varied to make the system stiff. Solutions were obtained for $k_2 = 3$ (non-stiff) and for $k_2 = 50$ and $k_2 = 1000$ which made LOOP 3 stiff. Eigenvalues were not calculated because stiffness was obtained (and proven) in this manner for Example I.

Reduction of Thermodynamic Data by Means of the Multiresponse Maximum Likelihood Principle

The statistical behavior of some common parameter estimation methods is investigated by simulation. The methods based on the maximum likelihood principle give the best estimates of parameters in a given model. An analysis of the possibilities for detecting systematic errors by means of the maximum likelihood method showed that this method gives at best the same information as other well known tests.

S. KEMENY and J. MANCZINGER,
Department of Chemical Engineering, Technical University of Budapest, Budapest, Hungary

S. SKJOLD-JØRGENSEN,
Instituttet for Kemiteknik, Danmarks Tekniske Højskole, Lyngby, Denmark

K. TOTH
G. Richter Pharmaceutical Works, Budapest, Hungary

SCOPE

In recent years several methods have been proposed for the reduction of binary vapor-liquid equilibrium (VLE) information.

A VLE data point gives information either on the full set of variables (x,T,y,P) or on a subset of the variables (e.g., x,T,P). According to the Gibbs' phase rule there are two independent variables (e.g., x,T) in each binary VLE data point. Two vari-

Correspondence concerning this paper should be addressed to S. Kemény.
0001-1541-82-5024-0020-$2.00