Iterative Techniques in Optimization:

II. The Linearity Difficulty in Dynamic Programming and Quasilinearization

COLIN McDERMOTT, E. S. LEE, and L. E. ERICKSON

Kansas State University, Manhattan, Kansas

The quasilinearization technique is used to overcome the dimensionality difficulty in dynamic programming. The advantage of this approach is that not only the dimensionality or the fast memory requirement can be reduced, but the computation time required can also be reduced considerably. A technique is also devised to overcome the linearity difficulty encountered in this reduction in dimensionality. To illustrate the approach, the three dimensional crosscurrent extraction problem is solved as a one- and a two-dimensional problem in the dynamic programming algorithm. Only approximately 1 min. is required to solve this one-dimensional problem. However, if this problem were solved as a three-dimensional problem, computation time of the order of hours would be required.

In an earlier paper of this series (*1*), the quasilinearization technique has been used to reduce the dimensionality difficulty in dynamic programming. In this paper, certain computational difficulties are discussed. The most severe difficulty in using the proposed technique to reduce the dimensionality difficulty is the linearity difficulty. This is caused by the fact that when we linearize the nonlinear equation, only the first derivatives are considered. This linearity difficulty is especially severe when we wish to reduce a general multidimensional optimization problem to a one-dimensional problem in the dynamic programming algorithm. This is because all the equations including both the transformation or process equations and the objective function must be linearized in order to reduce to a one-dimensional problem. Since the problem now becomes completely linear, no internal optimum can be obtained.

In this paper, a technique is devised to overcome this linearity difficulty. To illustrate the approach, the three-dimensional crosscurrent extraction problem with miscible solvent is solved as a one- and a two-dimensional problem. The advantage of this technique is that both the dimensionality problem and the required computation time are reduced considerably.

THE PROBLEM

Consider a system represented by the following nonlinear difference equations

$$f_i(v_1(n-1), \ldots, v_m(n-1);\ v_1(n), \ldots, v_m(n);\ w_1(n), \ldots w_m(n)) = 0 \quad (1)$$

$$i = 1, 2, \ldots, m; \quad n = 1, 2, \ldots, N$$

with initial conditions

$$v_i(0) = v_i^0, \quad i = 1, 2, \ldots, m \quad (2)$$

The problem is maximization of the nonlinear objective function

$$H[v_1(N), v_2(N), \ldots, v_M(N)] \quad (3)$$

with $M < m$. In addition, the problem must satisfy the inequality constraints

$$w_{i,\min} \leq w_i(n) \leq w_{i,\max}, \quad i = 1, 2, \ldots, m \quad (4)$$

Notice that the above problem is essentially the same optimization problem as that discussed in an earlier paper (*1*), except that the nonlinear inequality constraint

$$\sum_{n=1}^{N} f[\mathbf{w}(n)] \leq L \quad (5)$$

has been omitted from the above problem. Although the presence of the nonlinear inequality constraint, Equation (5), complicates the problem owing to the necessity of introducing an additional parameter, the Lagrange multiplier, the presence of this inequality constraint reduces the linearity difficulties to be discussed later. Thus, the computational procedure discussed in this paper can be generalized easily to include the inequality constraint. However, linearity difficulties may make the computational procedure discussed in the earlier paper (*1*) useless if the optimization problem does not have the nonlinear inequality constraint, Equation (5).

As has been discussed in the earlier paper (*1*), the above m-dimensional problem can be reduced to an M-dimensional problem by the use of quasilinearization. Equation (1) can be linearized by Taylor series expansion. The linearized equation can be represented by (*1*)

$$\mathbf{v}_{k+1}(n) = \mathbf{A}(n)\mathbf{v}_{k+1}(n-1) + \mathbf{p}(n) \quad (6)$$

where

$$\mathbf{A}(n) = -[\mathbf{J}_{\mathbf{v}(n)}]^{-1}\,\mathbf{J}_{\mathbf{v}(n-1)} \quad (7)$$

and

$$\mathbf{p}(n) = [\mathbf{J}_{\mathbf{v}(n)}]^{-1}\,[\mathbf{J}_{\mathbf{v}(n-1)}\mathbf{v}_k(n-1) + \mathbf{J}_{\mathbf{v}(n)}\,\mathbf{v}_k(n) - \mathbf{J}_{\mathbf{w}}\{\mathbf{w}_{k+1}(n) - \mathbf{w}_k(n)\} - \mathbf{f}(\mathbf{v}_k(n-1),\ \mathbf{v}_k(n),\ \mathbf{w}_k(n))] \quad (8)$$

where $\mathbf{A}(n)$ is an mxm matrix and $\mathbf{p}(n)$ in an m-dimensional vector. Notice that the matrix $\mathbf{A}$ is independent of the control variable $\mathbf{w}_{k+1}$. However, the vector $\mathbf{p}$ is a function of the control $\mathbf{w}_{k+1}$.

With a given set of values for $\mathbf{w}_{k+1}$, the analytical solution of the linear equation, Equation (6), is well known. At $n = N$, this solution can be represented by

$$\mathbf{v}_{k+1}(N) = \mathbf{c} + \sum_{s=0}^{N-1} \mathbf{K}(N-s)\ \mathbf{p}(N-s) \quad (9)$$

where

$$\mathbf{c} = \prod_{s=0}^{N-1} \mathbf{A}(N-s)\ \mathbf{v}_{k+1}(0) \quad (10)$$

and

$$\mathbf{K}(N-s) = \prod_{l=0}^{N-1} \mathbf{A}(N-l) \left[\prod_{j=s}^{N-1} \mathbf{A}(N-j) \right]^{-1} \quad (11)$$

Colin McDermott is at the University of Birmingham, Birmingham, England.

Since $\mathbf{c}$ is independent of $\mathbf{w}_{k+1}$ and only $\mathbf{p}$ is a function of $\mathbf{w}_{k+1}$, the constants $\mathbf{c}$ can be considered as the new initial conditions of the problem replacing the original given initial conditions v_i^0, $i = 1, 2, \ldots, m$. Thus, the dimensionality of this problem has been reduced from m to M.

As has been discussed in the first paper (*1*), the dimensionality of the above problem can be further reduced to 1 by the introduction of an additional state variable.

LINEARITY DIFFICULTY

It is obvious that the full reduction of the dimensionality to 1 will not work owing to the linearity difficulty. Since we linearized both the objective function and the transformation equations, the optimization problem becomes completely linear. A linear optimization problem has no optimum if there is no constraint on the control variable. For the problem represented by Equations (1) through (4), the optimal values of w_i would always be against the constraints, Equation (4), if Equations (1) and (3) were linear equations. Since we wish the iterative linearization scheme to converge to the solution of the original nonlinear problem, and also since we know that the control variables will always be against the constraints after linearization, the scheme will never converge to the desired solution.

It should be noted that the case of full reduction in dimensionality is not the only case to have the linearity difficulty; the case of partial reduction in dimensionality also has linearity difficulty. This is especially true if the objective function is nearly linear and the transformation equations are highly nonlinear. In order to partially avoid this linearity difficulty, the nonlinear inequality constraint, Equation (5), was used in the original problem in the previous paper (*1*).

REFORMULATION OF THE PROBLEM

In order to overcome the linearity difficulty, let us examine the linearization scheme more carefully. As has been discussed in the previous paper (*1*), Equation (6) would still be a linear equation even if the control variables appear nonlinearly in the term $\mathbf{p}(n)$. However, the control variable $\mathbf{w}_{k+1}$ must not appear in the matrix $\mathbf{A}(n)$. This is because we wish to make $\mathbf{c}$ in Equation (10) independent of $\mathbf{w}_{k+1}$. In other words, $\mathbf{c}$ must be a set of known constants before the start of the current $(k + 1)$st iteration. Thus, in the linearization of Equation (1) terms involving only the control variables do not need to be linearized.

The above discussion suggests that if we could recover the nonlinearity of the original equation by the use of a nonlinear function of the control variables and add this function to the objective function obtained from the linearized equations, the linearity difficulty may be overcome.

TABLE 1. CONVERGENCE RATE WITH PARTIAL REDUCTION IN DIMENSIONALITY

Iteration	$w(1)$	$w(2)$	$w(3)$	ϕ
0	0.5	0.5	0.5	0.07657
1	0.275	0.278	0.279	0.05247
2	0.162	0.181	0.187	0.05247
3	0.275	0.278	0.279	0.05247
4	0.223	0.229	0.233	0.05108
5	0.241	0.228	0.239	0.05104
6	0.236	0.220	0.234	0.05103
7	0.241	0.220	0.237	0.05102

Let this nonlinear function be

$$\sum_{i=1}^{N} R[\mathbf{w}_{k+1}(n)] \tag{12}$$

The functional equations of dynamic programming for the partial reduction in dimensionality case become

$$g_N(c_1, c_2, \ldots, c_M) = \max_{\mathbf{w}(1)} \left[g_{N-1} \left[c_1 + \sum_{j=1}^{m} k_{1j}(1) p_j(1), \ldots, c_M + \sum_{j=1}^{m} k_{Mj}(1) p_j(1) \right] + R(\mathbf{w}_{k+1}(1)) \right] \quad N = 2, 3, \ldots, N \tag{13}$$

$$g_1(c_1, c_2, \ldots, c_M) = \max_{\mathbf{w}(1)} \left[H \left[c_1 \sum_{j=1}^{m} k_{1j} p_j(1), \ldots, c_M + \sum_{j=1}^{m} k_{Mj}(1) p_j(0) \right] + R(\mathbf{w}_{k+1}(1)) \right] \tag{14}$$

This nonlinear function R must possess two properties. The value of R must approach zero as the iterative calculation approaches the solution of the original nonlinear problem so that at the optimum, Equations (13) and (14) reduce to the original functional equations. The second property is that it must have second derivatives of the proper sign and sufficient magnitude to recover some of the nonlinearity property of the original problem so that the iterative scheme converges to the desired optimum of the original nonlinear problem. A suitable function for R is

$$\sum_{i=1}^{N} R[\mathbf{w}_{k+1}(n)] = S \sum_{i=1}^{N} [\mathbf{w}_k(n) - \mathbf{w}_{k+1}(n)]^2 \tag{15}$$

where S is a positive parameter in a minimization problem and a negative parameter in a maximization problem. The magnitude of S will affect the rate of approach to the optimum and should, of course, be chosen so that the rate of approach to the optimum is as rapid as possible. Notice that the function R as defined by Equation (15) approaches zero as the results of the $(k + 1)$st iteration approach the results of the kth iteration.

In order for the procedure to converge, a proper value of S must be used. In actual calculations, it was found convenient to use different values of S for different iterations. The actual value of S used in each iteration may be obtained in the following manner. For the first iteration, an arbitrary value of $S = 1$ is assumed. This value of S is used as long as the profit in the current iteration is the same or better than that obtained in the previous iteration. However, the value of S is doubled in the next iteration if we find that the profit in the current iteration is worse than the profit in the previous iteration.

CROSSCURRENT EXTRACTION

To illustrate the technique, the optimum of a crosscurrent extraction problem will be obtained. The state variable equations are (*1*)

$$q(n-1) + w(n) - q(n) - u(n) = 0 \tag{16}$$

$$q(n-1)x_R(n-1) - q(n)x_R(n) - u(n)x_E(n) = 0 \tag{17}$$

$$q(n-1)y_R(n-1) - q(n)y_R(n) - u(n)y_E(n) = 0 \tag{18}$$

A mixture containing A and C is to be extracted by the solvent B. x and y represent the concentrations of C and A, respectively. The raffinate and extract streams are represented by symbols with subscripts R and E, respectively. The variables y_R, y_E, and x_E in the above equations can be eliminated by using the equilibrium relationships

$$y_R = a + b\ x_R + cx_R^2 \tag{19}$$

$$y_E = d + ex_R + fx_R^2 \tag{20}$$

$$x_E = g + hx_R + ix_R^2 + jx_R^3 \tag{21}$$

The equilibrium relationships used in a previous paper (4) are used in the present calculations. The quantity to be maximized is

$$\psi = \sum_{n=1}^{N} [u(n)x_E(n) - \lambda w(n)] \tag{22}$$

The control variable is $w(n)$, and x_R, q, and u are state variables. In order to convert the objective function into

PARTIAL REDUCTION IN DIMENSIONALITY

The above optimization problem is a three-dimensional problem involving three state variables, or $m = 3$. Since only two state variables q and x_R appear in the objective function, Equation (24), or $M = 2$, the problem can be solved as a two-dimensional problem if partial reduction in dimensionality is used. If we define the vector

$$\mathbf{v}(n) = \begin{bmatrix} q(n) \\ u(n) \\ x_R(n) \end{bmatrix} \tag{25}$$

Equations (16) to (18) can be put into the form of Equation (1) with $m = 3$. After linearization, Equations (6) to (8) can be obtained with the following matrices:

$$\mathbf{J}_{\mathbf{v}(n+1)} = \begin{bmatrix} 1 & 0 & 0 \\ x_{R,k}(n-1) & 0 & q_k(n-1) \\ y_{R,k}(n-1) & 0 & q_k(n-1)\dfrac{\partial y_{R,k}(n-1)}{\partial x_{R,k}(n-1)} \end{bmatrix} \tag{26}$$

$$\mathbf{J}_{\mathbf{v}(n)} = \begin{bmatrix} -1 & -1 & 0 \\ -x_{R,k}(n) & -x_{E,k}(n) & -q_k(n) - u_k(n)\dfrac{\partial x_{E,k}(n)}{\partial x_{R,k}(n)} \\ -y_{R,k}(n) & -y_{E,k}(n) & -q_k(n)\dfrac{\partial y_{R,k}(n)}{\partial x_{R,k}(n)} - u_k(n)\dfrac{\partial y_{E,k}(n)}{\partial x_{E,k}(n)} \end{bmatrix} \tag{27}$$

$$\mathbf{p}(n) = [\mathbf{J}_{\mathbf{v}(n)}]^{-1} \begin{bmatrix} -w_{k+1}(n) \\ x_{R,k}(n-1)q_k(n-1) \\ \qquad - x_{R,k}(n)\left[q_k(n) + u_k(n)\dfrac{\partial x_{E,k}(n)}{\partial x_{R,k}(n)} \right] \\ x_{R,k}(n-1)q_k(n-1)\dfrac{\partial y_{R,k}(n-1)}{\partial x_{R,k}(n-1)} \\ \qquad -x_{R,k}(n)\left[q_k(n)\dfrac{\partial y_{R,k}(n)}{\partial x_{R,k}(n)} \right. \\ \qquad\qquad \left. + u_k(n)\dfrac{\partial y_{E,k}(n)}{\partial x_{R,k}(n)} \right] \end{bmatrix} \tag{28}$$

one involving only the output values of the state variables for the final stage, we observe that the maximum value of ψ corresponds to the minimum value of ϕ defined by

$$\begin{aligned} \phi &= q(0)\ x_R(0) - \psi \\ &= q(0)\ x_R(0) - \sum_{n=1}^{N} [u(n)x_E(n) - \lambda w(n)] \end{aligned} \tag{23}$$

The quantity

$$q(0)\ x_R(0) - \sum_{n=1}^{N} u(n)\ x_E(n)$$

is the amount of component C entering the system which does not appear in the extract streams. This is, of course, the amount of C leaving stage N as raffinate. Thus, we may write

$$\phi = q(N)\ x_R(N) + \sum_{n=1}^{N} \lambda w(n) \tag{24}$$

The problem now becomes the minimization of ϕ subject to the constraints of Equations (16) to (21).

The partial derivatives in the above equations can be obtained by differentiating Equations (19) to (21) with respect to x_R. With the matrices **A** and **p** known, the problem becomes the minimization of the objective function

$$\begin{aligned} \phi &= q(N)\ x_R(N) + \sum_{n=1}^{N} \lambda W_{k+1}(n) \\ &= \left[c_1 + \sum_{s=0}^{N-1} \left\{ \sum_{j=1}^{3} k_{1j}(N-s)p_j(N-s) \right\} \right] \\ &\quad \left[c_3 + \sum_{s=0}^{N-1} \left\{ \sum_{j=1}^{3} k_{3j}(N-s)p_j(N-s) \right\} \right] \\ &\qquad + \sum_{n=1}^{N} \lambda w_{k+1}(n) \\ &= H\left\{ c_1 + \sum_{s=0}^{N-1} \left[\sum_{j=1}^{3} k_{1j}(N-s)p_j(N-s) \right] \right. \end{aligned}$$

$$c_3 + \sum_{s=0}^{N-1} \left[\sum_{j=1}^{3} k_{3j}(N-s) p_j(N-s) \right] \Bigg\} \Bigg\}$$

$$+ \sum_{n=1}^{N} \lambda w_{k+1}(n) \qquad (29)$$

with the initial states c_1 and c_3. Thus, we have reduced the original three-dimensional problem into a two-dimensional problem. The functional equation of dynamic programming is

$$g_N(c_1, c_3) = \max_{w(1)} \Bigg[g_{N-1} \Bigg\{ c_1 + \sum_{j=1}^{3} k_{1j}(1) p_j(1),$$

$$c_3 + \sum_{j=1}^{3} k_{3j}(1) p_j(1) \Bigg\} + \lambda w_{k+1}(1)$$

$$+ s(w_k(1) - w_{k+1}(1))^2 \Bigg]$$

$$N = 2, 3, \ldots, N \qquad (30)$$

If the process only had one stage, we would obtain

$$g_1(c_1, c_3) = \max_{w(1)} \Bigg[H \Bigg\{ c_1 + \sum_{j=1}^{3} k_{1j}(1) p_j(1),$$

$$c_3 + \sum_{j=1}^{k} k_{3j}(1) p_j(1) \Bigg\}$$

$$+ \lambda w_{k+1}(1) + S(w_k(1) - w_{k+1}(1))^2 \Bigg] \qquad (31)$$

This minimization problem was solved by the following computational procedure:

1. Choose a set of initial approximations of the decision variables $w(1), w(2), \ldots, w(N)$.
2. Evaluate the vectors $\mathbf{v}(n)$ for Equations (16) to (21) for all stages of the system and use these values to calculate the matrix $\mathbf{A}(n)$ and the vector $\mathbf{p}(n)$ as defined by Equations (7), (8), (26), (27), and (28).
3. Obtain an improved set of decision variables, $w(1)$, $w(2), \ldots, w(N)$ using the functional equations, Equations (30) and (31).
4. Repeat steps 2 and 3 until the required accuracy is obtained.

If ϵ is the maximum error allowed, the required accuracy can be defined by

$$|w_{k+1}(n) - w_k(n)| \leqq \epsilon, \quad n = 1, 2, \ldots, N \qquad (32)$$

Since x_R cannot be solved explicitly from Equations (16) to (18), the Newton-Raphson iteration scheme is used to solve these equations in step 2.

Dynamic programming is used to obtain the improved set of decision variables in step 3. To solve the function equations, Equations (30) and (31), the feasible ranges of the values of c_1 and c_3 must be known. Since these c variables are newly introduced ones, and since they do not have any particular physical significance, the probable ranges of the values of these variables are often unknown. However, approximate values of c can be obtained by solving Equations (16) to (21) and (26) to (28) by using the values of the control variables given in step 1.

TABLE 2. CONVERGENCE RATE WITH FULL REDUCTION IN DIMENSIONALITY, $N = 3$

Iteration	$w(1)$	$w(2)$	$w(3)$	ϕ
0	0.2	0.2	0.2	0.05215
1	0.217	0.209	0.212	0.05140
2	0.225	0.214	0.219	0.05118
3	0.230	0.217	0.223	0.05109
4	0.233	0.218	0.226	0.05106
5	0.235	0.220	0.228	0.05104
10	0.239	0.220	0.233	0.05102

A simple enumeration search was used to locate the optimum in the functional equations of dynamic programming, Equations (30) and (31). Since the problem is still a two-dimensional problem, a Legendre polynomial approximation was used for convenience to represent the results of the previous stage. This method is described fully elsewhere (5).

The numerical values used for this problem are

$q(0) = 1$	$a = 0.9865$	$f = 2.6043$
$x_R(0) = 0.4$	$b = -0.8295$	$g = 0.0150$
$y_R(0) = 0.6$	$c = -1.2203$	$h = 4.1882$
$\lambda = 0.05$	$d = 0.009$	$i = -11.23$
	$e = -0.1156$	$j = 8.481$

(33)

This problem was solved on an IBM-360/50 computer. Part of the results for $N = 3$ are shown in Table 1. The initial approximation used in step 1 is $w(n) = 0.5$, $n = 1, 2, 3$. The optimum values of the state variables obtained are

$q(1) = 0.724$	$u(1) = 0.517$	$x_R(1) = 0.211$
$q(2) = 0.597$	$u(2) = 0.347$	$x_R(2) = 0.085$
$q(3) = 0.561$	$u(3) = 0.273$	$x_R(3) = 0.029$

A value of $\psi = 0.3490$ was obtained by Lee (4). This corresponds to a value of ϕ in our case of 0.0510. Thus, the optimal results are in close agreement with the results obtained by Lee. The functions ψ and ϕ are defined by Equations (22) and (24), respectively.

Only approximately 3 min. of computation time were needed to obtain the seven iterations shown in Table 1. If this problem were solved by the standard dynamic programming algorithm without linearization, computation time in the order of hours would be needed to solve a three-dimensional problem.

FULL REDUCTION IN DIMENSIONALITY

The dimensionality of the crosscurrent extraction prob-

TABLE 3. CONVERGENCE RATE WITH FULL REDUCTION IN DIMENSIONALITY, $N = 10$

Iteration	$w(1)$	$w(4)$	$w(7)$	$w(10)$	ϕ
0	0.2	0.2	0.2	0.2	0.09817
1	0.175	0.175	0.175	0.175	0.08576
2	0.175	0.175	0.175	0.175	0.08576
3	0.151	0.150	0.150	0.175	0.07342
4	0.127	0.125	0.125	0.125	0.06132
5	0.126	0.125	0.125	0.125	0.06132
10	0.200	0.091	0.092	0.092	0.04917
15	0.136	0.049	0.053	0.053	0.03567
20	0.138	0.048	0.055	0.056	0.03565

TABLE 4. OPTIMUM VALUES OF X_R, $N = 10$

n	X_R
1	0.324
2	0.276
3	0.216
4	0.159
5	0.111
6	0.076
7	0.051
8	0.034
9	0.022
10	0.014

lem can be further reduced to 1 by using the procedure for full reduction in dimensionality. Let us define the additional state variable

$$v_4(n) = q(n)x_R(n) \tag{34}$$

The corresponding transformation equation is

$$v_4(n) - v_4(n-1) + q(n-1)\, x_R(n-1) - q(n)\, x_R(n) = 0 \tag{35}$$

The problem now is simply the minimization of the final value of $v_4(N)$. The initial condition for Equation (35) is

$$v_4(0) = q(0)\, x_R(0) \tag{36}$$

The transformation or state variable equations are Equations (16) to (18) and (35). Thus, we now have four state variable equations. The linearized state variable equations can still be represented by Equation (6). The state vector now becomes

$$\mathbf{v}(n) = \begin{bmatrix} q(n) \\ u(n) \\ x_R(n) \\ v_4(n) \end{bmatrix} \tag{37}$$

The matrices in Equations (6) to (8) become four-dimensional matrices which can be obtained by adding the fourth row and column to the matrices defined by Equations (26) to (28). These matrices can be obtained by linearizing Equations (16) to (18) and (35):

$$\mathbf{J}_{\mathbf{v}(n-1)} = \left[\begin{array}{ccc|c} & & & 0 \\ & \mathbf{B}_1 & & 0 \\ & & & 0 \\ \hline x_{R,k}(n-1) & 0 & q_k(n-1) & -1 \end{array}\right] \tag{38}$$

$$\mathbf{J}_{\mathbf{v}(n)} = \left[\begin{array}{ccc|c} & & & 0 \\ & \mathbf{B}_2 & & 0 \\ & & & 0 \\ \hline -x_{R,k}(n) & 0 & -q_k(n) & 1 \end{array}\right] \tag{39}$$

$$\mathbf{p}(n) = [\mathbf{J}_{\mathbf{v}(n)}]^{-1} \left[\begin{array}{c} \mathbf{b} \\ x_{R,k}(n-1)\ q_k(n-1) - x_{R,k}(n)\ q_k(n) \end{array}\right] \tag{40}$$

where the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ are 3×3 matrices and are the matrices defined by Equations (26) and (27), respectively. The vector $\mathbf{b}$ is the three-dimensional vector on the right-hand side of Equation (28).

By using the full reduction scheme, this problem can be solved as a one-dimensional dynamic programming problem. The computational procedure is essentially the same as the previous case, except that the optimum return function q_N is a function of only one variable, or $q_N(c_4)$, and the matrices are now given by Equations (38) to (40).

This problem was solved by using the numerical values listed in Equation (33). The convergence rates are shown in Tables 2 and 3 for $N = 3$ and $N = 10$, respectively. The initial approximation used is $w(n) = 0.2$, $n = 1, 2, \ldots, 10$. The optimum values obtained for $N = 3$ agree well with the results obtained in the previous section. Only 1 min. of computation time is needed to obtain the ten iterations listed in Table 2. The problem with the initial approximation $w(n) = 0.5$, $n = 1, 2, 3$, and $N = 3$ is also solved. It takes about the same number of iterations as shown in Table 2 to obtain the optimum.

For $N = 10$, the optimal values obtained for x_R are listed in Table 4. The optimal results are in close agreement with the results obtained by Lee (*4*). It takes approximately 4 min. computation time to obtain the results listed in Table 3.

It should be noted that the present scheme not only reduces the computer memory requirement but also significantly reduces the computation time as compared with the straightforward dynamic programming algorithm.

CONCLUSION

The use of quasilinearization combined with a scheme for overcoming the linearity difficulty is shown to be a powerful tool for overcoming the dimensionality difficulty in dynamic programming. One of the important advantages of this technique is that both the computer memory requirement and the required computation time are reduced considerably. In the full reduction in dimensionality of the crosscurrent extraction problem, only approximately 1 min. of computation time is needed to solve the problem. If this same problem were solved by the conventional dynamic programming algorithm as a three-dimensional problem, computation time of the order of hours would be needed.

A detailed discussion of the advantages and disadvantages of this approach has been given in a previous paper (*1*). The main disadvantage of this approach is that constraints on the state variables cannot be handled in a straightforward manner. This is due to the fact that the dummy variable c is not the original state variable.

ACKNOWLEDGMENT

This work was supported by the Office of Water Resources Research, U.S. Department of the Interior Grant No. 14-01-0001-1962. We wish to thank Syed Waziruddin for his help in obtaining the numerical results.

NOTATION

A = matrix defined by Equation (7)
c = solution of the homogeneous difference equation, the imbedding variable
g = optimal return function
H = criterion function
J = Jacobian matrix
K = matrix corresponds to the Green's function

M	= number of state variables in the objective function
m	= number of state variables
N	= last stage or the total number of stages
n	= n^{th} stage
$\mathbf{p}$	= vector defined by Equation (8)
q	= flow rate of raffinate stream
u	= outlet flow rate of extract stream
v	= state variable
w	= control variable or feed rate of extract solvent
x	= concentration of component C
y	= concentration of component A
λ	= Lagrange multiplier

Subscripts

E	= extract stream
k	= k^{th} iteration, assumed known
$k+1$	= $(k+1)$st iteration, assumed unknown
R	= raffinate stream

LITERATURE CITED

1. Lee, E. S., *AIChE J.*, 14, No. 6, 908 (Nov., 1968).
2. ———, "Quasilinearization and Invariant Imbedding," Academic Press, New York (1968).
3. Bellman, R., "Introduction to Matrix Analysis," McGraw-Hill, New York (1960).
4. Lee, E. S., *Ind. Eng. Chem. Fundamentals*, 3, 373 (1964).
5. Bellman, R., and S. Dreyfus, "Applied Dynamic Programming," Princeton Univ. Press, N. J. (1962).

Manuscript received November 18, 1968; revision received January 20, 1969; paper accepted January 22, 1969. Paper presented at AIChE New Orleans meeting.

Analysis of Kinetic Parameters from Batch and Integral Reaction Experiments

JOHN H. SEINFELD and GEORGE R. GAVALAS

California Institute of Technology, Pasadena, California

The estimation of chemical reaction rate constants from experimental batch or integral reactor data is examined. Consideration is given to the effect on the accuracy of the parameter estimates of the number of concentrations measured, extending the range of measurements to higher conversions, and experimental errors. The study is based on estimates of two rate constants obtained by quasilinearization in the pyrolytic dehydrogenation of benzene. It is found that if fewer than the number of independent components are to be measured, certain components produce better parameter estimates than others. The level of experimental errors is by far the most significant factor in obtaining reliable parameter estimates.

Several recent studies have been concerned with the nonlinear least-squares analysis of reaction rate data (*3, 7, 8, 9*). In each of these methods the sum of squares of the reaction rates r are minimized. If differential reactor data are available, this represents an acceptable method of obtaining rate constants. Often, however, considerable problems arise in the experimental determination of differential reaction rates due to the small concentration changes involved. If the study is conducted in a batch or integral flow reactor, the experimental data are in the form of concentration vs. time (or reciprocal space velocity) measurements. The reaction rates r must then be obtained by numerical differentiation of the experimental data, a highly inaccurate procedure.

The direct utilization of integral reactor data is a problem of estimation of parameters in ordinary differential equations. Several general techniques have been used for the solution of this problem (*2, 11*), and one of them, quasilinearization, has been shown to be particularly attractive because of its quadratic convergence characteristics (*2*). Bellman et al. (*1*) used quasilinearization to estimate the forward and reverse rate constants in the oxidation of nitric oxide from the classic data of Bodenstein and Lindner. Lee (*10*) estimated the Peclet number in a tubular reactor with axial mixing by means of quasilinearization. Donnelly and Quon (*4*) used quasilinearization to estimate rate constants for a series of monomolecular first-order reactions and the nonisothermal pyrolysis of propane.

An important part of the estimation problem is the assignment of confidence intervals or other measures of the reliability or accuracy of the estimated kinetic constants. An appropriate procedure for obtaining confidence intervals for parameter estimates in differential equations is presented by Rosenbrock and Storey (*11*), and a thorough example of the use of such an analysis is given by Heineken et al. (*5*).

The present note represents the results of a detailed numerical analysis with quasilinearization used on the estimation of rate constants for the pyrolytic dehydrogenation of benzene [two independent reactions in a tubular flow reactor for which experimental data are available in the literature (*6*)]. The following are studied:

1. The efficiency and range of convergence of the quasilinearization technique.

2. The number of components to be measured. If the number of reactions is R, there are R independent components. However, it may not always be necessary to measure all R concentrations, and, if not, it is important to determine which ones will yield the best estimates.

3. The effect of extending the range of measurements to higher conversions. To what extent should a particular reaction be carried so that estimates derived from the experimental observations are most meaningful.

4. The effect of the level of experimental errors on the accuracy of the estimates.

Our approach to 2 to 4 is analogous to that of (*5*).