New Approach to Approximation of Quantities Involving Physical Properties Derivatives in Equation-Oriented Process Design

A new method for approximating quantities that involve physical properties derivatives in equation-oriented process design is presented. It is a hybrid algorithm that makes combined use of Newton's method and the Schubert update. In doing so, available analytical derivative information is used in an optimal way.

This hybrid algorithm is surprisingly close, in terms of the number of iterations required for solution, to an implementation of Newton's method that uses finite difference approximations of any unavailable physical properties derivatives. However, the number of rigorous thermodynamics calculations is usually about 50% fewer for the hybrid method. This can result in a substantial savings for problems in which the physical properties calculations dominate the simulation time. Two examples are presented to support these claims.

ANGELO LUCIA
Department of Chemical Engineering
Clarkson College of Technology
Potsdam, NY 13676
and
SANDRO MACCHIETTO
Department of Chemical Engineering
University of Connecticut
Storrs, CT 06268

SCOPE

Newton's method is, in our opinion, the most widely accepted fixed point algorithm for equation-oriented process design calculations. It has stood the test of time, is extremely general, and is very robust. Unfortunately, in many practical design applications, complete analytical partial derivative information is either unavailable or expensive to calculate. This is especially true for physical properties (enthalpy and *K* value) derivatives with respect to composition when complex rigorous thermodynamics models are used.

The usual approaches used to attack this problem of unavailable derivative information are to either neglect it or approximate it in some way. It is well established that neglecting these derivatives, which can have significant magnitude in some cases, can increase the number of iterations needed to solve a problem or, in the worst case, cause failure. Finite difference approximations, on the other hand, usually drive up the computational cost to solve a problem because they require a large number of physical properties calculations. In many cases, thermodynamics calculations can dominate the simulation time. Finally, quasi-Newton methods, like the Broyden and Schubert updates, do not make use of any functional analytical derivative information and can behave rather erratically because of scaling.

In this work, a new approach to the approximation of these unavailable physical properties derivatives is presented. This new approach, which we simply call a hybrid method, makes optimal use of any analytical derivatives and, at the same time, approximates physical properties derivatives in a very efficient way.

Two small example problems are presented comparing our approach with a partial Newton method, in which these unavailable partial derivatives are neglected, Schubert's method and an implementation of Newton's method in which these unavailable physical properties derivatives are approximated by finite differences.

CONCLUSIONS AND SIGNIFICANCE

A new approach for approximating quantities that involve physical properties derivatives was presented. It was derived by making combined use of Newton's method and the Schubert update.

The new approach was compared with several other methods, including a finite difference implementation of Newton's method, on two small design problems. Preliminary numerical results on these, and other problems solved by both academic and industrial workers using our ideas, look extremely encouraging. Our hybrid algorithm is surprisingly close, in terms of the number of iterations required for solution, to a finite difference Newton method, while, at the same time, usually uses about 50% fewer physical properties calculations. Moreover, it appears to be very robust.

INTRODUCTION

In recent years, the equation-oriented approach to chemical engineering design has received a great deal of attention, largely because of the computational advantages it possesses (Rosen, 1980; Shacham et al., 1982). Specifically, for problems that involve recycle streams and design specifications, the more traditional sequential modular approach suffers from high computation cost due to the nested iteration that is required. The equation-oriented approach, on the other hand, requires only one level of iteration.

Correspondence concerning this paper should be addressed to A. Lucia.

It iterates, as the name implies, on the system of nonlinear equations that model the process. Consequently, problems with recycle and design constraints can be handled in a straightforward manner. Moreover, the equation-oriented approach lends itself more readily to process optimization (Berna et al., 1980).

At the heart of any equation-oriented methodology is the fixed point algorithm that is used to solve the large and sparse system of model equations. In the design area, Newton's method, the sparse Broyden (1971) or Schubert (1970) update, or some variant thereof are by far the most popular (Kubicek et al., 1976; Berna and Westerberg, 1978; Gorczynski and Hutchison, 1979). The general concensus, however, is that Newton's method with analytical partial derivatives is the preferred procedure (Shacham et al., 1982).

In principle, to use Newton's method, complete partial derivative information must be available. However, in a flowsheet environment, this is usually not the case, particularly where derivatives of physical properties (for example, K values and enthalpies) with respect to certain flowsheet variables (temperatures, pressures and compositions) are concerned. More times than not, some of these physical properties derivatives are simply not available; even when they are, they're usually prohibitively expensive to calculate.

To get around this problem, these physical properties derivatives are either neglected or approximated in some way. While neglecting them may be justified in certain situations, in other circumstances (for example, problems involving nonideal phase equilibria) such an approach can either significantly increase the number of iterations required to solve the flowsheet or even cause failure (Buzzi Ferraris and Morbidelli, 1981). Approximating these derivatives, on the other hand, can be both expensive and troublesome, especially when some finite differencing strategy is used. This is because many calls to the physical properties subroutines are needed to secure the derivative information and considerable care must be exercised when selecting the differencing step sizes for this information to be reliable.

More recently, some workers (Boston and Britt, 1978; Chimowitz et al., 1982) have reported good numerical results using local analytical models for physical properties in conjunction with an updating procedure for the model parameters. This, of course, results in analytical physical properties derivative expressions. While only limited experience with this approach is available, it appears to be a promising one.

In this paper, we suggest a new way of approximating the terms containing unavailable derivative information cheaply and yet accurately.

ILLUSTRATIVE EXAMPLE

Consider the case where a flowsheet description includes nonideal phase equilibria. Furthermore, for the sake of definiteness, consider a multistage, multicomponent distillation and, in particular, the equations

$$V_{j-1}y_{i,j-1} - (L_j + S_j)x_{i,j} - (V_j + W_j)y_{i,j} + L_{j+1}x_{i,j+1} + F_j z_{i,j} + G_j u_{i,j} = 0, \quad i = 1,2,\ldots,n_c \tag{1}$$

$$V_{j-1} - (L_j + S_j + V_j + W_j) + L_{j+1} + F_j + G_j = 0 \tag{2}$$

$$V_{j-1}h_{j-1} - (L_j + S_j)H_j - (V_j + W_j)h_j + L_{j+1}H_{j+1} + F_j\hat{H}_j + G_j\hat{h}_j + Q_j = 0 \tag{3}$$

$$y_{i,j} - K_{i,j}x_{i,j} = 0, \quad i = 1,2,\ldots,n_c \tag{4}$$

$$\sum_{i=1}^{n_c} (y_{i,j} - x_{i,j}) = 0 \tag{5}$$

where V denotes internal vapor flow rate; L, the internal liquid flow rate; S and W, liquid and vapor sidestream flow rates; F and G, the liquid and vapor feed flow rates; x, liquid mole fraction; and y, vapor mole fraction. Moreover, z and u are the liquid and vapor feed compositions, respectively; H, liquid-phase enthalpy; h, vapor-phase enthalpy; $\hat{H}$ and $\hat{h}$, the enthalpies of the liquid and vapor feed streams, respectively; and Q, the rate of heat addition. Finally, n_c is the number of components in the system, and i and j denote the component and stage indices, respectively.

Equations 1 through 5 are the component mass balance, total mass balance, energy balance, equilibrium and summation relationships for the jth tray in the column. There are, of course, similar equations for each equilibrium stage and the overhead condenser. Consequently, the steady-state behavior of the column is modelled by a reasonably large and sparse system of nonlinear equations.

We let the collection of these equations be denoted by

$$f(x) = 0. \tag{6}$$

Suppose further that we chose to solve this system of equations by Newton's method. Consequently, we are concerned with the iteration

$$B_k(x - x_k) = -f(x_k), \tag{7}$$

where B_k denotes the Jacobian matrix of f evaluated at x_k and k is an iteration number. Here x is a vector of unknown variables (i.e., internal vapor and liquid flow rates, liquid compositions and stage temperatures) and should not be confused with liquid mole fractions.

In the case of nonideal phase equilibria, the Jacobian matrix has nonzero elements that involve the partial derivatives of physical properties (K values and enthalpies) with respect to temperatures and compositions. For example, the partial derivatives of Eq. 4 with respect to composition are given by

$$\left(\frac{\partial}{\partial x_{k,j}}\right) = -\left[\delta_{k,j}K_{i,j} + x_{i,j}\left(\frac{\partial K_{i,j}}{\partial x_{k,j}}\right)\right], \quad k = 1,2,\ldots,n_c \tag{8}$$

where $\delta_{k,j}$ is the Kronecker delta which is defined by

$$\delta_{k,j} = \begin{cases} 1, & \text{if } k = i \\ 0, & \text{otherwise.} \end{cases}$$

Similarly for Eq. 3, we have

$$\left(\frac{\partial}{\partial T_j}\right) = -\left[(L_j + S_j)\left(\frac{\partial H_j}{\partial T_j}\right) + (V_j + W_j)\left(\frac{\partial h_j}{\partial T_j}\right)\right],$$

where T_j denotes the equilibrium temperature of the jth stage.

It is these derivatives $(\partial K_{i,j}/\partial x_{k,j})$, $(\partial H_j/\partial T_j)$ and $(\partial h_j/\partial T_j)$, especially the K value-composition derivatives, that are usually either unavailable or expensive to evaluate. Moreover, the usual ways of dealing with this difficulty (i.e., neglecting these derivatives or by using finite difference approximations), as mentioned, can cause failure or increase the computational cost. Instead, we suggest that the terms involving these unavailable physical properties derivatives be approximated by a suitable modification of the Schubert update. That way, we can update these terms very cheaply and, at the same time, only make one pass through the physical properties routines per flowsheet iteration.

In this light, it is convenient to think of the Jacobian matrix as having two parts, a computed part and an approximated part. For example, for the nonzero elements defined by Eq. 8, the computed part is the term $-\delta_{k,j}K_{i,j}$. Its approximated part, on the other hand, is the quantity $-x_{i,j}(\partial K_{i,j}/\partial x_{k,j})$. If we denote these matrix parts by $C = C(x)$ and A, respectively, we can define an iterative approximation to the Jacobian for a Newton-like method by the simple equation

$$B_k = C_k + A_k. \tag{9}$$

It is perhaps helpful to think of A_k as containing terms that involve physical properties derivatives.

The objective of this paper is to describe this Newton-like procedure. It represents a new way of solving design problems. We think of it as a hybrid algorithm because it makes combined use of Newton's method and the Schubert update.

NEWTON-LIKE METHOD

For the most part, the main ideas that underlie this Newton-like methodology have already been revealed. However, to avoid any confusion we organize these ideas in a formal way.

For this we let f denote the vector function that represents our design equations. Also, we let x be the vector of unknown flowsheet variables (for example, flowrates, temperatures and compositions). We denote the approximation to the Jacobian matrix of f by B and we let $C = C(x)$ and A be its computed and approximated parts, respectively. Finally, we let k denote the iteration number.

In this notation, the classical Newton iteration is defined by solving the linear system

$$C_k(x - x_k) = -f(x_k) \tag{10}$$

for the step $x - x_k$, where $B_k = C_k$ and C_k is calculated from analytical and/or finite difference derivatives. However, since for flowsheet problems complete analytical derivative information is expensive or unavailable and since there is usually a trade-off between accuracy and computational cost when the unavailable derivatives are estimated by finite differences, we suggest that Eq. 10 be replaced by Eq. 7, where B_k is defined as in Eq. 9. For design problems, C_k will include in general analytial derivatives from the mass balance equations, summation expressions and certain terms from the energy balance and equilibrium relationships. The approximated part, on the other hand, will usually consist of various physical properties derivatives.

The computed part of B_k is usually sparse and since the partial derivatives that define its nonzero elements are available, C_k is easily calculated. As for A_k, it can also be sparse with, in fact, a sparsity pattern different from that of C_k. Accordingly, we suggest that A_k be approximated by the Schubert update, which gives

$$A_{k+1} = A_k + \sum_{i=1}^{n} (\underline{s}_i^T \underline{s}_i)^+ e_i^T [y - C_{k+1}s - A_k s] e_i \underline{s}_i^T, \tag{11}$$

where $s = x_{k+1} - x_k$, $y = f(x_{k+1}) - f(x_k)$, $\underline{s}_i$ is the vector formed from s such that it has zeros in every position that the ith row of A has zeros, and the superscripts T and $+$ denote matrix transposition and the generalized inverse, respectively. Also, n is the number of rows of A.

In short, when we build A_{k+1} by Eq. 11, we first subtract from y any secant information that comes from the computed part of the Jacobian. This secant information is the matrix-vector product $C_{k+1}s$. As a result, we get a modified secant condition for A_{k+1}. In particular, A_{k+1} satisfies

$$A_{k+1}s = y - C_{k+1}s. \tag{12}$$

The interesting thing to note is that Eq. 12 implies very simply that

$$(C_{k+1} + A_{k+1})s = B_{k+1}s = y, \tag{13}$$

which is an overall secant condition. The reason we want Eq. 13 to be satisfied at each iteration of the calculations is because any quasi-Newton method satisfies such a relationship and because this secant condition is central to both the theory and numerical performance of these methods.

Now if A_k is truly null at each iteration, Eq. 7 is simply Newton's method. If, on the other hand, $C_k = 0$ for each iteration, we have Schubert's method. However, if both $A_k \neq 0$ and $C_k \neq 0$, we have a Newton-like process that satisfies the secant condition and any sparsity constraints.

The idea of viewing the Jacobian as having two parts is not new. It is something that has recently been suggested for solving nonlinear least squares problems with large residuals (Dennis, 1976; Betts, 1976; Dennis et al., 1977; Nazareth, 1980). Moreover, Marwil (1978) has suggested the same thing for systems of nonlinear equations. What is new, however, is the identification of a class of problems in chemical process design where this Newton-like strategy appears to be potentially profitable, like the one mentioned earlier.

Dennis and Walker (1981) have recently published very general convergence results for such a Newton-like method that show Q-superlinearity. In fact, their results are so general that they hold for any case (i.e., any allowable combination of nonsymmetric, symmetric, nonsparse and sparse matrices) where the matrix can be viewed as having computed and approximated parts. This includes the default cases of $A = 0$ and $C(x) = 0$ or Newton's and either Broyden's or Schubert's method respectively.

With this we give the steps of an algorithm that is based on the foregoing ideas.

NEWTON-LIKE ALGORITHM

1. Start with x_o, A_o and a convergence tolerance $\epsilon > 0$. Also, set the iteration number $k = 0$.
2. Calculate $f(x_k)$.
3. Calculate $C_k = C(x_k)$.
4. Check the convergence condition $\|f(x_k)\| \leq \epsilon$. If it is satisfied, stop. Otherwise, go to step 5.
5. Set $B_k = C_k + A_k$ and solve the linear system $B_k s_k = -f(x_k)$ for s_k.
6. Calculate an appropriate search direction d_k.
7. Set $x_{k+1} = x_k + d_k$.
8. Compute $f(x_{k+1})$ and C_{k+1}.
9. Define $s = d_k$ and $y = f(x_{k+1}) - f(x_k)$ and update A_k by Eq. 11.

We briefly describe the steps of the algorithm: Step 1 is simply an initialization step. It defines the starting guesses for the flowsheet variables, initializes the approximated part of the Jacobian (i.e., the part that involves terms with physical properties derivatives), and sets a convergence tolerance. Steps 2 and 3 simply evaluate the flowsheet equations and the corresponding analytical part of the Jacobian matrix, respectively. Step 4 checks a convergence condition. We use the two-norm of $f(x)$ as a measure of convergence. In step 5, the computed and approximated part of the Jacobian are added and the resulting linear system is solved for the Newton-like direction. The solution to the linear system is usually obtained using some variant of Gaussian elimination. In particular, we use the Yale Sparse Matrix Package—The Nonsymmetric Codes (Eisenstat et al., 1977), which calculates an LDU factorization of the coefficient matrix. Here a safeguard against matrix singularity could also be used.

Step 6 allows the use of some suitable trust region strategy. In this study, we investigated the use of the direct prediction step (i.e., setting $d_x = s_k$ for each k) and the dogleg strategy of Powell (1970). Step 7 computes the new values of the flowsheet variables, while step 8 evaluates the equations and the computed part of the Jacobian matrix. Finally, the approximated part of the Jacobian is updated in step 9.

NUMERICAL RESULTS AND DISCUSSIONS

We present some numerical results for the hybrid algorithm. Specifically, we compare its performance with that of a partial Newton (pNewton) method, in which certain physical properties derivatives have been neglected, the Schubert update and a finite difference Newton's method on a set of chemical process design problems. All examples were run on both an IBM 4341 in double-length arithmetic and a CDC Cyber 175 in single-precision arithmetic with very little difference in the numerical results. Therefore, we report the results obtained on the IBM 4341. Tables 1 and 2 show the results. Note, in these tables, the unparenthesized numbers correspond to the direct prediction step whereas those in parentheses are the results using the dogleg strategy.

The first example is taken from Chimowitz et al. (1982). It involves the determination of the dewpoint temperature and corresponding liquid-phase composition of three different multicomponent mixtures. The first (1a) is a six component mixture of n-butane, isobutane, n-pentane, n-hexane, n-heptane and n-hexadecane at a pressure of 1.5×10^6 Pa. Such a system can be considered ideal. The second (1b), on the other hand, is a nonideal mixture of ethanol, n-propanol and cyclohexane at 1.0×10^6 Pa of pressure. Finally, the third mixture (1c) is strongly nonideal and consists of acetone, water and acetonitrile at a pressure of 1.0×10^5 pascals.

The dewpoint temperature calculations consist of solving the equations

Table 1. Number of Function Calls

Problem	*p*Newton	Schubert	Hybrid	f.d.Newton
1a	5.9(5.9)	8.0(7.9)	5.9(5.9)	5.9(5.9)
1b	20.8(20.8)	*(41.1)	8.2(8.0)	4.9(5.0)
1c	19.8(19.9)	*(45.1)	8.0(7.9)	4.2(4.5)
2a	16(18)	41(62)	13(14)	5(7)
2b	19(**)	61(**)	17(**)	8(**)

* Direct prediction step failed on many examples within this problem set making comparison meaningless beyond this.

** Dogleg strategy failed for all values of trust region radius in the range 10^{-1} to 10^{+4}.

Table 2. Number of Calls to Thermodynamics Routines

Problem	*p*Newton	Schubert	Hybrid	f.d.Newton
1a	6.9(6.9)	9.0(8.9)	6.9(6.9)	6.9(6.9)
1b	21.8(21.8)	*(42.1)	9.2(9.0)	23.6(24.0)
1c	20.8(20.9)	*(46.1)	9.0(8.9)	20.8(22.0)
2a	17(19)	43(78)	14(16)	24(32)
2b	20(**)	86(**)	18(**)	36(**)

* Direct prediction step failed on many examples within this problem set making comparison meaningless beyond this.

** Dogleg strategy failed for all values of trust region radius in the range 10^{-1} to 10^{+4}.

$$K_i x_i - y_i = 0, \quad i = 1,2,\ldots,n_c$$

and

$$\sum_{i=1}^{n_c} x_i - 1 = 0,$$

for T (the dewpoint temperature) and the x_i's. Furthermore, the K values are, in all except for the ideal case, a function of both temperature and liquid composition.

For the ideal case, the Jacobian is given by

$$J = \begin{bmatrix} K_1 & & & & & x_1\left(\frac{\partial K_1}{\partial T}\right) \\ & K_2 & & & & x_2\left(\frac{\partial K_2}{\partial T}\right) \\ & & \cdot & & & \cdot \\ & & & \cdot & & \cdot \\ & & & & K_{n_c} & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial T}\right) \\ 1 & 1 & \cdot & \cdot & 1 & 0 \end{bmatrix} \tag{14}$$

Note it is sparse. The Jacobian matrices for cases 1b and 1c, on the other hand, are essentially full and defined by

$$J = \begin{bmatrix} \left[K_1 + x_1\left(\frac{\partial K_1}{\partial x_1}\right)\right] & x_1\left(\frac{\partial K_1}{\partial x_2}\right) & \cdots & x_1\left(\frac{\partial K_1}{\partial x_{n_c}}\right) & x_1\left(\frac{\partial K_1}{\partial T}\right) \\ x_2\left(\frac{\partial K_2}{\partial x_1}\right) & \left[K_2 + x_2\left(\frac{\partial K_2}{\partial x_2}\right)\right] & \cdots & x_2\left(\frac{\partial K_2}{\partial x_{n_c}}\right) & x_2\left(\frac{\partial K_2}{\partial T}\right) \\ \vdots & & \ddots & & \\ x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_1}\right) & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_2}\right) & \cdots & \left[K_{n_c} + x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_{n_c}}\right)\right] & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial T}\right) \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{15}$$

In each case, the results in Tables 1 and 2 represent the average number of iterations needed to converge to a tolerance of 10^{-5} for 20 random vapor-phase compositions. Also, the calculations were initiated by setting the liquid-phase composition equal to that of the vapor phase and the dew-point temperature to 600, 400 and 350 K for problem sets 1a, 1b and 1c, respectively. K values were calculated using the Hayden-O'Connell and UNIQUAC models for the vapor and liquid phases (Prausnitz et al., 1980). Furthermore, analytical partial derivatives of K values with respect to temperature are available by neglecting the temperature dependence of the activity and fugacity coefficients. This gives $(\partial K_i/\partial T) = K_i(\partial ln f_i^\circ/\partial T)$, where f_i° is the standard state fugacity of pure component i. In the UNIQUAC data base, $(\partial ln f_i^\circ/\partial T)$ is readily available. If the temperature derivatives of activity and fugacity coefficients are important, which can sometimes be the case, they are easily accommodated in A.

In the ideal case (1a), the partial Newton, hybrid and finite-difference Newton algorithms perform exactly the same regardless of whether they are compared on the basis of function evaluations or calls to the thermodynamic routines. Also note that because all analytical partial derivatives are available, no information is neglected in the partial Newton method, the approximated part of the Jacobian is null in the hybrid method, and no finite-difference derivatives are required; these three algorithms are equivalent to a pure Newton method. The Schubert update, on the other hand, performed slightly worse than the other three methods, and this can be attributed to the fact that any quasi-Newton method gathers secant information only in one direction per iteration. Thus, it requires a few iterations to accumulate useful secant information. Finally, the Jacobian matrix for problem set 1a was initialized to to us. However, for this measure, the performance of the hybrid the analytical Jacobian at the starting point for each of the four methods.

The two nonideal cases (1b and 1c), which are more interesting, favor the finite-difference Newton method when the number of function calls are used as a performance measure. This is no surprise algorithm is reassuringly close to that of the finite-difference Newton method. In contrast, the other two methods do a very poor job with the Schubert update being clearly the worst. In fact, for certain examples within problem sets 1b and 1c, we have observed that the partial Newton can take up to 62 iterations to reach the desired accuracy while the Schubert update can require as many as 178 iterations. The hybrid and finite difference Newton algorithms, on the other hand, never took more than 12 and six iterations, respectively, to solve any of the examples within these problem sets.

If, instead, we use the number of calls to the thermodynamics routines as a performance measure (something that is commonly done because the physical properties calculations can dominate the time required for a simulation), the results in Table 2 clearly favor the hybrid method. More specifically, the other three methods require approximately twice as many K value calculations as the hybrid algorithm for problem sets 1b and 1c. This is very encouraging.

For the two nonideal cases, all derivatives of K values with respect to composition were neglected in the partial Newton algorithm. Consequently, the Jacobian for this method had the structure and functionality defined by Eq. 14. For the quasi-Newton method, the entire Jacobian given by Eq. 15 less the constants in the last row was approximated by the Schubert update. In the hybrid method, the computed part of the Jacobian was defined by Eq. 14 while the approximated part was given by

$$A = \begin{bmatrix} x_1\left(\frac{\partial K_1}{\partial x_1}\right) & x_1\left(\frac{\partial K_1}{\partial x_2}\right) & \cdot & \cdot & \cdot & x_1\left(\frac{\partial K_1}{\partial x_{n_c}}\right) & 0 \\ x_2\left(\frac{\partial K_2}{\partial x_1}\right) & x_2\left(\frac{\partial K_2}{\partial x_2}\right) & \cdot & \cdot & \cdot & x_2\left(\frac{\partial K_2}{\partial x_{n_c}}\right) & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_1}\right) & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_2}\right) & \cdot & \cdot & \cdot & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_{n_c}}\right) & 0 \\ 0 & 0 & \cdot & \cdot & \cdot & & 0 \end{bmatrix}$$

Finally, in the finite-difference Newton method the partial derivatives of K values with respect to composition were, of course, approximated by finite difference derivatives.

For both nonideal problem sets, the Jacobian matrix was initialized to the analytical Jacobian defined by Eq. 14 for the partial Newton method, the Schubert update and the hybrid algorithm. Note this corresponds to $A_o = 0$ for the hybrid method. For the finite-difference Newton method, the K value derivatives with respect to composition were approximated by finite-difference derivatives on all iterations. Hence, for this method the initial Jacobian matrix was slightly different than that for the other three methods.

To further stress the importance of K value derivatives with respect to composition in these kinds of calculations and to exemplify the behavior recently reported by Buzzi Ferraris and Morbidelli (1982) in their paper on three phase distillation, we present a worst-case example from problem set 1b. In particular, consider the dew-point temperature calculation for the system ethanol (1), n-propanol (2) and cyclohexane (3) at a pressure of 1.0×10^6 Pa and let the vapor phase be defined as $y_1 = 0.48939$, $y_2 = 0.07919$, and $y_3 = 0.43142$. Also let the initial estimate of the liquid phase and Jacobian matrices for the various methods be defined as above. Finally, let the initial dew-point temperature estimate be T_o = 413.88 K, which is the true dew-point temperature for this mixture.

When we solve this particular problem by the partial Newton, Schubert, hybrid and finite-difference Newton methods in conjunction with the dogleg, we find that they require 62, 122, seven and three iterations, respectively. However, it is important to note that for this problem the dew-point temperature never differs from the true dew point by more than 0.62 K for any of the four methods. Consequently, the value of the dew-point temperature is for the most part exact and does not affect the iteration history of any of the methods. Rather, it is the composition derivatives of the K values that have a direct bearing on the calculations. Specifically, for the partial Newton method, the dew-point temperature reaches its true value after 11 iterations but requires 51 more iterations to converge the liquid-phase compositions. To our surprise, the Schubert update does even worse requiring an unbelievable 122 iterations. In fact, without the dogleg strategy, Schubert's method fails to solve this problem due to an excessively large temperature correction that in turn results in meaningless physical properties calculations. In contrast, the hybrid and finite-difference Newton methods solve this problem in only a few iterations. This we attribute to the fact that both of these methods approximate K value composition derivatives accurately. Again, it is important to note, however, that the finite-difference Newton method requires 16 calls to the K value routines while the hybrid algorithm requires only eight.

We close the discussion of problem 1 with a number of observations. First, it has been our experience that it is best to use all available first partial derivative information and that accurate K value temperature derivatives are required for good numerical performance in dew-point calculations. Furthermore, when the initial dew-point temperature estimate is within the two-phase region, we believe that this results in accurate affine approximations to the model equations. This, in our opinion, is why the dogleg strategy has virtually no effect on the performance of the partial Newton, hybrid and finite-difference Newton methods. Moreover, we believe that the absence of accurate derivative information (especially K value temperature derivatives) is the reason that the Schubert update requires the dogleg strategy for numerical stability.

Second, the hybrid method appears to be exceedingly robust on this type of problem. It also appears to retain much of the scale invariance of the finite-difference Newton method, provided the computed and approximated parts of the Jacobian are defined in the way we have defined them. However, we have found it beneficial to start the algorithm differently depending on the initial dew-point estimate. More specifically, for initial temperature estimates within or near the two-phase region, it seems sufficient to let $A_o = 0$ and to use the direct prediction step. If, instead, the initial temperature is well outside the two-phase region, it has been our experience that it is best to let A_o be calculated by finite differences and to use the dogleg strategy. We have found that 10^2 is a reasonable value for the trust region radius. If this safeguard is used, we find virtually no change in the performance of the hybrid method (i.e., it still requires on the average eight function calls to reach a solution) regardless of the initial temperature guess. For example, we have solved problem 1b with arbitrary initial temperature estimates in the range 250 to 450 K and observed no change in performance.

The second example is taken from Kerr (1981). It involves the simulation of a five-effect countercurrent evaporator system used to concentration a solution of sodium hydroxide from 10 to 40 wt. %.

For the jth effect we have the following equations:

$$L_{j-1}x_{j-1} - L_jx_j = 0 \quad (16)$$

$$L_{j-1} - L_j - V_j = 0 \quad (17)$$

$$L_{j-1}H_{j-1} - L_jH_j - V_jh_j + V_{j+1}(h_{j+1} - \hat{H}_j) = 0 \quad (18)$$

$$V_{j+1}(h_{j+1} - \hat{H}_j) - U_jA_j\Delta T_j = 0 \quad (19)$$

$$\Delta T_j - 0.461(\hat{T}_j - T_j) = 0 \quad (20)$$

$$T_j - 4.2 - 0.549\hat{T}_{j-1} + 34.7\,x_j - 0.069\hat{T}_{j-1}x_j - 208x_j{}^2, \quad (21)$$

where L is the liquid flow rate, V is the vapor flow rate, and x represents the concentration of the sodium hydroxide solution (weight fraction). Moreover, H and h are the liquid and vapor enthalpies, $\hat{H}$ denotes the condensate enthalpy, T and $\hat{T}$ are the corresponding temperatures of the liquid solution and condensate streams, and ΔT is the temperature gradient across the condenser tubes. Finally, U and A denote the overall heat transfer coefficient and heat transfer area, respectively. The expressions defining the physical properties (enthalpies and overall heat transfer coefficients) are given in the paper by Kerr. Furthermore, they are nonlinear functions of composition and temperature and, in the case of overall heat transfer coefficients, have composition derivatives of significant magnitude.

In addition to the above equations, there are three feed specifications and two design constraints. These are given by

$$L_o - 7.554 = 0 \quad (22)$$

$$x_o - 0.1 = 0 \quad (23)$$

$$T_o - 355.4 = 0 \quad (24)$$

$$x_5 - 0.4 = 0 \quad (25)$$

$$\hat{T}_o - 310.9 = 0. \quad (26)$$

Consequently, for this simulation problem, we have a system of 35 equations, mostly nonlinear, which we wish to solve for the liquid and vapor flow rates, compositions, effect temperatures, condenser temperature gradients, and condensate temperatures. Note that the physical properties expressions are not considered to be part of the system of nonlinear equations. Rather, they are assumed to be computed elsewhere as would be the case if a physical properties data base system was used.

Given this and the empirical nature of the physical properties expressions, we decided that for this problem that all physical properties derivatives, and therefore terms involving these derivatives, were unavailable. While this is neither true here nor again in the usual situation in which a physical properties data base is used, we nevertheless felt that it was a reasonable assumption. As a result, out of the 153 nonzero elements in the true Jacobian, 100 could be computed from analytical expressions and 53 were either neglected or approximated in some way. More specifically, the computed part of the Jacobian consisted of simple analytical partial derivatives obtained by differentiating the model equations. The approximated part, on the other hand, was comprised of any expressions involving physical properties derivatives. For example, the partial derivatives of Eq. 19 with respect to ΔT_j is given by

$$\frac{\partial}{\partial \Delta T_j} = -U_jA_j$$

and is therefore an element in the computed part of the Jacobian. The partial derivative of the same equation with respect to composition is

$$\frac{\partial}{\partial x_j} = -A_j\Delta T_j\left(\frac{\partial U_j}{\partial x_j}\right)$$

and is part of the approximated part since it involves a physical property derivative.

We solved the foregoing problem again by all four methods for two different starting points; once for a reasonably good set of initial estimates of the variables and then again for a poor starting point. In all cases the convergence tolerance was 10^{-8} and, where applicable, an initial trust region radius of 10^3 was used for the dogleg strategy. The results are shown in Tables 1 and 2 (problem 2a is the good starting point and 2b is, therefore, the poor starting point).

For problem 2a the variables were initialized as follows: $L_o = 7.554$, $x_o = 0.1$, $T_o = 355.4$, $\hat{T}_o = 310.9$, for $j = 1,2,\ldots,5$ the variables associated with the jth effect were set to

$$L_j = L_{j-1} - 1.133, \quad V_j = 1.133, \quad x_j = x_{j-1} + 0.06$$

$$T_j = T_{j-1} + 5.56, \quad \hat{T}_j = T_j + 5.56 \quad \text{and} \quad \Delta T_j = 5.56.$$

V_6 was assigned a value of 1.133 kg/s. Finally, the Jacobian approximation for all four methods was initialized in the same manner as in the dew-point temperature problems. In particular, $A_o = 0$ for the hybrid method.

For this starting point, Table 1 clearly shows that the finite difference Newton method requires considerable fewer function calls to solve this problem than any of the other methods. This, again, is no surprise to us. In particular, the finite difference Newton method requires only five function calls using the direct prediction step and seven function calls when the dogleg is used. The hybrid algorithm is second to the finite-difference Newton method when the number of function calls is used as a performance measure. It requires 13 and 14 function calls for the direct prediction step and dogleg strategy, respectively. To our surprise, the partial Newton method does well on this problem, even for the poor starting point. Finally, Schubert's method, as in the case of the dew-point calculations, is the worst of the four methods requiring 41 function calls using direct prediction and 62 using the dogleg. This is three times as many function calls as either the partial Newton or hybrid method use and four times that for the finite-difference Newton method. We attribute the poor performance of the Schubert update to the fact that it is a fixed scale update and the true Jacobian is ill-conditioned, largely due to the combined presence of mass and energy balance equations.

When the number of thermodynamics calls is used as the performance measure, Table 2, the hybrid algorithm is the best strategy. It requires approximately 42% fewer physical properties evaluations than the finite difference Newton method. Again, this is very encouraging.

For this problem the partial Newton method requires only slightly more physical properties evaluations than the hybrid method. This is still 30% better than the finite-difference Newton method and shows that neglecting physical properties derivatives can often be an economical strategy. However, when it is and when it's not a viable strategy strongly depends on the particular problem being solved. Schubert's method, on the other hand, requires substantially more physical properties evaluations than any of the other methods.

Finally, we remark that the dogleg strategy has virtually no effect on the partial Newton, hybrid and finite difference Newton methods. However, for the Schubert update, the dogleg causes a 50% increase in function calls and rigorous thermodynamics calculations even though a good initial Jacobian approximation was used. This, again, we attribute to the fact that the Schubert update, which is a fixed scale update, is being used to update an ill-conditioned matrix. It simply does not track changes that well.

Problem 2b in Tables 1 and 2 shows the results for, what we consider, poor starting values for the variables. In particular, the variables were initialized as follows: L_o, x_o, T_o and $\hat{T}_o$ were set as in problem 2a; then for $j = 1,2,\ldots,5$ the variables associated with the jth effect were defined by

$$L_j = L_{j-1}, \quad V_j = 0, \quad x_j = x_{j-1}$$

$$T_j = T_{j-1}, \quad \hat{T}_j = T_j \quad \text{and} \quad \Delta T_j = 44.44.$$

V_6 was also set equal to zero. Moreover, the Jacobian approximation for each of the four methods was initialized as before. We note that this particular starting point is far from the solution, yet the corresponding two-norm of the model equations is quite small compared to, say, the two-norm of the same equations for the starting point for problem 2a. This, as shown in the tables, causes serious problems for the dogleg strategy.

For the direct prediction step, the results are much the same as those in problem 2a. The finite difference Newton method does best when the number of function calls is used as a performance measure. The hybrid method, on the other hand, is the clear winner when the number of physical properties evaluations is counted. In particular, the hybrid algorithm requires 50% fewer thermodynamics calculations than the finite-difference Newton method. Moreover, the partial Newton method does very well, as it did on problem 2a, and the Schubert update, as shown in the tables, does poorly.

Finally, we close our discussion of problem 2 with the remark that the dogleg strategy fails on problem 2b regardless of the underlying fixed point algorithm and initial value of the trust region radius. This we attribute to the fact that the initial two-norm of the model equations is small (on to the the order of 10) and the demand by the dogleg strategy to reject any point that causes an increase in the two-norm keeps the vector of independent variables in a region far from the solution. If, instead, we track the path for the direct prediction step for the same problem, we see that the two-norm increases substantially before the variables begin to converge to the solution.

In summary, we have presented a number of results, some of which are new and others that either support or conflict with existing attitudes. First and foremost, we have presented a new way of approximating certain physical properties derivatives in equation-oriented process design calculations. This new method, which we simply call a hybrid method, makes combined use of Newton's method and the Schubert update. It also uses analytical derivative information in an optimal way. Preliminary numerical results look very promising. For the problems tested, the hybrid method is surprisingly close to a finite-difference implementation of Newton's method in terms of function calls, while, at the same time, usually requires 50% fewer rigorous physical properties calculations. Also, Taylor et al. (1982) have used our algorithm in multicomponent mass transfer calculations with the same success (i.e., significantly fewer rigorous thermodynamics calculations than a finite-difference Newton method). Furthermore, workers at Union Carbide Corporation have used our algorithm on various types of flash calculations and reported to us that it is both robust and saves on the number of rigorous thermodynamics calculations. All things considered, we think it is the best algorithm to use at the present time on these small problems.

Because we have, as yet, only solved some relatively small problems, we are reluctant to speculate on the performance of our approach on large problems, in particular, multicomponent multistage separations calculations. We are currently in the process of solving an industrial distillation in which the overhead product is an azeotrope. We prefer to let those results speak for themselves. We are hopeful that our approach will be equally successful.

To a lesser extent, we have shown that the partial Newton method can perform rather erratically. Although it's invariant under affine transformations of the variables, its behavior is highly problem-dependent. This agrees with results previously published in the literature (Buzzi Ferraris and Morbidelli, 1982) and, in our opinion, is due to the fact that the partial Newton method does not usually satisfy the secant condition in the limit. Moreover, we were quite surprised at the poor performance of the Schubert update. It was the worst of the four methods tested. This conflicts with reports by other workers (for example, Gallun and Holland, 1980) that show the relatively favorable performance of Schubert's method. We feel further investigation is warranted here and that scaling is the most important consideration because, in this case, the secant condition is satisfied. Finally, problem 2b shows that the dogleg, or demanding norm reduction, is not always a good philosophy. Apparently, there are no clear-cut answers. That's what makes doing all of this a lot of fun.

ACKNOWLEDGMENTS

We would like to thank J. E. Dennis for sending us an early copy of a paper which prompted this work and T. F. Anderson for giving us a copy of his thermodynamics data base. Also, we'd like to thank the Control Data Corp. for partial support of this work.

NOTATION

A_k, A_j = approximated part of the Jacobian at kth interation, heat transfer area for jth effect
B_k = Jacobian approximation at kth iteration
C_k = computed part of the Jacobian at the kth iteration
d_k = search direction
e_i = ith unit vector
f, f_i^o = vector function of nonlinear equations, standard state fugacity of pure component i
F_j = liquid feed flow rate to the jth stage of distillation column
G_j = vapor feed flow rate to the jth stage of distillation column
$h_j, \hat{h}_j$ = enthalpy of the jth internal vapor stream and jth vapor feed stream
$H_j, \hat{H}_j$ = enthalpy of the jth internal liquid stream and jth liquid feed stream or condensate stream
J = Jacobian matrix of f
$K_i, K_{i,j}$ = equilibrium ratio for the ith component, on the jth stage
L_j = liquid flow rate leaving the jth stage
s, s_k = change in the dependent variables or equivalently the Newton-like step, at the kth iteration
S_j = liquid sidestream flow rate leaving the jth stage
$T_j, \hat{T}_j$ = temperature of the jth effect, condensate temperature for the jth effect
ΔT_j = temperature gradient for the jth effect
$u_{i,j}$ = vapor feed composition of the ith component for the jth stage
U_j = overall heat transfer coefficient for the jth effect
V_j = internal vapor flow rate leaving the jth stage or effect
W_j = vapor sidestream flow rate for the jth stage
x, x_k, x_i, x_{ij} = vector of unknown variables, at the kth iteration, liquid composition of component i, on the jth tray
$y, y_{i,j}$ = change in vector function f from one iteration to the next, vapor composition of component i on the jth stage
$z_{i,j}$ = composition of component i for the liquid feed to stage j

Greek Letters

$\delta_{k,j}$ = Kronecker delta
ϵ = convergence tolerance

Subscripts

i = component number
j = stage number
k = iteration counter

Superscripts

o = standard state
T = transpose
$+$ = pseudoinverse

LITERATURE CITED

Berna, T. J., M. H. Locke, and A. W. Westerberg, "A new approach to optimization of chemical processes," *AIChE J.*, **26**, 37 (1980).

Betts, J. T., "Solving the nonlinear least squares problem: Application of a general method," *J. Opt. Theory Appl.*, **18**, 469 (1976).

Boston, J. F., and H. I. Britt, "A radically different formulation and solution of the single-stage flash problem," *Comp. Chem. Eng.*, **2**, 109 (1978).

Broyden, C. G., "The convergence of an algorithm for solving sparse nonlinear systems," *Math. Comp.*, **25**, 285 (1971).

Buzzi Ferraris, G., and M. Morbidelli, "Distillation models for two partially immiscible liquids," *AIChE J.*, **27**, 881 (1981).

Chimowitz, E. H., S. Macchietto, T. F. Anderson, and L. F. Stutzman, "Local models for representing phase equilibria in multicomponent, nonideal vapor-liquid and liquid-liquid systems. Part II: Applications to process design," to appear in *IEC Proc. Des. Dev.*

Dennis, J. E., "A brief survey on convergence results for quasi-Newton methods," *Nonlinear Programming*, R. Cottle and C. Lemke, Eds., SIAM-AMS Proceedings, **9**, 185 (1976).

Dennis, J. E., D. M. Gay, and R. E. Welsch, "An adaptive nonlinear least squares algorithm," Tech. Report No. TR77-321, Cornell Univ., Ithaca, NY (1977).

Dennis, J. E., and H. F. Walker, "Convergence theorems for least change secant update methods," Tech. Report No. TR-476-(141-171-163)-2, Rice Univ., Houston (1979).

Eisenstat, S. C., M. C. Gursky, M. H. Schultz, and A. H. Sherman, "Yale Sparse Matrix Package—Nonsymmetric Codes," Report No. 114, Yale University, New Haven, CT (1977).

Gallun, S. E., and C. D. Holland, "A modification of Broyden's method for the solution of sparse systems—with application to distillation problems described by nonideal thermodynamic functions," *Comp. Chem. Eng.*, **4**, 93 (1980).

Gorczynski, E. W., and H. P. Hutchison, "Towards a quasilinear process simulator—I. Fundamental ideas," *CACE*, **2**, 189 (1978).

Kerr, C. P., "Aid to iterative calculations," *Hydrocarbon Proc.*, 145 (1981).

Kubicek, M., V. Hlavacek, and F. Prochaska, "Global modular Newton-Raphson technique for simulation of an interconnected plant applied to complex rectification columns," *Chem. Eng. Sci.*, **31**, 277 (1976).

Marwil, E. S., "Exploiting sparsity in Newton-like methods," Ph.D. Thesis, Cornell University, Ithaca, NY (1978).

Nazareth, L., "Some recent approaches to solving large residual nonlinear least squares problems," *SIAM Rev.*, **22**, 1 (1980).

Powell, M. J. D., "A hybrid method for nonlinear equations," *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, Ed., Gordon and Breach, London (1970).

Prausnitz, J., T. Anderson, E. Grens, C. Eckert, R. Hsieh, and J. O'Connell, *Computer Calculations for Multicomponent Vapor-Liquid and Liquid-Liquid Equilibria*, Prentice-Hall (1980).

Rosen, E. M., "Steady state chemical process stimulation: A state-of-the-art review," *Computer Applications to Chemical Engineering*, R. G. Squires and G. V. Reklaitis, Eds., ACS, Washington, DC, 3 (1980).

Schubert, L. K., "Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian," *Math. Comp.*, **24**, 27 (1970).

Shacham, M., S. Macchietto, L. F. Stutzman, and P. Babcock, "Equation oriented approach to process flowsheeting," *Comp. Chem. Eng.*, **6**, 79 (1982).

Taylor, R., A. Lucia, and R. Krishnamurthy, "A Newton-like algorithm for the efficient estimation of rates of multicomponent condensation by a film model," to appear in *I. Chem. E. Symp. Ser.*, No. 75.

Westerberg, A. W., and T. J. Berna, "Decomposition of very large-scale Newton-Raphson based flowsheeting problems," *Comp. Chem. Eng.*, **2**, 61 (1978).

Manuscript received August 24, 1982; revision received November 9, and accepted December 14, 1982.

Heat Transfer Around a Horizontal Tube in Freeboard Region of Fluidized Beds

An experimental investigation was carried out to measure local and average heat transfer coefficients for horizontal tubes located in freeboard region of air fluidized beds. Tests were carried out at room temperature and atmospheric pressure in a rectangular fluidized bed, with mean particle diameters of 275 to 850 μm.

Both local and average heat transfer coefficients were found to vary with particle diameter, flow rate, static bed depth, and elevation in the freeboard region.

S. BIYIKLI, K. TUZLA, and J. C. CHEN

Institute of Thermo-Fluids Engineering and Science
Lehigh University
Bethlehem, PA 18015

SCOPE

In fluidized-bed combustors, the heat transfer tubes are located with a portion of the tubes submerged within the fluidized bed and the remaining portion placed in the freeboard space above the fluidized bed. The effective heat transfer coefficients for tubes located within the bed region can be an order of magnitude greater than the heat transfer coefficients for tubes in the freeboard space. Designers hope to utilize this different behavior to adjust operating power level of fluidized-bed combustors. Current plans call for "turning-down" the thermal output by dropping the fluidized-bed level to expose greater portions of the coolant tubes in the freeboard.

The rate of heat transfer between a fluidized bed and a submerged tube has been measured by a number of investigators for uniformly sized particles (Vreedenberg, 1958; Ainshtein, 1966; Andeen and Glicksman, 1976; Chen, 1976; Ozkaynak and Chen, 1978, 1980; Saxena et al., 1978; Staub, 1979; Chandran and Chen, 1980; Grewal and Saxena, 1980) and for mixed particle sizes (Biyikli and Chen, 1982). Heat transfer to tubes in freeboard region of fluidized beds depends on a great number of factors, including the properties of the bed material and the fluid, bed and tube geometries, tube elevation and fluidization state. To our knowledge, only a few data are available for circumferentially averaged heat transfer coefficients to tubes in freeboard region of fluidized beds (George and Grace, 1979; Wood et al., 1980; Byam et al., 1981), and there is no detailed information about the local heat transfer coefficients.

The objective of the present investigation was to measure the local and circumferentially averaged heat transfer coefficients for a horizontal tube placed in the freeboard region of fluidized beds, for different test particles and over a range of gas velocities and at different tube elevations.

K. Tuzla is on leave from Technical University of Istanbul.
Correspondence concerning this paper should be addressed to J. C. Chen.