Iterative Procedures for Nonlinear Integral Equations

DONALD G. ANDERSON
Harvard University, Cambridge, Massachusetts

Abstract. The numerical solution of nonlinear integral equations involves the iterative solution of finite systems of nonlinear algebraic or transcendental equations. Certain conventional techniques for treating such systems are reviewed in the context of a particular class of nonlinear equations. A procedure is synthesized to offset some of the disadvantages of these techniques in this context; however, the procedure is not restricted to this particular class of systems of nonlinear equations.

1. *Introduction*

Nonlinear integral equations have gained increasing interest in recent years, both from an analytic and a numerical point of view; the current state of the art is reviewed in [1]. The present remarks stem from research in the kinetic theory of gases involving the solution of coupled sets of singular, nonlinear integral equations [2, 3, 4]. Since the methods developed for the solution of finite sets of nonlinear equations have proved useful in a number of other contexts, they are reported here.

In Section 2, we describe the class of nonlinear equations of interest here. In Section 3, we survey a number of numerical procedures which have been proposed for the solution of such equations, illustrating their deficiencies in the problem at hand and obtaining desiderata for a new approach. In Section 4, a class of procedures is proposed on a heuristic basis; these are illustrated in Section 5 by a number of simple examples. To the extent that these procedures represent devices for accelerating the convergence of iterative vector sequences, they may prove useful in a wider context than the solution of a particular class of nonlinear integral equations

2. *The Class of Nonlinear Equations*

The details of the original problem can be suppressed; it is necessary only to point out certain salient features which motivate the heuristic considerations to follow. For concreteness, consider the problem of finding $f(x)$ satisfying

$$f(x) = \int_a^b dt\, K(f(t); |x - t|) + g(x) \tag{2.1}$$

for a given kernel K and inhomogeneous term $g(x)$. This analytic problem must be replaced by a discrete analog, thereby reducing the problem to that of solving a finite system of nonlinear equations. The approach followed in [2] need only be outlined briefly: The integral operator is discretized by replacing it by an m-point Gaussian quadrature formula, or a composite of such formulas. The function $f(x)$ is replaced by a finite expansion in Chebyshev polynomials $T_k(x)$, adapted to the interval (a, b). A determinate problem is obtained by enforcing the resulting equations at a selected set of collocation or interpolation sample points x_i. The resulting

This work was supported in part by the National Science Foundation under Grant GP-414, the Smithsonian Astrophysical Observatory, and the Division of Engineering and Applied Physics, Harvard University.

Journal of the Association for Computing Machinery, Vol. 12, No. 4 (October, 1965), pp. 547–560

discrete dependent variables are the expansion coefficients c_k or the sample values $f_i = f(x_i)$. Choosing the x_i as the roots or extrema of an appropriate Chebyshev polynomial, we can relate the c_k and the f_i through the well-known discrete orthogonality conditions [5]. The resulting discrete problem takes the form

$$f_i = \sum_{j=1}^{m} w_j K\left(\sum_{k=0}^{n} c_k T_k(t_j); |x_i - t_j|\right) + g_i . \tag{2.2}$$

Since the sets of interpolation and quadrature sample points x_i and t_j are, in general, disjoint, it is not convenient to reformulate the problem in terms of c_k or f_i separately. Thus both the dual sets of unknowns and the form of the equation which arises naturally from the discretization of the analytic problem are retained.

The original problem was discretized in parallel fashion. The following properties of the corresponding equations motivate the heuristic arguments to follow and define the class of problems of interest. However, considerations which rule out certain approaches in this context are not operative in a general problem. The most important consideration is the amount of computation required to evaluate f_i from (2.2) given c_k, i.e., a unit for the computer time involved in solving the problem. In the problems of interest, the computation time is dominated by the evaluation of the kernel function K which is a complicated nonlinear function of its arguments. Due to the nature of K, a sensible discretization requires a considerable number of interpolation and quadrature sample points; a typical problem might involve 50–100 quadrature points and a comparable number of discrete dependent variables. The general shape of the solution is known a priori, so that reasonably good initial iterates are available. Nevertheless, even on a high speed digital computer like the IBM 7094, the problem is not feasible if the number of iterations required to solve the discrete problem is comparable to the dimensionality of the problem.

The second important consideration is the "shape" of the kernel function. $K(f; |x - t|)$ is singular at $x = t$ and decreases as $|x - t|$ increases. Let λ be a characteristic "width" of the kernel; the mode of convergence of any iterative procedure for solving the equations depends strongly on λ. For "large" λ there is strong global coupling of all f_i dependent variables; for "small" λ there is strong local coupling of "adjacent" f_i dependent variables, but little long range interaction. Indeed, for small λ the equation tends to become an identity for any f since the kernel becomes almost reproducing and the inhomogeneous term $g(x)$ has significant magnitude only near the boundaries $x = a$ and $x = b$. The basic successive substitution iteration suggested by the form of the equations then becomes extremely ill-conditioned and slowly convergent.

The influence of these considerations on the formulation of an iterative procedure may be seen below.

3. *Survey of Conventional Iterative Procedures*

Let us focus our attention on the prototype problem

$$Fz = 0 \tag{3.1}$$

where F is an operator on the N-vector z. For iterative solution, it is convenient to write the equation in the form

$$z = Gz \tag{3.2}$$

by some judicious choice of the iteration operator G. The problem falls naturally into the form (3.2), but in general one can write

$$Gz = z - HFz \tag{3.3}$$

for some sufficiently regular, homogeneous operator H. For the heuristic discussion to follow, one need not be more precise.

The basic successive substitution iteration

$$z^{l+1} = Gz^l \tag{3.4}$$

suggests itself quite naturally. Depending on the form of G, the components z_i^{l+1} can be corrected one at a time—a *local* iteration—or all at once—a *global* iteration. The dual nature of the discretization suggests a global approach. The convergence of the basic iteration (3.4) is an extensively studied problem [1] which is not discussed here. In practice, in a problem of this complexity the question of the rate of convergence (or divergence) must be answered empirically. For present purposes, it is assumed that the basic iteration converges in the usual geometric fashion, but too slowly to make the problem tractable. The question then becomes that of modifying the basic iteration so as to accelerate the convergence. In this paper, consider several such procedures, drawing heavily on analogies with the relatively well studied problems of iterative procedures for the solution of univariate nonlinear and multivariate linear equations. By Occam's Razor, we also consider the simplest procedures that seem likely to work.

Perhaps the simplest class of iterative procedures after the basic iteration are those of relaxation type, in which H of equation (3.3) is chosen as a *multiplicative constant* (the relaxation parameter) or a sequence of such constants. Such linear iterative processes have been analyzed for the univariate nonlinear case [6, 7] and have an extensive lore in the multivariate linear case [8]. Under certain conditions it is possible to determine, either empirically or in terms of properties of F, an optimum relaxation factor which maximizes the rate of convergence in some sense. For the multivariate nonlinear case, relaxation procedures have only pragmatic self-justification but can be extremely effective in some instances. In the problems of interest, optimum relaxation factors were determined empirically but were found to depend strongly on the "width" of the kernel. It appears that relaxation is useful primarily in the case of strong local coupling—an observation that is consistent with the theoretical analyses which can be carried out in the linear case. While the relaxation device can accelerate this convergence, the process remains a geometrically converging first-order iteration and proves to be inadequate for the class of problems under consideration.

To obtain a higher-order iteration, consider choosing H of (3.3) as the inverse of the Jacobian matrix of F with respect to z. In this way the well-known generalization of the Newton-Raphson process [9] is obtained. Due to the dual nature of the dependent variables in the problem, it is easiest to retreat to the analytic equation, to obtain a continuous analog of the Newton-Raphson process through a perturbation method, and then to discretize this as above; all linearization procedures are essentially equivalent. Symbolically, the analytic problem reads

$$\bar{z} = \mathcal{G}\bar{z}. \tag{3.5}$$

If it is assumed that the error in the current iterate $\bar{z}^l$ is "first-order" and the operator $\mathcal{G}$ is expanded, retaining only first-order terms, one can write

$$\vec{s}^{\,l+1} = \vec{s}^{\,l} + \delta\vec{s}^{\,l} = \mathcal{G}\vec{s}^{\,l} + \frac{\partial \mathcal{G}}{\partial \vec{s}}\,\delta\vec{s}^{\,l}. \tag{3.6}$$

The perturbation operator $\partial\mathcal{G}/\partial^q$ is linear and plays the role of a generalized derivative of $\mathcal{G}$; for the prototype problem (2.1),

$$\frac{\partial \mathcal{G}}{\partial \vec{s}}\,\delta f = \int_a^b dt \frac{\partial K}{\partial f}\,\delta f. \tag{3.7}$$

Define a first-order residual $\mathcal{R}^l$ by

$$\mathcal{R}^l = \mathcal{G}\vec{s}^{\,l} - \vec{s}^{\,l}. \tag{3.8}$$

This defines what we mean by "first-order." The following linear integral equation is then obtained for the correction $\delta\vec{s}^{\,l}$:

$$\delta\vec{s}^{\,l} = \frac{\partial \mathcal{G}}{\partial \vec{s}}\,\delta\vec{s}^{\,l} + \mathcal{R}^l. \tag{3.9}$$

Discretizing this equation as above, one obtains the asymptotically quadratically converging process:

$$\begin{aligned} \Delta z^l &= \frac{\partial G}{\partial z}\,\Delta z^l + r^l, \\ z^{l+1} &= z^l + \Delta z^l. \end{aligned} \tag{3.10}$$

For a problem characterized by a dimensionality of order 75, the solution of the linearized problem (3.9) is a substantial exercise in itself; one must form the perturbation matrix $\partial G/\partial z$ and then invert $(I - \partial G/\partial z)$. Unless one is close enough to the solution so that the asymptotic, quadratic convergence comes into play, this is not a very practical method for solving the problems of interest. Though one can economize somewhat by retaining the inverse of the "Jacobian" $(I - \partial G/\partial z)$ for several iterations, thereby reducing the rate of convergence but increasing the global efficiency, this approach is useful only for improving accurate approximations obtained by other means.

With higher-order processes, such as the generalized Newton-Raphson (or tangent) method, eliminated, one turns naturally to methods of higher degree, seeking to utilize the trend of recent iterates to accelerate the convergence. The simplest candidates for generalization from the univariate case are the secant method, or other forms of *regula falsi* [6], and such sequence transformations as the Aitken δ^2-process and the Wynn ϵ-algorithm [10]. From the multivariate linear case, one has the linear acceleration or semi-iterative methods of Varga [8] and others. In the former case, one is faced with ambiguity as to how to carry a one-dimensional process into N dimensions; in the latter, one lacks the powerful tools of linear algebra. It is not surprising, therefore, that there are a multitude of possibilities, with largely heuristic origins, depending on the particular aspect one wishes to generalize. For the problems of interest, we will simply point out some of the disadvantages of some algorithms that have been proposed, and obtain therefrom some desiderata for a new approach.

The asymptotic convergence properties of the basic iteration (3.4) are determined by the eigenvalues of the Jacobian matrix of G with respect to z at the root, in particular the dominant eigenvalue(s). In the simplest case of a single dominant eigen-

value and after a suitable number of iterations, the components of the error vector, which tends to lie in the direction of the dominant eigenvector, are effectively decoupled, in the sense that they converge geometrically and independently. The rate of convergence (or divergence) is determined by the magnitude of the dominant eigenvalue in a fashion analogous to the univariate case. *If* enough preparatory basic iterations are carried out, one might hope to obtain superlinear asymptotic convergence by applying univariate acceleration devices to the individual components z_i^l. Fox and Goodwin [11] consider such a process using the Aitken δ^2-process. Noble [1] considers the use of a generalized δ^2-process adapted to the case in which there is a complex conjugate pair of dominant eigenvalues. The secant method could also be applied; when used as an acceleration process, it is associated with the name of Wegstein [12]. Wynn [10] considers such a component-by-component process using the ϵ-algorithm. However, one cannot ignore the nonlinear coupling of the error components, and such acceleration procedures often lead to disastrous results especially in the case of strong local coupling. The situation is not notably improved by averaging over the components of the iterates in some fashion to obtain an "effective" geometic convergence factor. In the problems of interest, one cannot afford to do enough preparatory iterations to decouple the components of the error vector; hence component-by-component acceleration procedures are unsuitable.

The basic direct or inverse interpolation interpretation of the univariate secant method is easily generalized to the multivariate case. Geometrically, if linearization along a secant line to linearization over a set of secant hyperplanes is generalized, then direct and inverse interpolation are equivalent as in the univariate case. This approach (attributed to Gauss) is treated in Ostrowski [6], but the most concise form of the algorithm is that due to Wolfe [13]. Likewise, one can extend Jeeves' [14] approach to the univariate secant method and approximate the derivatives in the Jacobian matrix of the Newton-Raphson process by difference quotients [7]. Somewhat similar in spirit is the formal generalization of the Steffenson iteration (the Aitken δ^2-process applied dynamically) in Henrici [15]. All of these procedures share three disadvantages: (1) There are degeneracies which prevent the inversion of the $N \times N$ matrices involved and make them extremely ill-conditioned when not actually singular. (2) For N of order 75 the prospect of carrying out the inversion is itself a deterrent, as it was in the case of the Newton-Raphson process, even if possible in principle. (3) The methods are of degree $N+1$ or more; one simply cannot afford to obtain enough samples from the basic iteration even to begin to accelerate the convergence in this way. Thus again, this class of procedures is unsuitable for the problems of interest.

Wynn [10, 16, 17] has formally extended the ϵ-algorithm to vector sequences through the use of the Samuelson inverse for a vector:

$$z^{-1} = z/\| z \|^2. \tag{3.11}$$

Such a process couples the error components and can be used from the beginning of the iteration. Wynn has conducted a number of successful experiments with this process, including the iterative solution of systems of linear algebraic equations and nonlinear equations arising from the discretization of integral equations. In the author's limited experience, the vector ϵ-algorithm seems often to be a good tactic but a poor strategy. To reason by analogy with the univariate case, the problem

seems to be the "static" character of this sequence transformation—the transformed iterates are not "dynamically" fed back into the iterative process. The Aitken δ^2-process, of which the ϵ-algorithm is a generalization, is considerably less effective if applied statically as a sequence transformation than if applied dynamically as the Steffenson iteration [6, 7, 15, 18]. Similarly, the Wegstein process is often globally more efficient than the Steffenson iteration, to which it reduces by applying the acceleration only at every other step of the basic iteration. The dynamic feedback offsets the lower order of convergence. The vector ϵ-algorithm shares with the procedures to be described below the property of being useful in some problems and not in others, for reasons which are not apparent a priori.

In summary, a dynamic, low-degree, coupled iterative process is desired which accelerates the convergence of the basic iteration. In subsequent sections, a class of such processes is suggested on a heuristic basis. The basic idea is very simple and has many analogies with other devices which have been considered, but the closest direct analogy the author has discovered in the literature is a somewhat similar device for the linear case due to Khabaza [19]. In one variant or another, the procedure has proved useful in the problems outlined above, and in a number of other problems involving the iterative solution of finite sets of linear and nonlinear equations arising from integral, ordinary and partial differential equations. For eigenvalue problems, the device can be combined with the power method to some advantage. Experiments have been carried out with a large number of variants, but only the most useful are recorded here. No global convergence theorem seems likely though hints as to an asymptotic analysis are available. In any case, it is easy to find out empirically in any given problem whether or not this is a useful approach.

4. *Low-Degree Generalized Secant Methods*

A spectrum of ideas is involved here; we consider first a specific algorithm and then indicate how variants and generalizations can be obtained. The first algorithm was originally obtained by generalizing the univariate secant method (*regula falsi*) geometrically. Though it is most straightforward to proceed algebraically, it is, however, useful to note the geometric interpretation. Rather than generalizing the secant line through two sample points to a set of N secant hyperplanes through $N+1$ points—which yields Wolfe's method—we consider a hyperline through two sample points. This yields a dynamic, coupled, second-degree method. A hyperline does not in general intersect the subspace defining the solution—the analog of the line $y = 0$ or $y = x$, depending on whether the univariate equation $F(x) = 0$ or $x = f(x)$ are being considered. Consequently, the algorithm chooses that point on the hyperline which is in some sense "closest" to this subspace.

Begin with a basic iteration

$$z^{l+1} = Gz^l, \tag{4.1}$$

and define a coupled pair of iterative sequences x^l and y^l related by

$$y^l = Gx^l. \tag{4.2}$$

It is sought to define x^{l+1} as a function of y^l, x^l, y^{l-1} and x^{l-1} so that the sequences x^l and y^l converge more rapidly than the basic sequence z^l. We define x^{l+1} as a

linear, parameterized form in previous iterates, define a quadratic residual, and obtain the free parameters by minimizing this residual. Define a residual vector r^l by

$$r^l = y^l - x^l. \tag{4.3}$$

Define the inner product of two N-vectors u and v by

$$(u, v) = \sum_{i=1}^{N} u_i v_i w_i \tag{4.4}$$

where the weights w_i are positive. In some cases, it is convenient to generalize (4.4) in the usual way to include a nondiagonal, positive-definite metric and/or complex vector components. Define

$$\begin{aligned} u^l &= x^l + \theta^l(x^{l-1} - x^l), \\ v^l &= y^l + \theta^l(y^{l-1} - y^l), \end{aligned} \tag{4.5}$$

and

$$R^l = \tfrac{1}{2}(v^l - u^l, v^l - u^l). \tag{4.6}$$

Choose the free parameter θ^l so as to minimize the *linearized residual* R^l; this yields

$$\frac{\partial R^l}{\partial \theta^l} = (r^{l+1} - r^l, v^l - u^l) = 0. \tag{4.7}$$

Hence

$$\theta^l = (r^l, r^l - r^{l-1})/(r^l - r^{l-1}, r^l - r^{l-1}). \tag{4.8}$$

Define

$$x^{l+1} = u^l + \beta^l(v^l - u^l) \qquad (\beta^l > 0). \tag{4.9}$$

We refer to the algorithm so defined as the *extrapolation algorithm*. Usually the choice $\beta^l = 1$ is most appropriate; the choice $\beta^l = 0$ is excluded for reasons to be discussed below. The extrapolation algorithm resembles, in some respects, various search or descent methods for root finding [18]; indeed, a "least-squares solution" can be found if no true solution z exists. As in such methods, it is often useful to "under-relax" the iteration by choosing $0 < \beta^l < 1$, but the optimum β^l must be determined empirically.

The extrapolation algorithm reduces to the univariate secant method for $N = 1$; it is interesting to note that it can be obtained formally from the secant method by introducing the Samuelson inverse of (3.11). Since the vector algebra using inner products and the Samuelson inverse is not associative, an alternate algorithm can be formally obtained in the same way. We refer to this alternate algorithm as the *relaxation algorithm*, since it corresponds to defining a relaxation parameter dynamically, but we prefer to approach it algebraically as follows. Define

$$\begin{aligned} \bar{u}^l &= y^l + \eta^l r^l, \\ \bar{v}^l &= y^{l-1} + \eta^l r^{l-1}, \end{aligned} \tag{4.10}$$

and

$$\bar{R}^l = \tfrac{1}{2}(\bar{v}^l - \bar{u}^l, \bar{v}^l - \bar{u}^l). \tag{4.11}$$

Choose η^l to minimize $\bar{R}^l$; this yields

$$\frac{\partial \bar{R}^l}{\partial \eta^l} = (r^{l-1} - r^l, \bar{v}^l - \bar{u}^l) = 0. \tag{4.12}$$

Hence

$$\eta^l = (y^l - y^{l-1}, r^l - r^{l-1})/(r^l - r^{l-1}, r^l - r^{l-1}). \tag{4.13}$$

Then define

$$x^{l+1} = \bar{u}^l; \tag{4.14}$$

in this "cross-sequence" interpolation, no additional averaging seems appropriate.

Just as one can obtain the Aitken δ^2-process, applied dynamically as the Steffenson iteration, in the univariate case by applying the secant method alternately with basic iteration steps, one can also obtain a generalized δ^2-process in the same fashion. The relaxation and extrapolation ($\beta^l = 1$) algorithms are identical when applied in this fashion and yield a single generalized δ^2-process. This process is distinct from the generalized δ^2-process obtained by carrying the Wynn vector ϵ-algorithm through one stage. However, as in the univariate case, it is usually a more effective strategy to accelerate at each step with the secant process rather than to use a generalized δ^2-process.

Variants of these algorithms can be obtained in many ways; two classes of variants have proved of particular interest. The first class concerns the choice of the metric of the inner product defining the "linearized residual" R^l. Usually, it is advantageous to define the residual in a relative sense by choosing w_i inversely proportional to some measure of the local "size" of the solution vector—either a priori or dynamically as the iteration proceeds. This is of particular significance when solving coupled sets of integral equations where one lumps subvectors of somewhat different character together to form an iteration vector. Alternately, one can accelerate the subvectors individually, ignoring the coupling between them; when applied to subvectors of dimension one, the process degenerates to the component-by-component acceleration processes discussed above. In multidimensional problems where an array of dependent variables is reduced to a vector iterate, an "alternating direction" [8] redefinition of the subvectors to be accelerated at consecutive iterations seems useful. A variant which is sometimes useful with the extrapolation algorithm, but not the relaxation algorithm, is the choice of a metric proportional to the local residual r_i^l. Attention is thereby focussed on the "worst" components at each stage.

The extrapolation algorithm can easily be generalized, but it does not seem useful to extend the relaxation algorithm in similar fashion due to its "cross-sequence" character. The extrapolation algorithm was obtained above by minimizing a certain linearized residual over a hyperline; higher degree methods are obtained by minimizing over linear subspaces of higher dimension. Define

$$\begin{aligned} u^l &= x^l + \sum_{j=1}^{M} \theta_j^l (x^{l-j} - x^l), \\ v^l &= y^l + \sum_{j=1}^{M} \theta_j^l (y^{l-j} - y^l), \end{aligned} \tag{4.15}$$

and

$$R^l = \tfrac{1}{2}(v^l - u^l, v^l - u^l). \tag{4.16}$$

Choose θ_j^l so as to minimize R^l; this yields

$$\frac{\partial R^l}{\partial \theta_i^l} = (r^{l-i} - r^l, v^l - u^l) = 0. \tag{4.17}$$

Hence

$$\sum_{j=1}^{M} (r^l - r^{l-i}, r^l - r^{l-j})\theta_j^l = (r^l, r^l - r^{l-i}) \tag{4.18}$$

for $i = 1, 2, \cdots, M$. Again x^{l+1} is defined by (4.9).

Provided that the vectors r^{l-i} are linearily independent, the residual R^l can be brought to zero for $M = N$; in this case, the extrapolation algorithm becomes a variant of Wolfe's method. However, the extrapolation algorithm is dynamic, coupled and can be used from the beginning of the iteration; the degree can be allowed to increase up to $N+1$ as the iteration proceeds. In practice, the low-degree cases with $M = 1, 2, \cdots$ are the most useful, and it seems best to limit M to about 5. By analogy with higher-degree, multipoint iterative processes in the univariate case, one might say that the power of an iterative method increases slowly with degree for $M > 3$ since the "early," poor approximations are not samples of significant information content for the high-degree methods which retain them. It is of interest to consider the relationship between the present algorithm and Wolfe's method in the light of Tornheim's convergence theorem for the latter [20]. Since R^l is brought to zero at each stage, Wolfe's method converges superlinearily—*if* it converges. The Wolfe process fails to converge if the set of previous iterates fails to span the space containing the solution—the linear equations involved become singular. Furthermore, as the iteration proceeds, the iterates must tend to become more and more alike, the degeneracy becomes more and more of a problem, and the linear system becomes extremely ill-conditioned, if not actually singular. The low-degree extrapolation algorithm can be regarded as a device to alleviate this difficulty, at least in part, since the degree of ill-conditioning and potentiality for degeneracy is at least proportional to the degree of the iterative process. For the same reason, the restriction $\beta^l > 0$ in the extrapolation algorithm is required in order that the new iterate x^{l+1} not become trapped in the subspace spanned by the previous x^l iterates; the admixture of y^l iterates as each stage introduces "new dimensions" for the subspace to be searched. The potentiality for degeneracy is not, however, eliminated in the extrapolation algorithm, only reduced. Especially near the end of the iteration; the linear system of dimension M will become ill-conditioned or even singular. It is convenient to make the convention that M will be automatically decreased to suppress redundant previous iterates until the degeneracy of the linear system is lifted. This can be accomplished in the routine used to solve the system of M linear equations. The matrix of coefficients is positive definite if there is no redundancy. One needs simply to redefine M as the dimension of the largest positive-definite upper left principal minor in the case of redundancy. Not only do the vectors $r^{l-i} - r^l$ tend to become linearily dependent, but they also tend individually to zero; at some point, there will not be enough significant figures left in these residuals to define θ_i^l, and the linear acceleration method ceases to be useful. At this stage, a Newton-Raphson iteration with the inverse of the Jacobian matrix evaluated only once can be used to great advantage.

If the system of N equations considered happens to be linear, Wolfe's method converges immediately, if it converges. The low-degree extrapolation algorithm will

in general require several iterations—typically less than N for any practical convergence criterion and reasonable first iterate. It is easy to verify that, in this case, the linearized residual R^l is an actual residual for the problem—the residual vector corresponding to a linear combination of iterates is the corresponding linear combination of the individual residual vectors. If the basic iteration converges, it appears that R^l decreases monotonically to zero; the accelerated iteration converges at least as rapidly and in general more rapidly than the basic iteration. If the basic iteration diverges, R^l does not necessarily converge monotonically to zero, but with a good enough initial iterate convergence is possible, indeed probable.

Khabaza [19] has considered an iterative procedure for the solution of large, sparse systems of linear equations based on the idea of determining at each stage a "least-squares, matrix polynomial approximation" to the inverse matrix. It can easily be shown, with suitable redefinition of terminology, that this procedure is equivalent to carrying out M steps of a Jacobi basic iteration [8], and then applying the extrapolation algorithm to the resulting "samples"—with $\beta^l = 0$. (This choice of β^l is acceptable here since "fresh samples" are obtained from the M basic iteration steps at each stage.) Khabaza discusses a number of examples in which the process compares favorably with standard procedures such as successive over-relaxation [8]; even better results might be anticipated from a more dynamic process. Again, low-degree processes ($M \sim 3$) prove most effective.

An asymptotic convergence theorem can be based on a linearization about the final solution analogous to that considered above in the perturbation method and these observations about the linear problem. It is for this reason that the quantity R^l of the extrapolation algorithm has been termed a "linearized residual." Again, *if* the process converges one might anticipate superlinear convergence. One important property of the superlinear secant process which this generalization retains, at least in some problems, is that of converting a divergent basic successive substitution iteration into a convergent accelerated iteration. In view of the fact that "naturally occurring" nonlinear integral equations often do not yield a convergent successive substitution iteration, this is a highly desirable property. These comments are illustrated in the next section by several highly simplified and somewhat atypical examples; however, similar results have been obtained in more realistic problems.

One final point of some importance should be made. At each stage, we minimize R^l over a linear subspace; the efficacy of the procedure depends strongly on having the right "shaped" subspaces. Inaccurate initial iterates of the wrong "shape" will affect later iterates due to the "memory" built into the algorithm. Since the subspaces searched are linear, the "shape" is altered only through the admixture of y^l vectors from the basic iteration. In the original integral equation problems, the iterates have an internal structure whose general character is known a priori, initial iterates of the right form are available, and the algorithm proves to be quite effective.

5. *Examples*

The results obtained by applying various of the algorithms outlined above to three simple examples will now be described briefly. These results illustrate the character of those obtained in more complicated problems.

The first example is a highly degenerate, linear problem constructed for ease in evaluating variants and generalizations of the low-degree, generalized secant methods described above. Consider the system of N linear equations:

$$Az = Db \tag{5.1}$$

with

$$\begin{aligned} A_{ij} &= D && (i = j) \\ &= 1 && (i \neq j) \end{aligned} \tag{5.2}$$

with D a free parameter. We choose b such that the solution z has the form

$$z_i = 2/i, \tag{5.3}$$

and rewrite the equation in the form (3.2) so that the basic iteration is a Jacobi iteration [8]:

$$z = Hz + b \tag{5.4}$$

with

$$\begin{aligned} H_{ij} &= 0 && (i = j) \\ &= -1/D && (i \neq j). \end{aligned} \tag{5.5}$$

Table I contains the results of applying several of the iterative procedures outlined above, in units of the computation involved in a single basic iteration. The computation involved in the acceleration processes themselves is ignored, since it is negligible in any realistic, nonlinear problem. If the initial iterate is chosen as e ($e_i \equiv 1$), all iterates are confined to the two-space spanned by z and e. The dimension N is essentially nugatory, except in so far as the relationship between N and D determines the convergence properties of the basic iteration. For $N = 20$, $D = 25$ yields a convergent basic iteration and $D = 15$ yields a divergent basic iteration. Due to the essentially two-dimensional character of this linear problem, the $M = 2$ extrapolation algorithm and the Wynn vector ϵ-algorithm are exact, terminating iterative processes. The dynamic δ^2-process obtained by carrying the Wynn vector ϵ-algorithm through one stage does not terminate. The use of a metric proportional to the local residual improves the $M = 1$ extrapolation algorithm, but a unit metric ($w_i \equiv 1$) is more appropriate for $M > 1$ and for the relaxation algorithm. All of the acceleration processes considered are adequate to overcome the weakly divergent basic iteration for $D = 15$ in this linear problem.

TABLE I

Algorithm	$D = 25$	$D = 15$
Basic Iteration	67	∞
Extrapolation ($M = 1$), unit metric	7	10
Extrapolation ($M = 1$), residual metric	5	5
Relaxation, unit metric	7	7
Extrapolation ($M = 2$), unit metric	3	3
Wynn Dynamic δ^2-Process	10	14
Wynn Vector ϵ-Algorithm	6	6

A very simple nonlinear example is provided by the eigenvalue problem for the same matrix: $Az = \lambda z$. A conventional basic iteration for the dominant eigenvalue and eigenvector of such a matrix is the power method. One forms a sequence of eigenvector iterates according to

$$z^{l+1} = Az^l \tag{5.6}$$

and a sequence of eigenvalue iterates according to

$$\lambda^{l+1} = (z^{l+1}, z^{l+1})/(z^{l+1}, z^l), \tag{5.7}$$

using the conventional Euclidean inner product. This can be reformulated as follows: We define a nonlinear operator α and a coupled pair of iterative sequences by

$$y^l = \alpha x^l = \{\lambda^{l+1}\}^{-1} Ax^l = \left\{\frac{(Ax^l, Ax^l)}{(Ax^l, x^l)}\right\}^{-1} Ax^l. \tag{5.8}$$

Since the problem is homogeneous, we normalize the iterates and define a basic iteration by

$$x^{l+1} = y^l/(y^l, y^l)^{1/2}. \tag{5.9}$$

This basic iteration is just the power method written in a form suitable for the application of our acceleration processes. Note that in the power method, λ^l plays no essential role in defining the new $(l+1)$-th iterate; consequently, an acceleration process like the Aitken δ^2-process applied to the eigenvalue sequence is, perforce, a static transformation. On the other hand, if a vector acceleration scheme is applied to the eigenvector sequence we obtain a dynamic process; furthermore, we obtain eigenvectors which are essentially as accurate as the eigenvalues, which is not the case in the conventional power method.

For the matrix A defined above, the dominant eigenvalue is $D + N - 1$ $(D > 0)$ and the corresponding eigenvector is e. The results for $D = 101$, $N = 20$, and a unit metric are given in Table II, in units of the equivalent number of basic iterations. Again this is a rather degenerate problem; the addition of the normalization condition restricts the iterates to a hypercircle of radius unity in the two-space spanned by the initial iterate $(y_i^{-1} = 2/i)$ and the solution. Consequently the $M = 1$ extrapolation algorithm is essentially equivalent to Wolfe's method, in the sense that the residual R^l can be brought to zero at each stage. The behavior of the acceleration schemes in this example is indicative of that to be expected in a geometrically-converging, first-order iterative process. The behavior of a component-by-component (or component-averaged) secant or Aitken δ^2-acceleration would be comparable. The generalized δ^2-process considered in this example is that obtained from the extrapolation and/or relaxation algorithms; note the efficacy of a fully dynamic approach.

The third, somewhat more realistic, example involves the solution of two nonlinear integral equations:

$$f^2(x) = \frac{3\sqrt{2}\pi}{16} \int_{-1}^{1} dt\, f(t) \cos^2 \frac{\pi\,|\,x - t\,|}{4} - \frac{1}{4} \tag{5.10}$$

and

$$f(x) = \frac{3\sqrt{2}\pi}{16} \int_{-1}^{1} dt\, f^2(t) \cos \frac{\pi\,|\,x - t\,|}{4} - \frac{1}{4} \cos \frac{\pi x}{4}. \tag{5.11}$$

TABLE II

Algorithm	*Iterations*
Basic Iteration	36
Extrapolation ($M = 1$)	6
Relaxation	6
Generalized δ^2-Process	8

TABLE III

Algorithm	*Equation* (5.10)		*Equation* (5.11) $f^0 = 1$
	$f^0 = 1$	$^0 = 1 + x/2$	
Basic Iteration	26	26	∞
Extrapolation ($M = 1$)	6	11	6
Relaxation	7	10	6
Extrapolation ($M = 2$)	5	6	8
Wynn Dynamic δ^2-Process	14	18	10
Wynn Vector ϵ-Algorithm	8	12	∞

Both of these equations have the solution

$$f(x) = \cos \frac{\pi x}{4} \tag{5.12}$$

and were discretized as described in Section 2. The basic successive substitution iteration is convergent for (5.10) but strongly divergent for (5.11). The results for various acceleration schemes are given in Table III, in units of the basic iteration. In this problem, the residual metric is ineffective, since the error is relatively uniformly distributed, and the unit metric yields the same results. First, the two equations were run with initial iterate $f_i^0 \equiv 1$; then, equation (5.10) was rerun with initial iterate $f_i^0 = 1 + x_i/2$. The "shape" dependence of the acceleration schemes is shown very clearly; while the geometric convergence of the basic iteration is unaffected by the initial iterate, the acceleration schemes are affected through their "memory." The static Wynn vector ϵ-algorithm cannot cope with the divergence of the basic iteration with (5.11), and even the dynamic δ^2-process derived from the vector ϵ-algorithm is better.

While these examples are over-simplified, the character of the results is typical of that which has been encountered in more complex problems. In all experiments conducted thus far, the accelerated iteration is more rapidly convergent than the basic iteration.

6. *Conclusion*

A class of iterative procedures for the numerical solution of systems of nonlinear equations has been described. In the context of the numerical solution of a class of nonlinear integral equations, these procedures possess several advantages over other approaches which have been considered. As devices for accelerating the convergence of vector sequences, these procedures are potentially useful in a much larger context.

RECEIVED APRIL, 1965

REFERENCES

1. NOBLE, B. The numerical solution of non-linear integral equations. *Non-Linear Integral Equations*, P. M. Anselone (Ed.), U. of Wisconsin Press, Madison, 1964.
2. ANDERSON, D. G. Numerical experiments in kinetic theory. Doctoral Thesis, Harvard U., Cambridge, Mass., June 1963.
3. ——. Numerical solutions of the Krook kinetic equation. Tech. Rep. No. 15, Engineering Sciences Lab., Harvard U., Feb. 1965. In preparation.
4. —— AND MACOMBER, H. K. Numerical experiments in kinetic theory. *Proceedings of the Fourth International Symposium on Rarefied Gas Dynamics*, Academic Press, New York, 1965.
5. LANCZOS, C. *Tables of Chebyshev Polynomials*. NBS Appl. Math. Series No. 9, Government Printing Off., Washington, D. C., 1952.
6. OSTROWSKI, A. M. *Solution of Equations and Systems of Equations*. Academic Press, New York, 1960.
7. TRAUB, J. F. *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs, N. J., 1964.
8. VARGA, R. S. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N. J., 1962.
9. ALTMAN, M. A generalization of Newton's method. *Bull. Acad. Pol. Sci.* 3 (1955), 189.
10. WYNN, P. Acceleration techniques for iterated vector and matrix problems. *Math. Comput. 16* (1962), 301.
11. FOX, L., AND GOODWIN, E. T. The numerical solution of non-singular integral equations. *Phil. Trans. 245A* (1953), 501.
12. WEGSTEIN, J. H. Accelerating convergence of iterative processes. *Comm. ACM 1* (June 1958), 9.
13. WOLFE, P. The secant method for simultaneous non-linear equations. *Comm. ACM 2* (Dec. 1959), 12.
14. JEEVES, T. A. Secant modification of Newton's method. *Comm. ACM 1* (Aug. 1958), 9.
15. HENRICI, P. *Elements of Numerical Analysis*. John Wiley & Sons, New York, 1964.
16. WYNN, P. Acceleration techniques in numerical analysis. *Information Processing 1962*, C. M. Popplewell (Ed.), North Holland, Amsterdam, (1963).
17. ——. General purpose vector epsilon algorithm ALGOL procedures. *Numer. Math. 6* (1964), 22.
18. HOUSEHOLDER, A. S. *Principles of Numerical Analysis*. McGraw-Hill, New York, 1953.
19. KHABAZA, I. M. An iterative least-square method for solving large sparse matrices. *Comput. J. 6* (1963–64), 202.
20. TORNHEIM, L. Convergence of multipoint iterative methods. *J. ACM 11* (1964), 210.