The Control of Nonlinear Systems:

Part IV. Quasilinearization as a Numerical Method

B. F. ROTHENBERGER and LEON LAPIDUS

Princeton University, Princeton, New Jersey

Five different variations on the basic quasilinearization algorithm for optional control are applied to a six-variable nonlinear gas absorber system. Convergence of the algorithms to the optimal trajectory with better than 1% accuracy is achieved in less than ten iterations. The region of initial starting points which yield convergence is quite large. The main difficulty encountered is due to instabilities of the maximum principle equations. This limitation is, however, not severe and it is possible to predict a priori when the difficulty will occur. It is concluded that quasilinearization is a feasible algorithm for the optimization of nonlinear unconstrained systems.

The application of optimal control theory to systems characteristic of the chemical process industry is hampered by the multidimensional, nonlinear nature of these systems. The maximum principle formulation of such optimization problems leads to a nonlinear, two-point, boundary-value problem which is difficult to solve numerically. While much progress has been made toward developing an iterative numerical algorithm suitable for practical systems in the aerospace industry, the application of these procedures to problems of interest to chemical engineers has, in general, not been demonstrated. Consideration of the existing numerical methods for dynamic optimization indicates that quasilinearization is potentially one of the most feasible numerical procedures.

Quasilinearization is an iterative method that approximates the original nonlinear, boundary-value problem by a linear one which is easier to solve numerically (*3*). The quasilinear algorithm has two characteristics which make it superior to most other methods. First, convergence is rapid (*7*) and second, the treatment of large systems of equations is facilitated by very meager computer storage requirements (*9*). Quasilinearization has been compared to two popular algorithms, the gradient method and the second variation method (*10*), and proved to be superior for several specific problems (*15*).

However, quasilinearization has been used almost exclusively to treat aerospace systems. Only one chemical engineering problem has been investigated. Lee (*13*) solved a two-variable, highly nonlinear tubular reactor problem, obtaining four-digit accuracy in only four iterations with a poor initial starting point for the algorithm. In addition, the largest problem tested by quasilinearization is a three-variable aerospace problem (*15*). Furthermore, there are many variations on the basic idea of quasilinearization and the algorithms based on these variations have not been compared. With these points in mind, this investigation will be concerned with the following questions.

1. Is quasilinearization capable of handling systems typical of the chemical process industry?
2. Can quasilinearization effectively treat multidimensional chemical engineering systems?
3. Which of the various quasilinear algorithms is best?

In order to shed some light on these questions, the optimization of a six-plate gas absorber (six variables) with two control variables and a nonlinear equilibrium expression is considered. With regard to the first two questions posed above, it is shown that (1) convergence to better than 1% accuracy is obtained in less than ten iterations; and (2) the instability of the maximum principle adjoint equations for this system, while limiting the effectiveness of even the best quasilinear variation, does not prevent convergence for realistic cases.

In addition, five different quasilinearization algorithms are examined and ranked according to their effectiveness. While successful application of quasilinearization to one typical chemical engineering problem does not justify a claim of universal applicability, the results of this investigation do indicate that the quasilinear algorithm is a feasible method for the optimization of nonlinear, unconstrained, chemical process systems. A further paper will illustrate the application to constrained problems.

THE MAXIMUM PRINCIPLE FORMULATION

We shall consider nonlinear, unconstrained systems defined by n ordinary differential equations:

$$\frac{d\mathbf{x}(t)}{dt} \equiv \dot{\mathbf{x}}(t) = \mathbf{F}[\mathbf{x}(t), \mathbf{u}(t)], \quad t_o \leqq t \leqq t_f \tag{1}$$

where $\mathbf{x} = [x_1 \ldots, x_n]$ is the n dimensional state vector, $\mathbf{u} = [u_1, \ldots, u_r]$ is the r dimensional control vector, and t is the independent variable, usually time. It is desired to find $\mathbf{u}(t)$ which minimizes the scalar performance index

$$I(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^{u} c_i x_i(t_f) = \mathbf{c}'\mathbf{x}(t_f) \tag{2}$$

given the initial conditions

$$\mathbf{x}(t_o) = \mathbf{x}^o \tag{3}$$

The final time t_f is assumed known and fixed, the c_i are weighting factors, and the prime indicates the transpose of a vector or matrix.

Employing Pontryagin's maximum principle (*16, 18*), define a Hamiltonian function by

$$H(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \sum_{i=1}^{n} z_i F_i(\mathbf{x}, \mathbf{u}) = \mathbf{z}'\mathbf{F} \tag{4}$$

B. F. Rothenberger is with Sun Oil Company, Marcus Hook, Pennsylvania.

where the adjoint vector of n dimensions, $\mathbf{z} = [z_1, \ldots, z_n]$, is defined by

$$\frac{d\mathbf{z}(t)}{dt} \equiv \dot{\mathbf{z}}(t) = -\frac{\partial H(\mathbf{x}, \mathbf{z}, \mathbf{u})}{\partial \mathbf{x}} \qquad (5)$$

In addition, the adjoint vector is prescribed at $t = t_f$ by

$$\mathbf{z}(t_f) = -\mathbf{c} \qquad (6)$$

The optimal control policy, $\mathbf{u}(t) = \mathbf{u}^*(t)$, is chosen by maximizing H at each instant of time with respect to $\mathbf{u}$. If we require that $\mathbf{F}(\mathbf{x}, \mathbf{u})$ be nonlinear in every component of $\mathbf{u}$, then $\mathbf{u}^*(t)$ is given by the solution to the r algebraic equations

$$\frac{\partial H(\mathbf{x}, \mathbf{z}, \mathbf{u})}{\partial u_i} = 0, \quad i = 1, \ldots, r \qquad (7)$$

Solving Equation (7) for $\mathbf{u}(t)$ as a function of $\mathbf{x}(t)$ and $\mathbf{z}(t)$ and substituting the result into Equations (1) and (5), we obtain the following set of $2n$ ordinary differential equations:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}) \qquad (8)$$

and

$$\dot{\mathbf{z}} = \mathbf{g}(\mathbf{x}, \mathbf{z}) \qquad (9)$$

Equations (8) and (9) are subject to $2n$ boundary conditions, namely, the n initial conditions of Equation (3) and the n final conditions of Equation (6). This, then, is the nonlinear, two-point, boundary-value problem which must be solved to yield the necessary conditions for an optimal control policy.

THE QUASILINEAR ALGORITHM

In its basic form, quasilinearization assumes $\mathbf{x}(t)$ and $\mathbf{z}(t)$. Call these initial assumptions $\mathbf{x}^{(0)}(t)$ and $\mathbf{z}^{(0)}(t)$, $t_o \leq t \leq t_f$, and then iteratively improve these values until the $(m+1)^{\text{th}}$ and m^{th} approximations to both $\mathbf{x}(t)$ and $\mathbf{z}(t)$ differ by less than some prescribed ϵ. $\mathbf{x}^{(m+1)}(t)$ and $\mathbf{z}^{(m+1)}(t)$ are then the optimal trajectories $\mathbf{x}^*(t)$ and $\mathbf{z}^*(t)$. Equation (7) produces $\mathbf{u}^*(t)$. In each iterative cycle the known $2n$ boundary conditions on $\mathbf{x}(t_o)$ and $\mathbf{z}(t_f)$ are used and held fixed. This approach is quite different from those numerical methods which iterate on the missing boundary conditions, $\mathbf{x}(t_f)$ or $\mathbf{z}(t_o)$, or those which iterate on the control policy (*5, 6, 8, 10, 12*). As an illustration, it has been shown (*1*) that iterating on the missing boundary conditions may lead to considerable computational difficulties due to the inherent instability of the adjoint equations to forward integration.

To illustrate the features of the quasilinearization algorithm, we consider the problem described by Equations (8) and (9). Assume that the m^{th} approximation to $\mathbf{x}(t)$ and $\mathbf{z}(t)$, denoted by $\mathbf{x}^{(m)}$ and $\mathbf{z}^{(m)}$, is known. Expressions for the $(m+1)^{\text{th}}$ approximation are obtained by expanding the right-hand side of Equations (8) and (9) in a power series about the m^{th} approximation, keeping only the first two terms. The following equations result:

$$\frac{dx_i^{(m+1)}}{dt} = f_i(\mathbf{x}^{(m)}, \mathbf{z}^{(m)}) + \sum_{j=1}^{n} \left[(x_j^{(m+1)} - x_j^{(m)}) \left.\frac{\partial f_i}{\partial x_j}\right|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} + (z_j^{(m+1)} - z_j^{(m)} \left.\frac{\partial f_i}{\partial z_j}\right|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} \right] \qquad (10)$$

and

$$\frac{dz_i^{(m+1)}}{dt} = g_i(\mathbf{x}^{(m)}, \mathbf{z}^{(m)}) + \sum_{j=1}^{n} \left[(x_j^{(m+1)} - x_j^{(m)}) \left.\frac{\partial g_i}{\partial x_j}\right|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} + (z_j^{(m+1)} - z_j^{(m)} \left.\frac{\partial g_i}{\partial z_j}\right|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} \right] i = 1, \ldots, n. \qquad (11)$$

Equations (10) and (11) are solved until the difference between the m^{th} and the $(m+1)^{\text{th}}$ approximations is arbitrarily small. Kalaba (*7*) proves that convergence is monotonic if f_i and g_i, as well as their first and second partial derivatives, are bounded. In addition, convergence is quadratic if the interval $[t_o, t_f]$ is sufficiently small. This suggests suitability for numerical computation.

Note that Equations (10) and (11) are linear in the $(m+1)^{\text{th}}$ approximation. Linear, boundary-value problems are easily solved by superposition (*3*). As an illustration obtain a particular solution, denoted by $\mathbf{x}_p(t)$ and $\mathbf{z}_p(t)$, of Equations (10) and (11) with arbitrary initial conditions. Generate $2n$ homogeneous solutions, where the k^{th} solution is $\mathbf{x}_h^{(k)}(t)$ and $\mathbf{z}_h^{(k)}(t)$, with initial conditions $\mathbf{x}_h^{(k)}(t_o) = s\boldsymbol{\delta}^{(k)}$, $\mathbf{z}_h^{(k)}(t_o) = \mathbf{0}$ for $k \leq n$ and $\mathbf{x}_h^{(k)}(t_o) = \mathbf{0}$, $\mathbf{z}_h^{(k)}(t_o) = s\boldsymbol{\delta}^{(k-n)}$ for $k > n$, where $s\boldsymbol{\delta}^{(k)}$ is an n dimensional vector with the general element being

$$s\delta_i^{(k)} = \begin{cases} s, & i = k, \quad s \text{ arbitrary} \\ 0, & i \neq k \end{cases} \qquad (12)$$

and $\mathbf{0}$ is an n dimensional null vector. These solutions are combined by the superposition principle so that

$$x_i^{(m+1)}(t) = x_{p_i}(t) + \sum_{j=1}^{2n} \alpha_j^{(m+1)} \, x_{h_i}^{(j)}(t) \qquad (13)$$

and

$$z_i^{(m+1)}(t) = z_{p_i}(t) + \sum_{j=1}^{2n} \alpha_j^{(m+1)} \, z_{h_i}^{(j)}(t) \qquad (14)$$

$\boldsymbol{\alpha}^{(m+1)}$ is chosen so that $\mathbf{x}^{(m+1)}$ and $\mathbf{z}^{(m+1)}$ in Equations (13) and (14) satisfy the $2n$ known boundary conditions on Equations (8) and (9). This requires the solution to $2n$ simultaneous, linear, algebraic equations, for which many techniques are available. For purposes of further discussion this method will be called $M1$.

There are a number of interesting variations on this basic algorithm which are important. Thus McGill and Kenneth (*14*) suggest a somewhat simpler method than the one above. Instead of allowing the initial conditions for the particular solution to be completely arbitrary, let $\mathbf{x}_p(t_o) = \mathbf{x}^o$ with $\mathbf{z}_p(t_o)$ arbitrary. Then form n homogeneous solutions to Equations (10) and (11) where $\mathbf{x}_h^{(k)}(t_o) = \mathbf{0}$ and $\mathbf{z}_h^{(k)}(t_o) = s\boldsymbol{\delta}^{(k)}$. The solutions are combined as before except that the summation in Equations (13) and (14) goes up to n instead of $2n$, since the initial conditions of $\mathbf{x}^{(m+1)}(t_o) = \mathbf{x}^o$ are automatically satisfied for any finite $\boldsymbol{\alpha}^{(m+1)}$. Now only n algebraic equations must be solved and n homogeneous solutions generated and stored. Denote this as method $M2$.

A further modification of the $M1$ algorithm can also be made. Generate the $2n$ homogeneous solutions where the initial conditions on the k^{th} solution are $\mathbf{x}_h^{(k)}(t_o) = \mathbf{x}^{(m)}(t_o) + s\boldsymbol{\delta}^{(k)}$, $\mathbf{z}_h^{(k)}(t_o) = \mathbf{z}^{(m)}(t_o)$ for $k \leq n$ and $\mathbf{x}_h^{(k)}(t_o) = \mathbf{x}^{(m)}(t_o)$, $\mathbf{z}_h^{(k)}(t_o) = \mathbf{z}^{(m)}(t_o) + s\boldsymbol{\delta}^{(k-n)}$ for $k > n$. For the particular solution let $\mathbf{x}_p(t_o) = \mathbf{x}^{(m)}(t_o)$, $\mathbf{z}_p(t_o) = \mathbf{z}^{(m)}(t_o)$ in generating the $(m+1)^{\text{th}}$ approximation. Call this the $M3$ method. It might be expected that this choice of initial conditions will increase the convergence of the superposition solutions. However, if the solutions start to diverge this technique will probably increase the rate of divergence since the m^{th} solution is carried over into the $(m + 1)^{\text{th}}$ approximation in the superposition initial conditions as well as in the right-hand side of the linearized equations.

Sylvester and Meyer (*19*) have proposed yet another modification of quasilinearization. Define the state and adjoint variable error vectors $\mathbf{e}$ and $\boldsymbol{\epsilon}$, by

$$e_i = x_i^{(m+1)} - x_i^{(m)} \tag{15}$$

$$\epsilon_i = z_i^{(m+1)} - z_i^{(m)} \tag{16}$$

where $i = 1, \ldots, n$. Then Equations (10) and (11) become

$$\frac{de_i}{dt} = -\frac{dx_i^{(m)}}{dt} + f_i(\mathbf{x}^{(m)}, \mathbf{z}^{(m)}) + \sum_{j=1}^{n} \left[e_j \frac{\partial f_i}{\partial x_j}\bigg|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} + \epsilon_j \frac{\partial f_i}{\partial z_j}\bigg|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} \right] \tag{17}$$

and

$$\frac{d\,\epsilon_i}{dt} = -\frac{dz_i^{(m)}}{dt} + g_i\,(\mathbf{x}^{(m)}, \mathbf{z}^{(m)}) + \sum_{j=1}^{n} \left[e_j \frac{\partial g_i}{\partial x_j}\bigg|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} + \epsilon_j \frac{\partial g_i}{\partial x_j}\bigg|_{\mathbf{x}^{(m)},\mathbf{z}^{(m)}} \right] \tag{18}$$

$$i = 1, \ldots, n.$$

If the numerical errors involved in solving Equations (10) and (11) and (17) and (18) are the same, then the magnitude of the numerical error in $\mathbf{x}$ and $\mathbf{z}$ using Equations (17) and (18) will be smaller than the error resulting from the use of Equations (10) and (11). This is due to the fact that the error is now associated with $\mathbf{e}$ and $\boldsymbol{\epsilon}$, whose magnitudes are much smaller than $\mathbf{x}$ and $\mathbf{z}$.

These quasilinearization error equations can be solved, as before, by generating homogeneous and particular solutions and forming a linear combination so that the boundary conditions are satisfied. Note that if each successive approximation is required to match the boundary conditions, the boundary values for the solution to Equations (17) and (18) are identically zero. Also, it is obvious that if the method converges, all e_i and ϵ_i approach zero. In light of the above, the homogeneous and particular solutions can best be obtained by the technique in *M*2. Denote the *M*2 algorithm applied to the error equations of Sylvester and Meyer as the *M*4 method.

Even though there are numerous ways of solving a system of linear algebraic equations, such as those in $\boldsymbol{\alpha}^{(m+1)}$, most of the methods work well only if the coefficient matrix is not ill-conditioned. This situation is likely to occur in optimal control problems due to the instability of the adjoint equations to forward integration (*1*). Bellman and Kalaba (*3*) suggest a way of circumventing this difficulty.

As before, let $\mathbf{e}_h^{(i)}$ and $\boldsymbol{\epsilon}_h^{(i)}$ represent the i^{th} homogeneous solution to Equations (17) and (18). In the generation of the homogeneous solutions it is likely that the components of $\boldsymbol{\epsilon}_h^{(i)}$ will become large due to the instability of the adjoint equations. It is desirable that the homogeneous solutions be linearly independent, but if the components of $\boldsymbol{\epsilon}_h^{(i)}$ grow rapidly as integration proceeds this condition will not be met.

Consider the *M*4 method and denote the $n \times 2n$ matrix of homogeneous solutions by $\mathbf{H}(t)$. The i^{th} row of $\mathbf{H}(t)$ is the vector $[e_{h_1}{}^{(i)}(t), \ldots, e_{h_n}{}^{(i)}(t), \epsilon_{h_1}{}^{(i)}(t), \ldots, \epsilon_{h_n}{}^{(i)}(t)]$. For the *M*4 method

$$\mathbf{H}(t_o) = \left[\begin{array}{c:cccc} & s & & & 0 \\ & & s & & \\ 0 & & & \cdot & \\ & & & & \cdot \\ & 0 & & & s \end{array}\right] \tag{19}$$

If the total number of integration steps from $t = t_o$ to $t = t_f$ is N, then Bellman and Kalaba's algorithm is the following.

1. Integrate the homogeneous form of Equations (17) and (18) from $\mathbf{H}(t_o)$ to $t = t_1$, $t_o < t_1 < t_f$.
2. Orthonormalize the rows of $\mathbf{H}(t_1)$, at the same time carrying out the same linear transformation on $\mathbf{H}(t_o)$.
3. Integrate from the new $\mathbf{H}(t_1)$ to $t = t_2$, $t_1 < t_2 < t_f$.
4. Orthonormalize $\mathbf{H}(t_2)$ as before, again transforming $\mathbf{H}(t_o)$.
5. Proceed as above until $t = t_f$ is reached.

The result is a matrix of final values, $\mathbf{H}(t_f)$, with orthonormal rows and a matrix of initial conditions, $\mathbf{H}(t_o)$, which will generate $\mathbf{H}(t_f)$. Once $\mathbf{H}(t_o)$ is found, $\mathbf{e}_h^{(j)}(t)$ and $\boldsymbol{\epsilon}_h^{(j)}(t)$ can be determined as before by combining the homogeneous and particular solutions. From here there are two options. One may use the $\mathbf{H}(t_o)$ matrix from the m^{th} iteration as the initial $\mathbf{H}(t_o)$ matrix for the $(m + 1)^{\text{th}}$ iteration or the $\mathbf{H}(t_o)$ given by Equation (19).

Call this method *M*5 and denote by *M* the number of numerical integration steps taken in going from t_1 to t_2, as described in step 3 of the above scheme. Let the two options be option 1 and option 2, respectively.

Obviously the computation time of *M*5 is much greater than that of *M*4, particularly if the orthonormalization must be performed many times between $t = t_o$ and $t = t_f$. However, *M*5 does, in theory, appear to alleviate the difficulties caused by unstable adjoint variables.

COMPUTATIONAL CONSIDERATIONS

Roughly speaking, the methods based on superposition can be ranked in order of increasing complexity or expectance of success as follows: *M*1, *M*2, *M*3, *M*4, and *M*5. One might start to solve a particular problem by using *M*1 and then proceed upward until a solution is obtained. A summary of the salient features of these methods is listed in Table 1.

In employing superposition, it has been assumed that the homogeneous and particular solutions can be stored at every grid point $t = \Delta, 2\Delta, \ldots, N\Delta$. N is the number of integration steps taken in numerically integrating from t_o to t_f. These solutions are then combined to yield the $(m + 1)^{\text{th}}$ approximation once $\boldsymbol{\alpha}^{(m+1)}$ has been calculated. On this basis the number of storage locations required by the *M*2, *M*4, and *M*5 methods for digital computation is

$$C(n, N) = 2n\;N(n + 1) \tag{20}$$

where n is the dimension of the state vector. If the maximum number of storage locations available to the user is 30,000, then for $N = 20$ the upper limit on n is 26. For

Table 1. A Comparison of the Superposition Variations of Quasilinearization

Method	Linearized equations given by	Particular solution Initial conditions		ρ, Number of homogeneous solutions	Homogeneous solution Initial conditions, $k = 1, 2, \ldots, \rho$		Adjoint Equation stabilization
		On $\mathbf{x}$ or $\mathbf{e}$	On $\mathbf{z}$ or $\mathbf{e}$		On $\mathbf{x}$ or $\mathbf{e}$	On $\mathbf{z}$ or $\mathbf{e}$	
M1	Equations (10) and (11)	0	0	$2n$*	$\begin{cases} s\boldsymbol{\delta}^{(k)}, k \leq n \\ 0, k > n \end{cases}$	$0, k \leq n$ $s\boldsymbol{\delta}^{(k-n)}, k > n$	No
M2	Equations (10) and (11)	$\mathbf{x}^o$	0	n	0	$s\boldsymbol{\delta}^{(k)}$	No
M3	Equations (10) and (11)	$\mathbf{x}^{(m)}(t_o)$	$\mathbf{z}^{(m)}(t_o)$	$2n$	$\begin{cases} \mathbf{x}^{(m)}(t_o) + s\boldsymbol{\delta}^{(k)}, k \leq n \\ \mathbf{x}^{(m)}(t_o), k > n \end{cases}$	$\mathbf{z}^{(m)}(t_o), k \leq n$ $\mathbf{z}^{(m)}(t_o) + s\boldsymbol{\delta}^{(k-n)}, k > n$	No
M4	Equations (17) and (18)	0	0	n	0	$s\boldsymbol{\delta}^{(k)}$	No
M5	Equations (17) and (18)	0	0	n	0	$s\boldsymbol{\delta}^{(k)}$	Yes

* n is the dimension of the state vector.

$N = 100$, n can be no larger than 11. If the step size must be reduced further and $N = 1{,}000$, only a three-variable problem can be accommodated.

Most of the storage requirement given by Equation (20) is due to the necessity of retaining the full grid of homogeneous and particular solutions. Kenneth and Taylor (9) propose a way of eliminating this. The calculation of $\boldsymbol{\alpha}^{(m+1)}$ requires at most only the initial and final values of the homogeneous and particular solutions for the type of boundary-value problem considered here. With this in mind, Kenneth and Taylor suggest keeping only those values and the m^{th} approximation. Once $\boldsymbol{\alpha}^{(m+1)}$ is determined, the initial value of the $(m + 1)^{\text{th}}$ approximation is calculated and the linearized equations are integrated forward from this initial condition to yield the $(m + 1)^{\text{th}}$ approximation, which is stored over the m^{th} approximation. The storage requirement is now reduced to

$$C(n, N) = 2nN \tag{21}$$

For $N = 1{,}000$, the maximum allowable n for a 30K memory is 15. This is a much more acceptable value. Note that this technique requires one additional integration, but the increase in computation time will be negligible.

Bellman (2) proposes a procedure which eliminates the need for retaining the $(m + 1)^{\text{th}}$ approximation entirely. At each iteration k, Bellman stores only the initial value of the k^{th} approximation. With the assumption that the initial guess is written as a function of t and need not be stored at each grid point, then to generate the m^{th} approximation needed to calculate the $(m + 1)^{\text{th}}$ approximation, the linear equations must be integrated m times, from $k = 1$ to $k = m$, using the m sets of stored initial conditions. This means that $2mn$ additional equations must be integrated at the $(m + 1)^{\text{th}}$ iteration. Although the storage requirement is quite small, the computation time will be greatly increased.

In any case, the maximum n for feasible numerical solution is not determined by storage limitations, but by the amount of computation time needed at each iteration. In other words, quasilinearization is a time-limited rather than storage-limited algorithm. At this point, the effect of n on the number of iterations required for convergence is unknown. It is conceivable that as n increases the number of iterations will also rise due to the increased complexity of the system. This would add an additional time restriction on the maximum feasible n.

A NUMERICAL EXAMPLE

To illustrate the computational difficulties and advantages of quasilinearization, we consider here the transient behavior of a plate type of absorption tower (*11*). The state equations are given by

$$h_g \frac{dy_i}{dt} + h_l \frac{dx_i}{dt} = L[x_{i-1} - x_i] + G(y_{i+1} - y_i), \qquad i = 1, \ldots, N \tag{22}$$

and we assume the nonlinear equilibrium relationship

$$y_i = \alpha x_i + \beta x_i^2 \tag{23}$$

With $N = 6$ for an explicit formulation, the state vector is $\mathbf{x}(t) = [x_1, \ldots, x_6]$ and the control vector $\mathbf{u}(t)$ is made up of the two entering feed compositions $x_0(t)$ and $y_7(t)$, $\mathbf{u}(t) = [x_0(t), y_7(t)]$. Given some initial starting steady state operating condition of the tower, it is desired to determine the optimal $x_o(t)$ and $y_7(t)$ which will minimize an integral weighted sum of squares of the deviation of $\mathbf{x}(t)$, $x_0(t)$,and $y_7(t)$ from a final desired steady state. Since $y_7(t)$ can be related to a corresponding liquid composition by Equation (23), the performance index which we wish to minimize is given by

$$I = \int_0^{t_f} \sum_{i=0}^{7} w_i\, [x_i(t) - \overline{x}_i]^2\, dt \tag{24}$$

where the w_i are weighting factors and $\overline{x}_i$ are the final and desired values. We also rewrite the control vector as $\mathbf{u}(t) = [x_o(t), x_7(t)]$. All the actual numerical parameters and steady states may be found in Table 2.

This performance index effectively penalizes the system for deviations from the desired steady state. Note that the control variables $x_o(t)$ and $x_7(t)$ are constrained in a sense, since large deviations from $\overline{x}_o$ and $\overline{x}_7$ are penalized by the inclusion of the control variables in the objective function.

Using Equation (23), we can rewrite Equation (22) as a vector state equation:

Table 2. Numerical Parameters

$N = 6$	$L = 40.8$ lb./min.	System parameters
$\alpha = 0.72$	$G = 66.7$ lb./min.	
$\beta = 0.08$	$h_g = 1.0$ lb.	
$t_o = 0$	$h_l = 75.0$ lb.	

$x_0(0) = 0$	Initial steady state	$\overline{x}_0 = 0$	Desired steady state
$x_1(0) = 0.0642757$		$\overline{x}_1 = 0.0985273$	
$x_2(0) = 0.117801$		$\overline{x}_2 = 0.179724$	
$x_3(0) = 0.161906$		$\overline{x}_3 = 0.245595$	
$x_4(0) = 0.197937$		$\overline{x}_4 = 0.298369$	
$x_5(0) = 0.221700$		$\overline{x}_5 = 0.340235$	
$x_6(0) = 0.250754$		$\overline{x}_6 = 0.373192$	
$x_7(0) = 0.269696$		$\overline{x}_7 = 0.398979$	

$$\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{25}$$

where $\mathbf{x} = [x_1, \ldots, x_6]$ and $\mathbf{u} = [x_o, x_7]$. Applying the maximum principle, we find the Hamiltonian:

$$H(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \sum_{i=1}^{6} \dot{x}_i z_i - \sum_{i=0}^{7} w_i(x_i - \overline{x}_i)^2 \tag{26}$$

The adjoint equations are

$$\frac{dz_1}{dt} = \left[L + G(\alpha + 2\beta x_1) + \frac{2\beta h_g D_1}{A_1} \right] \frac{z_1}{A_1} - \frac{L}{A_2} z_2 + 2w_1(x_1 - \overline{x}_1) \tag{27}$$

$$\frac{dz_j}{dt} = - \frac{G(\alpha + 2\beta x_j)}{A_{j-1}} z_{j-1} + \left[L + G(\alpha + 2\beta x_j) + \frac{2\beta h_g D_j}{A_j} \right] \frac{x_j}{A_j} - \frac{L}{A_{j+1}} z_{j+1} + 2w_j(x_j - \overline{x}_j), \quad j = 2, \ldots, 5 \tag{28}$$

and

$$\frac{dz_6}{dt} = - \frac{G(\alpha + 2\beta x_o)}{A_5} z_5 + \left[L + G(\alpha + 2\beta x_o) + \frac{2\beta h_g D_6}{A_6} \right] \frac{z_6}{A_6} + 2w_6(x_6 - \overline{x}_6) \tag{29}$$

where

$$D_i \equiv L(x_{i-1} - x_i) + G[\alpha(x_{i+1} - x_i) + \beta(x_{i+1}{}^2 - x_i{}^2)] \tag{30}$$

and

$$A_i \equiv h_g(\alpha + 2\beta x_i) + h_l \tag{31}$$

Maximizing $H(\mathbf{x}, \mathbf{z}, \mathbf{u})$ at every t yields for $\mathbf{u}(t) = [x_o(t), x_7(t)]$:

$$x_o = \overline{x}_o + \frac{Lz_1}{2A_1 w_o} \tag{32}$$

and

$$x_7 = \frac{2A_6 w_7 \overline{x}_7 + \alpha G z_6}{2(A_6 w_7 - \beta G z_6)} \tag{33}$$

The boundary conditions on Equations (27) to (29) are $\mathbf{z}(t_f) = \mathbf{0}$.

Following quasilinearization, Equations (25) and (27) to (29) are linearized, after substitution of Equations (32) and (33) for $\mathbf{u}(t)$. The linearization is straightforward (*17*) and is not reproduced here.

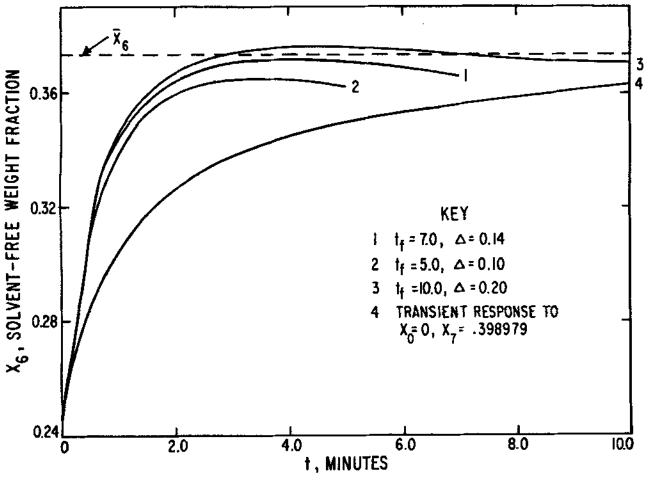

Fig. 1. $x_6(t)$ vs. t for different final t_f's. Method M3 is used.

Final times t_f of from 5 to 15 min. were investigated with a weighting function, in most cases, of $w_i = 1.0, i = 0, 1, \ldots, 7$. The numerical integration step size Δ was varied and the results indicate that a step size of 0.35 min. or smaller is sufficient to give six-digit precision. ϵ was set at 10^{-4}. This corresponds to less than 0.5% of the difference between $x_i(0)$ and $\overline{x}_i$, the desired final state. $\mathbf{z}(t)$ was not required to converge to within $\epsilon = 10^{-4}$, however. For $\mathbf{z}(t)$, $\epsilon = 0.01$ was felt to be sufficiently stringent value for two reasons. First, accuracy in the adjoint variable trajectory is required only to determine the control trajectory. Second, $\mathbf{z}(t)$ is more difficult to determine than $\mathbf{x}(t)$ because of instabilities in the adjoint equations. A small ϵ for $\mathbf{z}(t)$ would cause excessive iteration. For these two reasons $\mathbf{z}(t)$ was assumed to have converged if $\mathbf{z}^{(m)}(t)$ and $\mathbf{z}^{(m+1)}(t)$ deviated by no more than $\epsilon = 0.01$ even though $\epsilon = 10^{-4}$ for $\mathbf{x}(t)$. In what is to follow, whenever a value of ϵ is given it shall be taken to be ϵ for $\mathbf{x}(t)$, and ϵ for $\mathbf{z}(t)$ will be 10^2 times the ϵ for $\mathbf{x}(t)$.

The initial control policy was assumed to be

$$\left. \begin{aligned} x_o(t) &= \overline{x}_o + \zeta_1 \exp\left(-\frac{10t}{t_f} - 1\right) \\ x_7(t) &= \overline{x}_7 + \zeta_2 \exp\left(-\frac{10t}{t_f} - 1\right) \end{aligned} \right\} \tag{34}$$

where the constants were chosen so that the control policy was reasonable. This guess was suggested by experience and intuition only. For most runs, the values of ζ_1 and ζ_2 were 0.1 and 0.4, respectively.

A brief word about the choice of t_f is in order to start our discussion. Note that the assumption that t_f is fixed is implicit in the development of the basic system equations. Strictly speaking, t_f is not known and if t_f remains free it turns out that the optimum value of t_f for this formulation is infinite. This is unrealistic from a numerical point of view so t_f is treated as fixed. The question to be answered, then, is what is a reasonable value of t_f? An examination of the variation of the optimal trajectory $\mathbf{x}^*(t)$ with t_f answers this. As an illustration Figure 1 shows the result for $x_6{}^*(t)$ calculated by the *M*3 method with $s = 0.02$ for $t_f = 5.7$ and 10 min. Also shown in Figure 1 is the transient response of x_6 to a control of $x_o(t) = \overline{x}_o$ and $x_7(t) = \overline{x}_7$ for all t, that is, the control is switched immediately to the level which corresponds to the desired steady state $\overline{\mathbf{x}}$. Figure 1 reveals that $x_6(t_f)$ is reasonably close to $\overline{x}_6$ for $t_f = 10$. These same results are also obtained for $x_1(t), \ldots, x_5(t)$. On the other hand, the approach to $\overline{x}_6$ for $t_f = 5$ is not very close and for $t_f = 7$ is barely acceptable. Qualitatively, then, it is safe to choose $t_f = 10$ min. as a realistic final time, with $t_f = 7$ min. probably not being too out of line.

DISCUSSION OF RESULTS

Table 3 summarizes the results of the various methods for $t_f = 7$. It is seen that the *M*1 and *M*2 methods failed to converge with $\epsilon = 10^{-4}$. Furthermore, the *M*4 technique, while converging for the $t_f = 7$ case, did not converge for $t_f = 10$. Only the *M*3 and *M*5 techniques met the $\epsilon = 10^{-4}$ requirement for $t_f = 10$. For $\epsilon = 10^{-4}$ they required twelve and six iterations, respectively, to converge. Figures 2 and 3 show the resulting x_6 trajectories with two successful methods used for $t_f = 10$.

At the same time we note further that with the *M*3 method the value of the Hamiltonian as a function of t_f is given by: $t_f = 5$, $H = -7.521 \times 10^{-3}$; $t_f = 7$, $H = -3.237 \times 10^{-3}$; and $t_f = 10$, $H = -8.65 \times 10^{-4}$. In

Table 3. A Comparison of Numerical Methods for Solving the Gas Absorber Problem with $t_f = 7$ and $\Delta = 0.14$

Method	No. of iterations	Computation time/ iteration, sec.	Remarks
$M1, s = 1$	18	40	Oscillating in the third significant figure. Missing $\mathbf{z}(t_f) = \mathbf{0}$ by as much as 0.04 in the 18th trial.
$M1, s = 0.1$	19	40	Oscillating in the third significant figure. Missing $\mathbf{z}(t_f) = \mathbf{0}$ by as much as 0.008 in the 19th trial.
$M2, s = 1$	33	23	Oscillating in the fourth significant figure. Missing $\mathbf{z}(t_f) = \mathbf{0}$ by as much as 0.09 in the 33rd trial.
$M2, s = 0.1$	33	23	Oscillating in the fourth significant figure. Missing $\mathbf{z}(t_f) = \mathbf{0}$ by as much as 0.03 in the 33rd trial.
$M3, s = 0.02$	4	40	Converges. H constant to four significant figures. All $z_i(t_f) < 10^{-9}$.
$M4, s = 1$	3	30	Converges, Agrees with $M3$ run to six significant figures in **x**, **z**, and **u**. H constant to three significant figures.
$M5, s = 0.001$ $M = 10$ (option 1)	3	55	Converges. Agrees with $M3$ run to five significant figures in **x**, **z**, and **u**. H constant to three significant figures.

other words the Hamiltonian appears to be approaching zero as t_f increases; this is the expected behavior since an infinite t_f is the optimal final time.

No results were obtained by any method for $t_f = 12$ min. or more. However, the $M5$ method came very close to converging for $t_f = 12$ and some effort was made to improve the algorithm. These modifications will be presented shortly, along with a discussion of their results.

A COMPARISON OF THE QUASILINEARIZATION VARIATIONS

To begin the discussion of why no solution was obtained for $t_f > 10$ and why some methods worked while others did not, consider the stability of the linearized differential equations. These equations are integrated from $t = 0$ to $t = t_f$ in the generation of the homogeneous and particular solutions. It has already been pointed out that it is not uncommon for the adjoint equations to be unstable in forward integration if the state equations are stable. An idea of the stability of the linearized equations can be gleaned from an examination of the eigenvalues of the coefficient matrix A, where the homogeneous equations are written as

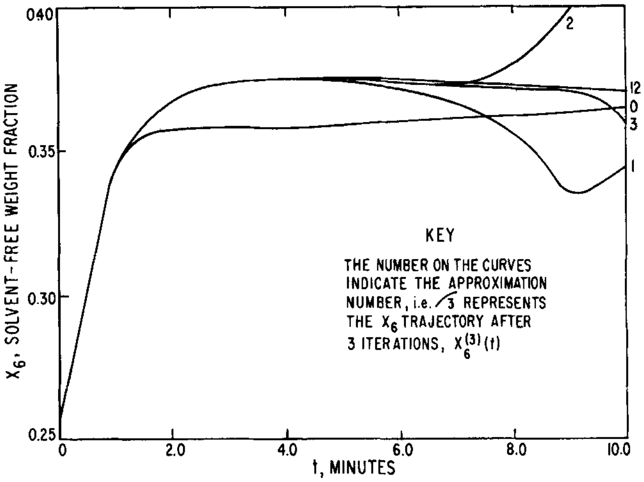

Fig. 2. $x_6(t)$ vs. t for $t_f = 10$. Method M3 is used.

$$\begin{bmatrix} \dot{\mathbf{x}}^{(m+1)} \\ \dot{\mathbf{z}}^{(m+1)} \end{bmatrix} = \mathbf{A}^{(m)} \begin{bmatrix} \mathbf{x}^{(m+1)} \\ \mathbf{z}^{(m+1)} \end{bmatrix} \tag{35}$$

Note that $\mathbf{A}^{(m)}$ is a function of t. For $m = 5$ the eigenvalues of **x** at $t = 0$, $t = 5$, and $t = 10$ start approximately at $\lambda_1 = -2.30$ and finish with $\lambda_6 = -0.22$; the eigenvalues of **z** are merely the negative of those for **x** but in reverse order. The effect of m was less than 0.02 for all eigenvalues. The presence of positive λ_i indicates instability. The numerical difficulties are best illustrated by considering $e^{\lambda_i t_f}$. Using $\max|\lambda_i|$ and $t_f = 10$, $e^{\lambda_i t_f}$ is the order of 10^{10}. Thus $\mathbf{x}_h(t)$ and $\mathbf{z}_h(t)$ grow rapidly in integrating from $t = 0$ to $t = t_f$. Furthermore, since $\mathbf{x}_h(t_f)$ and $\mathbf{z}_h(t_f)$ are used in calculating $\boldsymbol{\alpha}^{(m+1)}$, accuracy is especially needed at $t = t_f$. It is easily seen, then, that as t_f is increased accuracy is lost and the calculations involving $\mathbf{x}_h(t_f)$ and $\mathbf{z}_h(t_f)$ may even become too large for computer storage.

Recognizing the importance of the homogeneous and particular solutions at $t = t_f$, those methods using superposition will now be examined more closely. Write Equations (13) and (14), the set of linear algebraic equations in $\boldsymbol{\alpha}^{(m+1)}$, as

$$\mathbf{A}\ \boldsymbol{\alpha}^{(m+1)} = \mathbf{b} \tag{36}$$

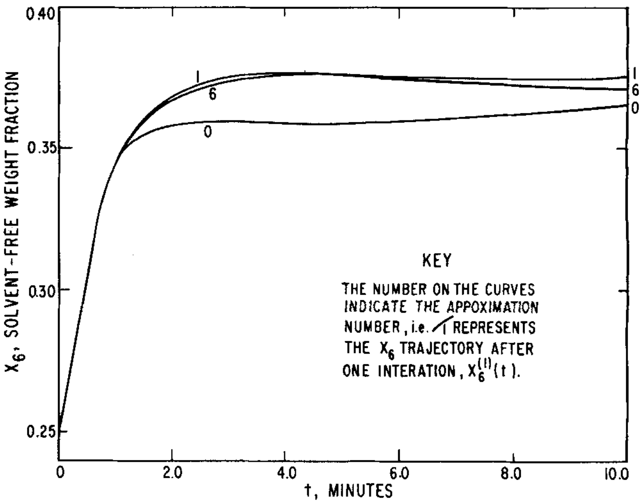

Fig. 3. $x_6(t)$ vs. t for $t_f = 10$. Method M5 with Option 2 is used.

TABLE 4. THE VARIATION OF THE ELEMENTS OF THE COEFFICIENT MATRIX A AND THE RIGHT-HAND SIDE **b** IN THE MATRIX A $\boldsymbol{\alpha}^{(m+1)} = \mathbf{b}$ FOR CALCULATING THE SUPERPOSITION CONSTANTS $\boldsymbol{\alpha}^{(m+1)}$ FOR $t_f = 7$ AND $\Delta = 0.14$

Iteration number, m	$M1, s = 1$		$M2, s = 1$		$M3, s = 0.02$		$M4, s = 1$		$M4$. Option 1, $M = 10$, $s = 0.001$	
	$\overset{(12)}{x_{h2}}(7)$	$-z_{p1}(7)$	$\overset{(6)}{x_{h2}}(7)$	$-z_{p2}(7)$	$\overset{(12)}{x_{h2}}(7)$	$-z_{p1}(7)$	$\overset{(6)}{\epsilon_{h2}}(7)$	$-\epsilon_{p2}(7)$	$\overset{(6)}{\epsilon_{h2}}(7)$	$-\epsilon_{p2}(7)$
1	526,762	—4474.15	1.2224×10^6	4344.45	63,300.9	—2241.26	1.2240×10^6	3098.67	0.50622	3098.67
2	534,357	—4511.42	1.2410×10^6	4278.08	60,549.7	7.1579	1.2424×10^6	—99.67	0.50649	—99.95
3	534,695	—4511.66	1.2407×10^6	4276.76	60,550.9	1.03×10^{-3}	1.2424×10^6	—18.72	0.50649	—20.01
4	534,811	—4511.38	1.2416×10^6	4277.64	60,550.4	5.59×10^{-9}				
5	534,793	—4511.11	1.2419×10^6	4278.26						
10	534,322	—4511.38	1.2415×10^6	4277.28						

where the right-hand side vector **b** has elements $[x_i^o - x_{Pi}(0)]$ and $[-z_{Pi}(t_f)]$ and the coefficient matrix **A** contains elements like $[z_{h_i}^{(i)}(t_f)]$. The variation of typical elements of **A** and **b** with m is given in Table 4 for all the methods under discussion with $t_f = 7$ and $\Delta = 0.14$. All six techniques exhibited little change in the coefficient matrix with m, although the elements of **A** in $M5$ were five or six orders of magnitude smaller than those in $M1$, $M2$, $M3$, or $M4$. On the other hand, the elements of **b** remained roughly constant with m only in $M1$ and $M2$, dropping at least three orders of magnitude before convergence in $M3$, $M4$, and $M5$. The difficulty caused by the marked instability of the adjoint equations, as presented in the previous paragraph, is displayed in Table 5. Here the increase in the magnitude of the largest element in **A** with t_f is given for the $M3$ method with $s = 0.02$.

First, we wish to point out the difference between these results and others (17), not reported here, on the famous Amundson-Bilous tubular reactor problem (4). In the reactor problem the elements of **A** and max $|\lambda_i|$ decreased as m increased and exp $[(\max |\lambda_i|)\ t_f]$ was on the order of 10^5 for the worst initial guess. In the present case, as has already been noted, the elements of **A** and max $|\lambda_i|$ were relatively constant and exp $[(\max |\lambda_i|)\ t_f]$ was on the order of 10^{10}. In the reactor problem the elements in **A** decreased as the approximation improved and this made the calculation of $\boldsymbol{\alpha}^{(m+1)}$ more accurate with each successive iteration. This was not the case with the absorber problem. The conclusion to be drawn from this is that the adjoint equations for the absorber are so much more unstable than those for the tubular reactor that the difficulties inherent in this mask any improvement that convergence provides.

TABLE 5. A COMPARISON OF THE ABSOLUTE VALUE OF THE LARGEST ELEMENT IN THE COEFFICIENT MATRIX FOR CALCULATING $\boldsymbol{\alpha}^{(j)}$ IN THE M3 METHOD WITH $s = 0.02$

t_f, min.	Δ, min.	Absolute value of the largest element the coefficient matrix	Iteration number, j
5	0.1	1.27×10^3	1
5	0.1	1.24×10^3	2 (final iteration)
7	0.14	1.10×10^5	1
7	0.14	1.08×10^5	4 (final iteration)
10	0.2	9.75×10^7	1
10	0.2	9.78×10^7	12 (final iteration)
12	0.24	9.44×10^9	1

Second, as the approximate trajectory converged to the optimal trajectory, the particular solution calculated by the $M3$, $M4$, and $M5$ methods approached the optimal trajectory. This means that $\mathbf{z}_p(t_f)$ decreased in magnitude as m increased. The reason this occurred is that the initial conditions for the $(m + 1)^{\text{th}}$ particular solution in the $M3$ method are $\mathbf{x}_p(0) = \mathbf{x}^o$ and $\mathbf{z}_p(0) = \mathbf{z}^{(m)}(0)$ and $\mathbf{z}^{(m)}(0)$ approaches $\mathbf{z}^*(0)$ as m increases. In the $M4$ and $M5$ methods $\mathbf{e}_p(0) = \mathbf{0}$ and $\boldsymbol{\varepsilon}_p(0) = \mathbf{0}$, which are the values to which the error vectors converge if convergence is obtained. This correspondence of the particular solution initial conditions to the initial conditions of the optimal trajectory is not found in the $M1$ or $M2$ methods. Note also that as m increases the linearized equations more closely approximate the nonlinear equations. This also aids in reducing the magnitude of $\mathbf{z}_p(t_f)$ as iteration proceeds. The result of all the above is a more accurately determined $\boldsymbol{\alpha}^{(m+1)}$.

Third, the magnitude of the elements of **A** in the $M5$ method was much lower than those obtained by any of the other techniques. This was due to the fact that the orthonormalization specifically determines that homogeneous solution initial condition matrix, $\mathbf{H}(0)$, which, at least theoretically, produces an orthonormal set of homogeneous solutions at $t = t_f$. Again this enables a more exact $\boldsymbol{\alpha}^{(m+1)}$ to be obtained. Note also that the $M5$ method reduced the magnitude of the elements in **b**, as seen in the previous paragraph, further increasing the accuracy of $\boldsymbol{\alpha}^{(m+1)}$. The combination of these two effects is probably the reason the $M5$ method took only six iterations to converge with $\epsilon = 10^{-4}$ for $t_f = 10$, compared to twelve iterations for the $M3$ method, the only other one which converged for $t_f = 10$.

Fourth, the $M1$ and $M3$ methods require $2n$ homogeneous solutions (twelve in the case of the absorber problem), while the $M2$ and the $M4$ algorithms produce only n. This reduction of the number of homogeneous solutions by a factor of two almost halved the computation time per iteration. While no significant improvement in convergence was evident, this time saving feature is of considerable merit.

Finally, the $M4$ and $M5$ methods converged faster than the $M1$, $M2$, and $M3$ methods. The explanation for this is obvious, namely, that the former two deal with the error vectors, which means that all numerical errors are associated with quantities which are already small. This was the reasoning behind the development of the error vector approach in the first place, so it is not surprising that the numerical results should substantiate this reasoning.

Based on the above discussion and the conclusions drawn from the results, the various methods can be ranked, at least qualitatively, according to their success and usefulness. The most probable order is $M1$, $M2$, $M4$, $M3$, and $M5$, with the $M5$ algorithm the best. It is somewhat ques-

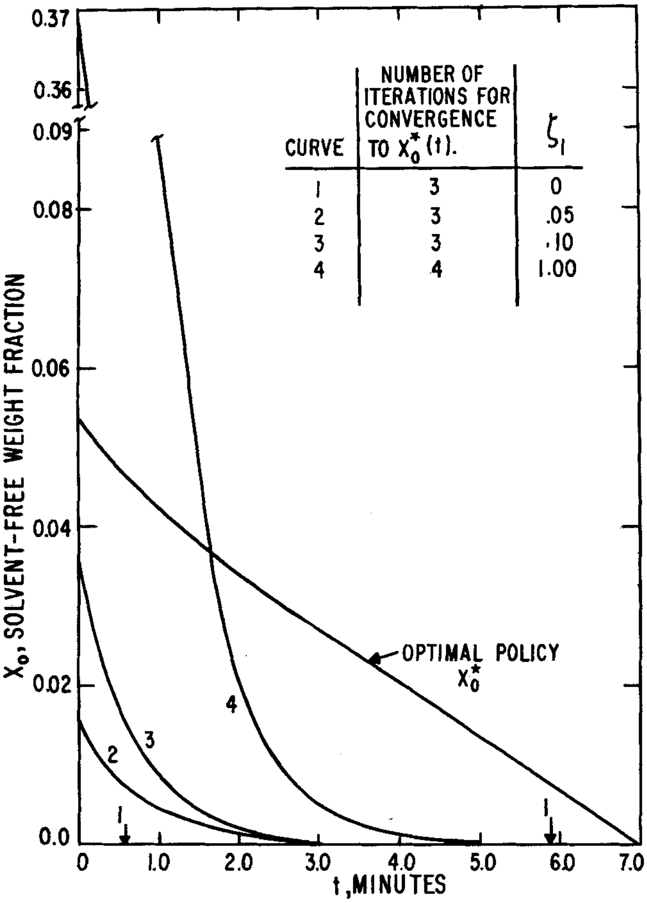

Fig. 4. The effect of $x_0(t)$ starting policy. Method M5 with Option 1 is used.

tionable whether $M3$ is indeed better than $M4$ or not. The $M3$ method converged for $t_f = 10$, while $M4$ did not, but the $M3$ technique requires twice as much computation time per iteration. There is no doubt, however, that the $M5$ algorithm is the most successful of the five investigated.

ATTEMPTS TO CONVERGE WITH $t_f > 10$

The inability to generate an optimal policy for $t_f > 10$ is disturbing and an attempt at correcting this by changing the weighting factors w_i in the objective function was made. Convergence of x_6 is particularly bad and the same is true of x_1, but not of the other plate compositions. The reason for this is the physical proximity of trays 1 and 6 to the control. Changes in control markedly affect x_1 and x_6, but the effect on the other trays is damped out. Weighting the objective function so that the effect of x_0 and x_7 on the internal plates is more important than the effect of control on plates 1 and 6 might be expected to produce smoother x_1 and x_6 trajectories. This, in turn, might allow convergence for $t_f > 10$. Admittedly the problem has been changed so that comparison with the case with $w_i = 1$ for all i will not have meaning. Nevertheless, w = (1, 1, 5, 10, 10, 5, 1, 1), w = (10, 1, 5, 10, 10, 5, 1, 10), w = (1, 1, 3, 7, 7, 3, 1, 1) and w = (10, 1, 3, 7, 7, 3, 1, 10) with $t_f = 10$ were tried with the $M3$ method. In no case was convergence obtained. Because of the dismal failure at $t_f = 10$ further efforts in this direction were dropped.

The $M5$ method, option 1, was put to the test with $t_f = 12$ for M values of 10 and 25 and $\Delta = 0.24$. Convergence was almost but not quite realized. This stimulated attempts to improve on these encouraging results. It was felt that error propagation resulting from the repeated linear transformations made on the homogeneous solution initial condition matrix $\mathbf{H}(0)$ could be important. Theoretically, the homogeneous solutions at $t = t_f$, obtained by integrating forward from $\mathbf{H}(0)$, should agree with the $\mathbf{H}(t_f)$ found in the orthonormalization procedure. Practically, this was not the case and deviations of as much as 10^3 were sometimes found. Using double precision cut these errors by an order of magnitude. In a further modification, option 2 was invoked in an effort to prevent this error propagation from being carried from one iteration to the next. This could happen with option 1, since the final $\mathbf{H}(0)$ from the m^{th} iteration is used as the initial $\mathbf{H}(0)$ in the $(m + 1)^{\text{th}}$ iteration. Unfortunately, $\epsilon = 10^{-4}$ criterion was not satisfied by any $M5$ scheme. It is impossible to say which option was best or whether double precision was really warranted. For this problem at least, Bellman and Kalaba's orthonormalization decreased the number of iterations needed over the $M3$ scheme if convergence was obtained, but failed to increase significantly the range of t_f over which solutions were realizable.

THE EFFECT OF THE STARTING POLICY ON CONVERGENCE

For all the runs previously reported the starting x_0 and x_7 policies are given by Equation (34) with $\zeta_1 = 0.1$ and $\zeta_2 = 0.4$. Figures 4 and 5 show the effect of ζ_1 and ζ_2 on the convergence of the $M5$ method, option 1. In Figures 4 and 5, as before, $\epsilon = 10^{-4}$ was the criterion for convergence. Figure 4 lists the results for $t_f = 7$ and it is evident that convergence is obtainable from virtually any starting policy. Figure 4 also shows that the exponential form used

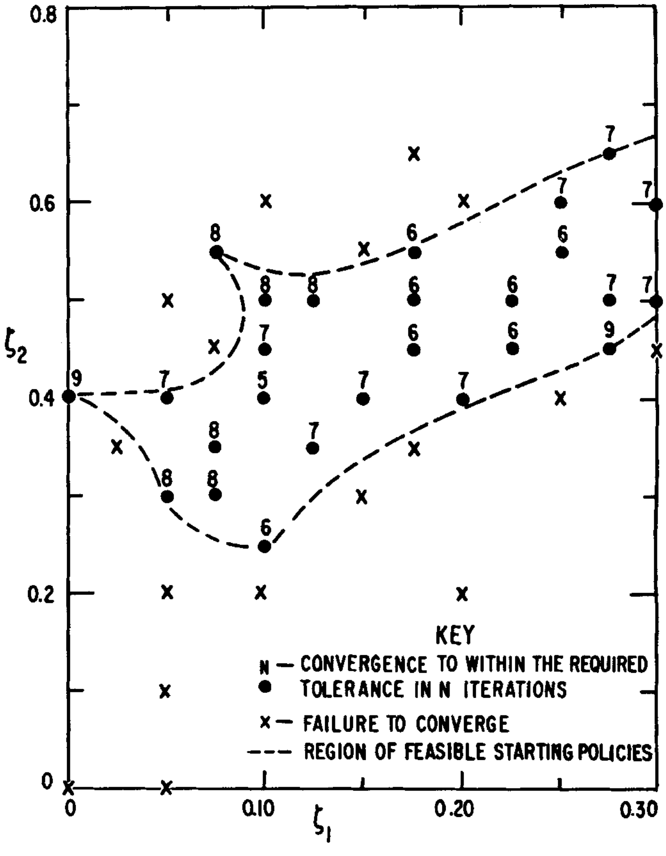

Fig. 5. Region of feasible starting policies. Method M5 with option 1 is used.

in the initial policy given by Equations (34) is not a very good approximation to the optimal policy. Figure 5 demonstrates that there is a region of initial policies outside of which convergence is impossible. This region, given for $t_f = 10$ in Figure 5, is evidently very dependent on t_f. Once again the effect of t_f on the numerical solution of the gas absorber problem is felt. However, considering the rather poor starting policy given by Equations (34) the region of feasible initial policies is large enough so that considerable freedom is available, even for $t_f = 10$.

SUMMARY OF RESULTS

The numerical solution of the gas absorber problem leads to the following conclusions:

1. Although the equilibrium expression, $y = 0.72x + 0.08x^2$, does not make the gas absorber system very nonlinear, the dimensionality of the problem and the marked instability of the adjoint equations make it a good test of the quasilinearization procedure.
2. Of the various forms of quasilinearization studied here, the $M5$ method gave the most satisfactory results.
3. The instability of the adjoint equations to some extent limited the effectiveness of even the $M5$ method. It was impossible to obtain convergence with the desired accuracy for t_f larger than 10 min. Perhaps fortuitously, $t_f = 10$ represents a reasonable value for the final time. Nevertheless, it is possible to predict when adjoint variable instability will make convergence impossible. Based on the numerical results of this investigation, if $\exp\left[\max_i |\lambda_i| \cdot (t_f - t_o)\right]$ is of the order of 10^{10} quasilinearization may be expected to fail.
4. The computation time on an IBM 7094 digital computer was never greater than 8 min.
5. The number of iterations required for convergence, when convergence was obtained, was always less than ten.
6. A bounded region of feasible starting policies exists. However, the region is sufficiently large so that very little experimentation is needed to locate it.

ACKNOWLEDGMENT

Financial assistance in the form of a Ford Foundation Fellowship, a Research Assistantship under National Science Foundation Grant GK-460, and a Standard Oil Company of California Fellowship is gratefully acknowledged. Use was made of Princeton University's computer facilities, supported in part by National Science Foundation Grant GP-579.

NOTATION

A_i = function defined by Equation (31)
$\mathbf{A}$ = coefficient matrix
$\mathbf{b}$ = column vector
$\mathbf{c}$ = performance index coefficients
C = required computer storage
D_i = function defined by Equation (30)
$\mathbf{e}$ = state variable error vector $[e_1, \ldots, e_n]$
$\mathbf{f}$ = vector function
$\mathbf{F}$ = vector function
$\mathbf{g}$ = vector function
G = gas flow rate, lb./min.
h_l = liquid holdup per plate, lb.
h_g = gas holdup per plate, lb.
H = Hamiltonian function
$\mathbf{H}$ = homogeneous solution matrix
I = performance index
L = liquid flow rate, lb./min.
M = number of numerical integration steps taken between orthonormalizations in the $M5$ method
n = dimension of the state vector
N = number of plates in the gas absorber; number of finite increments for numerical integration
r = dimension of the control vector
s = scalar multiplier
t = independent variable, usually time
$\mathbf{u}$ = control vector $[u_1, \ldots, u_r]$
$\mathbf{w}$ = weighting coefficient vector
$\mathbf{x}$ = state vector $[x_1, \ldots, x_n]$
x_i = solvent-free weight fraction in the liquid phase
y_i = solvent-free weight fraction in the gas phase
$\mathbf{z}$ = adjoint vector $[z_1, \ldots, z_n]$

Greek Letters

α = parameter in the gas-liquid equilibrium expression
$\alpha_i^{(m)}$ = superposition constant
B = parameter in the gas-liquid equilibrium expression
$\delta_i^{(k)}$ = Kronecker delta
Δ = differential increment in t
ϵ = tolerance
$\boldsymbol{\epsilon}$ = adjoint variable error vector $[\epsilon_1, \ldots, \epsilon_n]$
ζ_1, ζ_2 = constants in Equation (34)
λ_i = eigenvalue of $\mathbf{A}^{(m)}$

Subscripts

f = final
g = gas phase
h = homogeneous solution
l = liquid phase
p = particular solution

Superscripts

(m) = m^{th} approximation; m^{th} solution
0 = initial
— = desired steady state
′ = transpose
* = optimal

LITERATURE CITED

1. Aris, Rutherford, *Chem. Eng. Sci.*, **13**, 18 (1960).
2. Bellman, R., *Comm. the ACM*, **4**, 222 (1961).
3. ———, and R. Kalaba, "Quasilinearization and Nonlinear Boundary-Value Problems," Elsevier, New York (1965).
4. Bilous, Olegh, and N. R. Amundson, *Chem. Eng. Sci.*, **5**, 81, 115 (1956).
5. Breakwell, J. V., J. L. Speyer, and A. E. Bryson, *J. SIAM Control*, Ser. A, **1**, 193 (1963).
6. Bryson, A. E., and W. F. Denham, *J. Appl. Mech.*, **29**, No. 2, 247 (1962).
7. Kalaba, R., *J. Math. Mech.*, **8**, 519 (1959).
8. Katz, S., *Ind. Eng. Chem. Fundamentals*, **1**, 226 (1962).
9. Kenneth, P., and G. E. Taylor, *Grumman Res. Dept. Memo. RM-289J* (July, 1965).
10. Kopp, R. E., and R. McGill, "Computing Methods in Optimization Problems," A. V. Balakrishnan and L. W. Neustadt, ed., p. 65, Academic Press, New York (1964).
11. Lapidus, Leon, E. Shapiro, S. Shapiro, and R. E. Stillman, *AIChE J.*, **7**, 288 (1961).
12. Lapidus, Leon, and Rein Luus, "Optimal Control of Engineering Processes," Blaisdell, New York (1967).
13. Lee, E. S., *Chem. Eng. Sci.*, **21**, 183 (1966).
14. McGill, R., and P. Kenneth, *AIAA J.*, **2**, 1761 (1964).
15. Moyer, H. G., and G. Pinkham, "Computing Methods in Optimization Problems," A. V. Balakrishnan and L. W. Neustadt, ed., p. 95, Academic Press, New York (1964).
16. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, "The Mathematical Theory of Optimal Processes," Interscience, New York (1962).
17. Rothenberger, B. F., Ph.D. dissertation, Princeton Univ., N. J. (1966).
18. Rozonoer, L., *Automation Remote Control*, **20**, 1288, 1405, 1517 (1960).
19. Sylvester, R. J., and F. Meyer, *J. SIAM*, **13**, 586 (1965).

Manuscript received November 9, 1966; revision received January 19, 1967; paper accepted January 23, 1967.