where t_D is the mean of the prototype delay distribution $g(t)$

$$t_D = \int_0^\infty t g(t) dt \tag{13}$$

and $\bar{n}$ is the average number of hops executed by fluid elements:

$$\bar{n} = (1 + \alpha h) \sum_{n=0}^{N} n(\alpha h)^n \frac{(N-n)^n}{n!} e^{-\alpha h(N-n)} \tag{14}$$

Other methods of dealing with the end conditions are possible once specific assumptions have been made concerning the way in which fluid elements leave (and enter) the test section; that given here is simple and suitable for many practical purposes.

A considerable attraction of the time delay concept in residence time distribution modeling lies in the ease with which the postulated underlying mechanisms can be modified to suit individual circumstances without in any way destroying the extreme simplicity of the analysis; in this respect at least it compares very favorably with other commonly used abstractions of the flow-mixing process. This inherent flexibility is highlighted once again in the above derivation of the general hopping model; just as for the original time delay description, Eq. 9 respresents a family of models individually defined by their delay distribution $g(t)$. It has been used (Rathor, 1969) with the exponential delay distribution

$$g(t) = \frac{1}{t_D} e^{-t/t_D} \tag{15}$$

$$g_n(t) = \frac{t^{n-1e-t/t_D}}{t_D^n(n-1)!} \tag{16}$$

to describe flow in trickle beds, where it is shown that, with hopping distances of the same order as observed liquid rivulets (h = 3 to 5 cm), significant improvements over the corresponding time delay model response can be obtained.

NOTATION

$f(t)$ = residence time distribution, s^{-1}
$f^*(t)$ = defined by Eq. 6, s^{-1}
$g(t)$ = prototype delay distribution, s^{-1}
$g_n(t)$ = n-fold convolution of $g(t)$, s^{-1}
h = hopping distance, m
n = index, number of delays or hops
$\bar{n}$ = average number of hops
N = maximum possible number of hops
$p_n(x)$ = probability of n delays, or hops, occurring in distance x
P = probability, defined by Eq. 7
t = time, s
t_0 = maximum main path transit time, s
t_{0n} = main path transit time for elements that make n hops, s
$\bar{t}$ = mean residence time, s
t_D = mean delay time, s
$u(t)$ = unit step function: $t < 0$, $u = 0$; $t \geqslant 0$, $u = 1$
x = distance, m

Greek Letters

α = defined in discussion above Eq. 1, m^{-1}
θ = normalized time
$\phi(\theta)$ = normalized residence time distribution

LITERATURE CITED

Buffham, B. A., and Gibilaro, L. G., "A Unified Time Delay Model for Dispersion in Flowing Media," *Chem. Eng. J.*, **1**, 31 (1970).

Buffham, B. A., Gibilaro, L. G., and Rathor, M. N., "A Probabilistic Time Delay Description of Flow in Packed Beds," *AIChEJ*, **16**, 218 (1970).

Porter, K. E., "Liquid Flow in Packed Columns," *Trans. Inst. Chem. Engrs.* **46**, T69 (1968).

Rathor, M. N., "Stochastic and Related Models for the Residence Time Distribution in Trickle Flow in a Packed Bed, Ph.D. Thesis, Loughborough Univ. of Technology, England (1969).

Manuscript received July 12, 1983; revision accepted Oct. 20.

The Behavior of a Hybrid Fixed-Point Method in a Chemical Process Design Environment

D. C. MILLER AND
ANGELO LUCIA

Department of Chemical Engineering
Clarkson College of Technology
Potsdam, NY 13676

INTRODUCTION

In a recent paper, Lucia and Macchietto (1983) suggested the use of a hybrid fixed-point method for solving equation-oriented process design problems. This hybrid method, which makes combined use of Newton's method and the Schubert (1970) update, was subsequently applied to condenser design problems modeled by simultaneous heat and mass transfer by Taylor et al. (1983), to flash and distillation calculations by Westman et al: (1984), and to multistage, multicomponent separation problems by Lucia and Westman (1984). All of the numerical evidence gathered so far indicates that the hybrid method is both computationally efficient and robust. That is, it has displayed the robustness of Newton's method while, at the same time, has generally used significantly fewer rigorous physical physical properties calculations. Moreover, the hybrid method has been shown to be both more robust and

Correspondence concerning this paper should be addressed to A. Lucia more efficient than Schubert's original method.

From the outset, we have been overwhelmingly surprised by the good numerical performance of the hybrid method and, quite naturally, have wondered why this is so. After all, if the hybrid method uses a modification of the Schubert update and Schubert's method can perform rather poorly, then why should the hybrid approach give consistently good numerical results? Is it solely because we are making use of some analytical first partial derivative information? Or can it have something to do with the natural scaling of the variables, since it is well known that Schubert's original method is not invariant with respect to general affine transformations of the variables? In this note we shed some light on these very basic questions. Most importantly, we give some plausible arguments that we feel suggest that the natural scale of the variables in chemical process design problems can have a very definite impact on the numerical performance of the hybrid method. We illustrate the effects of scaling by using two different implementations of the hybrid approach to solve a set of dew-point problems for a ternary mixture with nonideal thermodynamics.

THE HYBRID APPROACH

In short, the principal idea behind the hybrid approach is a simple one. If x denotes a vector of unknown process variables and if f denotes a vector function of model equations, then iterative approximations, say $\mathbf{B}_k$, to the Jacobian matrix J(x) are defined by the rule

$$\mathbf{B}_k = \mathbf{C}_k + \mathbf{A}_k \tag{1}$$

where k is an iteration number. The matrix $\mathbf{C}_k = \mathbf{C}(\mathbf{x}_k)$ is called the computed part of the Jacobian and consists of all of the first partial derivative information that is readily available in analytical form. $\mathbf{A}_k$, on the other hand, is called the approximated part, contains any derivative information that is either unavailable in analytical form or expensive to calculate, and is built by the following straightforward modification of the Schubert update:

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \sum_{i=1}^{n} (\mathbf{s}_i^T\mathbf{s}_i)^+ \mathbf{e}_i^T[\mathbf{y} - \mathbf{C}_{k+1}\mathbf{s} - \mathbf{A}_k\mathbf{s}]\mathbf{e}_i\mathbf{s}_i^T \tag{2}$$

where $\mathbf{s} = \mathbf{x}_{k+1} - \mathbf{x}_k = -\mathbf{B}_k^{-1}\mathbf{f}(\mathbf{x}_k)$, $\mathbf{y} = \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k)$; $\mathbf{s}_i$ is the vector formed from s such that it has zeros in every position that the ith row of A does; and + and T denote the generalized inverse and transposition, respectively.

For example, for a simple dew-point calculation, we have

$$K_i x_i - y_i = 0 \qquad i = 1,2,\ldots,n_c \tag{3}$$

and

$$\sum_{i=1}^{n_c} x_i - 1 = 0. \tag{4}$$

Accordingly, the computed and true approximated parts of the Jacobian matrix are

$$\mathbf{C}_k = \begin{bmatrix} K_1 & & & & x_1k_1\left(\frac{\partial \ln f_1^0}{\partial T}\right) \\ & K_2 & & & x_2K_2\left(\frac{\partial \ln f_2^0}{\partial T}\right) \\ & & \ddots & & \vdots \\ & & & K_{n_c} & x_{n_c}K_{n_c}\left(\frac{\partial \ln f_{n_c}^0}{\partial T}\right) \\ 1 & \cdot & \cdot & 1 & 0 \end{bmatrix} \tag{5}$$

and

$$\mathbf{A}_k = \begin{bmatrix} x_1\left(\frac{\partial K_1}{\partial x_1}\right) & x_1\left(\frac{\partial K_1}{\partial x_2}\right) & \cdots & x_1\left(\frac{\partial K_1}{\partial x_{n_c}}\right) & x_1K_1\left[\left(\frac{\partial \ln\partial\gamma_1}{\partial T}\right) - \left(\frac{\partial \ln\phi_1}{\partial T}\right)\right] \\ x_2\left(\frac{\partial K_2}{\partial x_1}\right) & x_2\left(\frac{\partial K_2}{\partial x_2}\right) & \cdots & x_2\left(\frac{\partial K_2}{\partial x_{n_c}}\right) & x_2K_2\left[\left(\frac{\partial \ln\partial\gamma_2}{\partial T}\right) - \left(\frac{\partial \ln\phi_2}{\partial T}\right)\right] \\ & \vdots & & & \vdots \\ x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_1}\right) & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_2}\right) & \cdots & x_{n_c}\left(\frac{\partial K_{n_c}}{\partial x_{n_c}}\right) & x_{n_c}K_{n_c}\left[\left(\frac{\partial \ln\gamma_{n_c}}{\partial T}\right) - \left(\frac{\partial \ln\phi_{n_c}}{\partial T}\right)\right] \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{6}$$

assuming that the liquid and vapor phases are nonideal.

SOME PRELIMINARY NUMERICAL ANALYSIS

For the purpose of illustration, we consider two different implementations of the hybrid method, one in which $\mathbf{A}_k$ has the structure and functionality defined by Eq. 6 and a second implementation in which $\mathbf{A}_k$ contains only the K value-composition derivatives. Note that the first implementation is in fact more rigorous than the second, in a thermodynamic sense. For convenience, we denote these two implementations of the hybrid method with ∂T and without ∂T, respectively.

Table 1 shows results for a set of ten dew-point calculations for the ternary mixture of acetone, acetonitrile, and water at 1 × 10^5 pascals. We solved each of the ten problems in the set by both implementations of the hybrid method as well as with a partial Newton method (i.e., $\mathbf{A}_k = 0$ for each k), Schubert's original method, and Newton's method with $\mathbf{A}_k$ calculated by finite differences to an accuracy of $\|f\|_2 \leq 10^{-5}$. Moreover, for each problem we generated a random, normalized vapor composition, set

TABLE 1. RESULTS FOR DEW-POINT TEMPERATURE CALCULATIONS

		Iterations			
Problem	p Newton	Schubert	Hybrid $\overline{w}$ ∂T	Hybrid $\overline{w}$out ∂T	f.d. Newton
1	41	24(20)	14(F)	8(7)	5
2	17	F(37)	12(F)	10(8)	5
3	40	28(29)	20(23)	8(7)	5
4	19	9(9)	10(9)	7(5)	4
5	9	8(6)	7(5)	5(5)	3
6	21	19(27)	23(13)	8(6)	5
7	12	31(F)	20(15)	7(6)	4
8	5	6(6)	5(4)	4(4)	4
9	38	23(31)	13(16)	7(7)	5
10	40	16(15)	10(8)	7(6)	4

F = failure.

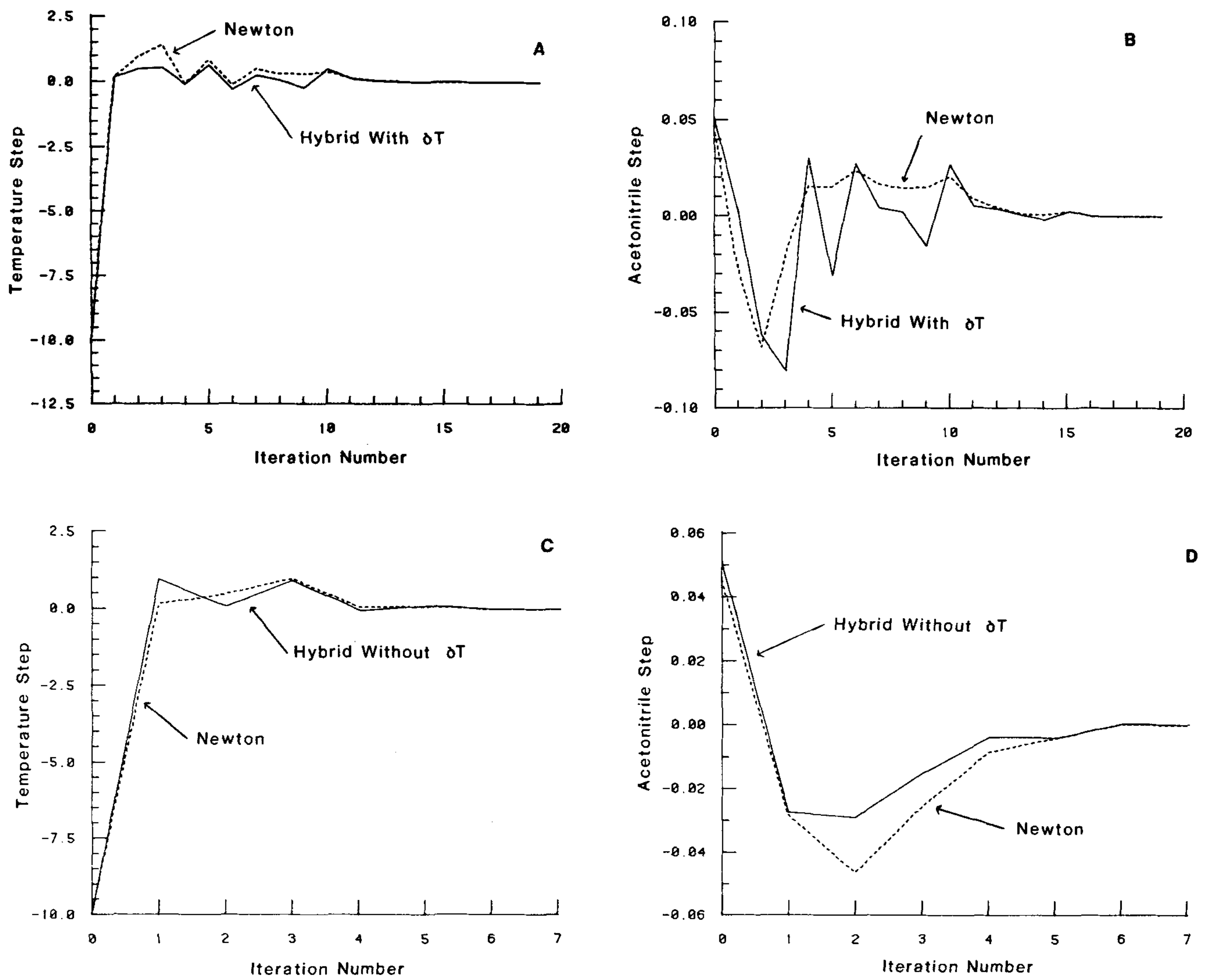

Figure 1. Iteration history for the changes in temperature and acetonitrile composition for example 3.

$x_i = y_i$, $i = 1,2, \ldots, n_c$ and $T_0 = 350$ K. Finally, for Schubert's method and each of the two implementations of the hybrid method, we investigated both $\mathbf{B}_0 = \mathbf{C}(\mathbf{x}_0)$ and $\mathbf{B}_0 = \mathbf{J}(\mathbf{x}_0)$. The numbers in parentheses are those for $\mathbf{B}_0 = \mathbf{J}(\mathbf{x}_0)$.

While there are many interesting aspects of the results shown in the table, we wish to concentrate on only two, the effects of both using readily available analytical derivative information and the scaling of the variables. First, the difference between the results for Schubert's method and those for the hybrid method with $\mathbf{A}_k$ defined by Eq. 6 shows that, in general, a fair amount of the good numerical performance of the hybrid method can be attributed to the use of readily available analytical derivative information. However, it does not explain all of the good behavior. Like Schubert's original method, the first implementation of the hybrid method exhibits what we call erratic behavior. Sometimes it does well; other times it does not.

This, quite naturally, brings us to the second point of interest, the effects of the scaling of the variables. As we said, the first implementation of the hybrid method (i.e., hybrid with ∂T) exhibits erratic behavior. The second (i.e., hybrid without ∂T), on the other hand, behaves much like the finite-difference Newton method, displaying a very small variance in the number of iterations required to converge to the desired accuracy. That we believe the deterioration in the performance of the first implementation is caused by the inherent poor scaling between temperature and composition stems from the following observations. For problems in which the hybrid method with ∂T does relatively poorly and in which it appears that the K value-composition derivatives are important for good numerical behavior, say for example, problem 3, we usually observe large steps in temperature in comparison to steps in composition during the early iterations. Because temperature and composition are coupled with regard to the quasi-Newton update, the net effect of these large steps in temperature is a masking of any K value-composition derivatives. In essence, we get little or no composition derivative information. After a few iterations, the temperature steps reach a point in which they are perhaps comparable to or only slightly larger than those for composition. However, at this point the temperature derivatives in $\mathbf{A}$ usually become distorted (i.e., several orders of magnitude too large and in some cases the wrong sign) and this distortion often causes subsequent iterative changes in both temperature and composition to behave erratically even if good composition derivative information is being gathered. Thus, the masking of composition derivatives and the distortion of the temperature derivatives, which are both manifestations of the sensitivity of the modified Schubert update to the natural scaling of the variables and which occur in series, can cause slow and erratic convergence, respectively. That is, many iterations can be required for solution and sometimes failure can occur.

In contrast, the second implementation (i.e., hybrid without ∂T) is, *by design*, void of any temperature-composition interactions with regard to the update. Consequently, we observe no masking of the composition derivative information and no distortion of any temperature derivatives. We simply let $\mathbf{A}$ pick up the K value-composition derivatives and let the temperature derivatives be approximated by the pure component fugacity-temperature derivatives. This, we feel, is usually the better implementation.

Actually, the analysis is somewhat more complicated than this because the approximated part generated by either version of the hybrid method will not converge to the true approximated part of the Jacobian matrix, say **A***. This is due to the fact that the null space of the corrections to the approximated part always contains the vector v = $1/\sqrt{n_c}\ (1,1,\ldots,1,0)^T$ as a consequence of the summation equation. Let M be the one-dimensional manifold generated by the vector v, which here is of length $n_c + 1$. Because v is always in the null space of the corrections to the approximated part, it follows that initial derivative information in A restricted to M remains unchanged throughout the calculations. The only derivative information that is gathered is that in the orthogonal complement of M, say $M^{\perp}$.

Since the sequence $\{A_k\}$ generated by the modified Schubert update does not converge to A*, it is therefore difficult to determine whether or not A is accumulating good derivative information by simply viewing the terms in the sequence and comparing them to the true approximated part. Instead, because of the remarks concerning the natural null space of the corrections to A, we must compare the projections of A and A* onto $M^{\perp}$.

An alternate procedure to determine whether or not A is accumulating good derivative information is to compute the corresponding Newton step, s^N, at each iterate of the solution trajectory for the hybrid method and compare it with the hybrid step s. If we are getting good derivative information, then the components of s should converge to those of s^N in both magnitude and direction. The results of such a comparison for the temperature and acetonitrile composition components of the Newton and hybrid steps are plotted in Figure 1. Note that for the hybrid method with ∂T we see a good tracking of the temperature component of the Newton step on iterations 4 through 6. However, up to this point we also observe a masking of the composition derivative information that is characterized by the erratic tracking of the acetonitrile component of the Newton step. The other compositions behave in a similar manner. Good composition derivatives have not been gathered because the steps in temperature have been either markedly or slightly dominant. At this point, the temperature distortion begins to take effect, the change in temperature starts to drift away from that of Newton's method, and we continue to see a rather erratic tracking of composition until iteration 11. In fact, on iteration 9 the hybrid and Newton steps in temperature are in opposite directions. Interestingly enough the step in temperature remains small from iteration 11 until convergence is reached. However, it remains comparable in magnitude to the steps in composition and, as a result, the distortion of the temperature derivatives never completely disappears. Thus, this first implementation of the hybrid method takes another nine iterations to reach the desired accuracy.

In contrast, for the implementation of the hybrid method without ∂T in A, we observe good tracking of the temperature component of the Newton step after three iterations and a good tracking of the composition after five iterations, and, of course, no instability.

Finally, we remark that we have solved many types of small problems (i.e., dew-point, single-stage, and multistage-linked flash problems) involving several different multicomponent systems and have observed exactly the same behavior that we are reporting here. Thus, we believe that we are correct in identifying the scaling of the variables as an important factor in the behavior of the hybrid method on design problems. However, in some ways our analysis generates more questions than it answers. For example, what does the sequence $\{A_k\}$ converge to? Does our analysis carry back to Schubert's original method, and does it generalize to other variable interactions (e.g., flow rates and compositions)? Will automatic scaling of the variables eliminate the bad behavior that we report here? And are the same effects of scaling present in large problems? These are questions that we are in the process of answering.

ACKNOWLEDGMENT

This material is based on work supported by the National Science Foundation under Grant No. CPE-8216857.

NOTATION

A	= approximated part of the Jacobian matrix
B	= approximation to the Jacobian
C	= computed part of the Jacobian matrix
e_i	= ith unit vector
f,f(x)	= vector of function values, evaluated at x
f_i^0	= standard state fugacity of the ith component
J,J(x)	= Jacobian matrix of f, evaluated at x
K	= equilibrium or distribution ratio
M	= null space associated with the summation equation
n,n_c	= dimension of problem, number of components in multicomponent mixture
s,s_i,s^N	= iterative change in the unknown variables, with sparsity pattern matching the ith row of approximated part of Jacobian, Newton step
T	= temperature
v	= vector defining the null space of the quasi-Newton corrections to the approximated part
x,x_i	= vector of unknown variables, composition of the ith component in the liquid phase
y,y_i	= change in function values from one iteration to the next, ith component vapor mole fraction

Greek Letters

γ	= activity coefficient
ϕ	= fugacity coefficient

Subscripts

i	= component index
k	= iteration counter

Superscripts

T	= transpose
+	= generalized inverse
0	= standard state
$\perp$	= orthogonal complement

LITERATURE CITED

Lucia, A., and S. Macchietto, "New Approach to Approximation of Quantities Involving Physical Properties Derivatives in Equation-Oriented Process Design," *AIChE J.*, **29**, 705 (1983).

Lucia, A., and K. R. Westman, "Low Cost Solutions to Multistage, Multicomponent Separation Problems by a Hybrid Fixed Point Algorithm," *Foundations of Computer-Aided Process Design*, A. W. Westerberg and H. H. Chen, Eds., 741 (1984).

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

Taylor, R., A. Lucia, and R. Krishnamurthy, "A Newton-like Algorithm for the Efficient Estimation of Rates of Multicomponent Condensation by a Film Model," *I. Chem. E. Symposium Series No. 75*, 380 (1983).

Westman, K. R., A. Lucia, and D. C. Miller, "Flash and Distillation Calculations by a Newton-like Method," accepted for publication in *Comput. Chem. Eng.* (1984).

Manuscript received July 26, 1983; revision received Nov. 8, and accepted Nov. 11.