70-23,532

IYER, Ramdas Ramnath, 1937-
GENERAL TECHNIQUE FOR SOLVING NONLINEAR, TWO-POINT BOUNDARY-VALUE PROBLEMS VIA THE METHOD OF PARTICULAR SOLUTIONS.

Rice University, Ph.D., 1970
Engineering, aeronautical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

RICE UNIVERSITY

GENERAL TECHNIQUE FOR SOLVING NONLINEAR, TWO-POINT BOUNDARY-VALUE PROBLEMS VIA THE METHOD OF PARTICULAR SOLUTIONS

by

R.R. IYER

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Thesis Director's signature

Angelo Miele

Houston, Texas

May, 1970

Abstract

GENERAL TECHNIQUE FOR SOLVING NONLINEAR, TWO-POINT BOUNDARY-VALUE PROBLEMS VIA THE METHOD OF PARTICULAR SOLUTIONS

by

R.R. IYER

In this thesis, a general technique for solving nonlinear, two-point boundary-value problems is presented; it is assumed that the differential system has order n and is subject to p initial conditions and q final conditions, where $p + q = n$. First, the differential equations and the boundary conditions are linearized about a nominal function $x(t)$ satisfying the p initial conditions. Next, the linearized system is imbedded into a more general system by means of a scaling factor α, $0 \le \alpha \le 1$, applied to each forcing term. Then, the method of particular solutions is employed in order to obtain the perturbation $\Delta x(t) = \alpha A(t)$ leading from the nominal function $x(t)$ to the varied function $\tilde{x}(t)$; this method differs from the adjoint method and the complementary function method in that it employs only one differential system, namely, the nonhomogeneous, linearized system.

The scaling factor (or stepsize) α is determined by a one-dimensional search starting from $\alpha = 1$ so as to ensure the decrease of the performance index P (the cumulative error in the differential equations and the boundary

conditions). It is shown that the performance index has a descent property; therefore, if α is sufficiently small, it is guaranteed that $\tilde{P} < P$. Convergence to the desired solution is achieved when the inequality $\tilde{P} \leq \epsilon$ is met, where ϵ is a small, preselected number.

Computationally, the present technique can be employed in two ways: (a) the function $x(t)$ is updated according to $\tilde{x}(t) = x(t) + \alpha A(t)$; or (b) the initial point $x(0)$ is updated according to $\tilde{x}(0) = x(0) + \alpha A(0)$, and the new nominal function $\tilde{x}(t)$ is obtained by forward integration of the nonlinear differential system. In this connection, five numerical examples are presented; they illustrate (i) the simplicity as well as the rapidity of convergence of the algorithm, (ii) the importance of stepsize control, and (iii) the desirability of updating the function $x(t)$ according to Scheme (a).

ACKNOWLEDGEMENTS

The author is indebted to Professor Angelo Miele for suggesting the topic and stimulating discussions. Also, he wishes to acknowledge the support of the National Science Foundation, Grant No. GP-18522.

TABLE OF CONTENTS

1. Introduction

In recent years, considerable attention has been devoted to the solution of the two-point boundary-value problem for nonhomogeneous, linear differential systems. Among the techniques available, we mention (a) the method of adjoint variables and (b) the method of complementary functions (Ref. 1). Methods (a) and (b) have one common characteristic: each requires the solution of two differential systems, namely, the original system plus the derived system; this derived system is the adjoint system in Case (a) and the homogeneous system in Case (b).

With particular regard to high-speed digital computing, programming can be made simpler if one employs the original system only. This technique, a modification of (b), consists of combining linearly several particular solutions of the original, nonhomogeneous system. For this reason, it has been called the method of particular solutions (Ref. 2). It has the following advantages with respect to the previous techniques: (α) it makes use of only one differential system, namely, the original, nonhomogeneous system; (β) each particular solution satisfies the same prescribed initial conditions; and (γ) in a physical problem, each particular solution represents a physically possible trajectory, even though it satisfies only the initial conditions and not the final conditions.

While the method of particular solutions has been developed for linear systems, it can also be used to solve nonlinear systems. First, quasilinearization must be employed, and the nonlinear system must be replaced by one that is linear in the perturbation about a nominal function (see, for example, Refs. 3-6); to this linear system, the method of particular solutions can be applied to find the perturbation leading to a new nominal function; then, the procedure is employed iteratively.

Recently, Heideman illustrated the above procedure through several particular nonlinear examples (Refs. 7-9). Here, a general technique for the solution of the two-point boundary-value problem associated with a nonlinear differential system of order n subject to p initial conditions and q final conditions is presented. Note that $p + q = n$. As a guide during progression of the algorithm, we employ the performance index P already introduced in Ref. 10: this is the cumulative error in the differential equations and the boundary conditions and can be proved to have a descent property. In addition, the performance index is also employed as a convergence criterion: the algorithm is terminated when the performance index becomes smaller than some preselected value.

2. Linear System

In this section, we review the main results of Ref. 2. Consider the linear differential system[1]

$$\dot{x} = M(t)x + N(t) \quad , \quad 0 \le t \le 1 \tag{1}$$

where t is the time, x is the state, and M and N are piecewise continuous functions of the time. Here, x is n x 1, M is n x n, and N is n x 1. At the initial time t = 0, p scalar components of x are prescribed, that is,

$$x^i(0) = \beta^i \quad , \quad i=1,2,\ldots,p \tag{2}$$

where the scalar quantities β^i are given. At the final time t = 1, q scalar relations must be satisfied; in matrix form, this can be written as[2]

$$Cx(1) = \gamma \tag{3}$$

where C and γ are given. Here, C is q x n and γ is q x 1. The problem is to find the function x(t) which solves Eq. (1) subject to (2)-(3).

In order to solve the proposed problem, the method of particular solutions (Ref. 2) is employed. Specifically, let

$$x_j = x_j(t) \quad , \quad j=1,2,\ldots,q+1 \tag{4}$$

---

[1] The assumption t = 0 concerning the initial time and the assumption t = 1 concerning the final time can be made without loss of generality. Indeed, a problem where the actual running time θ has the lower limit a and the upper limit b can be reduced to the present form by introducing the transformed time $t = (\theta-a)/(b-a)$.

[2] By assumption, p + q = n.

denote q + 1 particular solutions obtained by forward integration of Eq. (1) subject to the initial conditions

$$x_j^i(0) = \beta^i \ , \quad j=1,2,\ldots,q+1 \ , \quad i=1,2,\ldots,p$$
$$x_j^{p+k}(0) = \delta_{jk} \ , \quad j=1,2,\ldots,q+1 \ , \quad k=1,2,\ldots,q \tag{5}$$

where δ_{jk} denotes the Kronecker delta. Next, introduce the q+1 undetermined, scalar constants k_j and form the linear combination

$$x(t) = \sum_{j=1}^{q+1} k_j x_j(t) \tag{6}$$

By simple substitution, it can be verified that this linear combination satisfies the differential equation (1) and the initial conditions (2) providing

$$\sum_{j=1}^{q+1} k_j = 1 \tag{7}$$

It also satisfies the final condition (3) providing

$$C \sum_{j=1}^{q+1} k_j x_j(1) = \gamma \tag{8}$$

Equations (7)-(8) constitute a system of q + 1 scalar relations which are linear in the q + 1 constants k_j. After the constants k_j are known, the solution x(t) of the linear, two-point boundary-value problem is obtained from (6).

3. Nonlinear System

Consider the nonlinear differential system[3]

$$\dot{x} = \varphi(x,t) \quad , \quad 0 \le t \le 1 \tag{9}$$

where t is the time, x is the state, and φ is a continuous function of the arguments x and t. Here, x is n x 1 and φ is n x 1. At the initial time $t = 0$, p scalar components of x are prescribed, that is,

$$x^i(0) = \beta^i \quad , \quad i=1,2,\ldots,p \tag{10}$$

where the scalar quantities β^i are given. At the final time $t = 1$, q scalar relations must be satisfied; in matrix form, this can be written as[4]

$$\psi(x(1)) = 0 \tag{11}$$

where ψ is a q x 1 continuous function of x evaluated at $t = 1$. The problem is to find the function x(t) which solves Eq. (9) subject to (10)-(11).

In order to solve the proposed problem, consider a nominal function x(t) satisfying the initial conditions (10) exactly, but not necessarily Eqs. (9) and (11). Let

$$\tilde{x}(t) = x(t) + \Delta x(t) \tag{12}$$

denote a varied function satisfying the initial conditions (10) exactly and Eqs. (9)-(11) to first order. If quasilinearization is applied, we obtain the following differential system:

---

[3] See Footnote 1.

[4] See Footnote 2.

$$\Delta\dot{x} - \varphi_x^T(x,t)\Delta x + [\dot{x} - \varphi(x,t)] = 0 \quad , \quad 0 \le t \le 1$$

$$\Delta x^i(0) = 0 \quad , \quad i=1,2,\ldots,p \tag{13}$$

$$\psi_x^T(x(1))\Delta x(1) + \psi(x(1)) = 0$$

where φ_x is n x n and ψ_x is n x q.[5]

For convenience, we imbed this differential system into the more general system

$$\Delta\dot{x} - \varphi_x^T(x,t)\Delta x + \alpha[\dot{x} - \varphi(x,t)] = 0 \quad , \quad 0 \le t \le 1$$

$$\Delta x^i(0) = 0 \quad , \quad i=1,2,\ldots,p \tag{14}$$

$$\psi_x^T(x(1))\Delta x(1) + \alpha\psi(x(1)) = 0$$

where α denotes a scaling factor (or stepsize) in the range

$$0 \le \alpha \le 1 \tag{15}$$

Next, we introduce the auxiliary variable

$$A = \Delta x/\alpha \tag{16}$$

[5] The matrix φ_x is defined so that its ith column is the gradient of the ith component of φ with respect to the vector x. An analogous remark holds for the matrix ψ_x. The symbol T denotes transposition of a matrix.

and rewrite Eqs. (14) in the form

$$\dot{A} - \varphi_x^T(x,t)A + [\dot{x} - \varphi(x,t)] = 0 \quad , \quad 0 \leq t \leq 1$$

$$A^i(0) = 0 \quad , \quad i=1,2,\ldots,p \tag{17}$$

$$\psi_x^T(x(1))A(1) + \psi(x(1)) = 0$$

For a given nominal function x(t), the vector $\dot{x} - \varphi$ and the matrix φ_x are known functions of the time t; also, the vector ψ_x are known quantities at $t = 1$. This being the case, the system (17) is reduced to the form (1)-(3).

We now apply the method of particular solutions. Let

$$A_j = A_j(t) \quad , \quad j=1,2,\ldots,q+1 \tag{18}$$

denote q + 1 particular solutions obtained by forward integration of Eq. (17-1) subject to the initial conditions

$$A_j^i(0) = 0 \quad , \quad j=1,2,\ldots,q+1 \quad , \quad i=1,2,\ldots,p$$
$$A_j^{p+k}(0) = \delta_{jk} \quad , \quad j=1,2,\ldots,q+1 \quad , \quad k=1,2,\ldots,q \tag{19}$$

where δ_{jk} denotes the Kronecker delta. After introducing the q + 1 undetermined, scalar constants k_j, we form the linear combination

$$A(t) = \sum_{j=1}^{q+1} k_j A_j(t) \tag{20}$$

Because of the results of Section 2, this linear combination satisfies the system (17) providing

$$\sum_{j=1}^{q+1} k_j = 1 \quad , \quad \psi_x^T(x(1)) \sum_{j=1}^{q+1} k_j A_j(1) + \psi(x(1)) = 0 \tag{21}$$

Equations (21) constitute a system of q + 1 scalar relations which are linear in the q + 1 constants k_j. After the constants k_j are known, the solution A(t) of the linearized, two-point boundary-value problem is obtained from (20). With A(t) known and α specified, the perturbation $\Delta x(t)$ is computed from (16); then, the varied function $\tilde{x}(t)$ is computed from (12).

3.1. Performance Index. Here, we define the scalar performance index

$$P = \int_0^1 [\dot{x} - \varphi(x,t)]^T [\dot{x} - \varphi(x,t)] dt + \psi^T(x(1))\psi(x(1)) \tag{22}$$

Clearly, P = 0 if x(t) satisfies Eqs. (9) and (11), and P > 0 otherwise. Since P measures the cumulative error in the differential system, one can use it as a guide during progression of the algorithm as well as to establish convergence. In this connection, the following convergence criterion arises from the above definition:

$$P \leq \epsilon \tag{23}$$

where ϵ is a small, preselected number.

3.2. Descent Property. The first variation of the performance index is given by

$$\delta P = 2\int_0^1 [\dot{x} - \varphi(x,t)]^T [\Delta\dot{x} - \varphi_x^T(x,t)\Delta x] dt + 2\psi^T(x(1))\psi_x^T(x(1))\Delta x(1) \tag{24}$$

and, because of Eqs. (14-1) and (14-3), reduces to

$$\delta P = -2\alpha\int_0^1 [\dot{x} - \varphi(x,t)]^T[\dot{x} - \varphi(x,t)]dt - 2\alpha\psi^T(x(1))\psi(x(1)) \tag{25}$$

which, in the light of (22), becomes

$$\delta P = -2\alpha P \tag{26}$$

Since $P > 0$, Eq. (26) shows that the first variation of the performance index is negative for $\alpha > 0$. Therefore, if α is sufficiently small, the decrease of the performance index is guaranteed, that is,

$$\tilde{P} < P \tag{27}$$

3.3. Stepsize. After the linearized, two-point boundary-value problem is solved for any given iteration, the function $A(t)$ is available. With this function, one can form the one-parameter family of solutions

$$\tilde{x}(t) = x(t) + \alpha A(t) \tag{28}$$

and explore the behavior of the performance index with respect to the parameter α. For the family (28), the performance index (22) becomes a function of the form

$$P = P(\alpha) \tag{29}$$

At $\alpha = 0$, the slope of the function (29) is negative and is given by

$$P_\alpha(0) = -2P(0) \tag{30}$$

Assuming that a minimum of $P(\alpha)$ exists, one can perform a one-dimensional search (using quadratic interpolation, cubic interpolation, or quasilinearization) to determine the optimum value of the stepsize, that is, the value for which

$$P_\alpha(\alpha) = 0 \tag{31}$$

Ideally, this procedure should be used iteratively until the modulus of the slope satisfies the following inequality:

$$|P_\alpha(\alpha)| \leq \eta \tag{32}$$

where η is a small, preselcted number.

In practice, the rigorous determination of α takes time on a computer. Therefore, one might renounce solving Eq. (13) with a particular degree of precision and determine the stepsize in a noniterative fashion. To this effect, we first assign the value

$$\alpha = 1 \tag{33}$$

to the stepsize; this corresponds to full quasilinearization of Eqs. (9)-(11) and is the value which would solve Eq. (31) exactly, should the differential equation (9) and the boundary condition (11) be linear. Of course, the stepsize (33) is acceptable only if

$$P(\alpha) < P(0) \tag{34}$$

Otherwise, the previous value of α must be replaced by some smaller value in the range (15) (for example, with a bisection process) until Ineq. (34) is met. This is guaranteed by the descent property of Section 3.2.

3.4. Summary of the Algorithm. In the light of the previous discussion, we summarize the algorithm as follows:

(a) Assume a nominal function $x(t)$ consistent with the initial conditions (10) and, possibly, with the final conditions (11).

(b) For the nominal function, compute the vector $\dot{x} - \varphi$ and the matrix φ_x as functions of the time t; at the final time $t = 1$, compute the vector ψ and the matrix ψ_x.

(c) Determine the q+l particular solutions $A_j(t)$ by forward integration of the differential equation (17-1) subject to the initial conditions (19).

(d) Compute the q+l constants k_j by solving the linear algebraic system (21).

(e) Determine the function $A(t)$ with (20).

(f) Consider the one-parameter family of solutions (28), and determine the stepsize α in such a way that Ineq. (34) is satisfied; to this effect, perform a bisection process on α starting from $\alpha = 1$.

(g) Once the stepsize is known, compute the varied function $\tilde{x}(t)$ from (28).

(h) With $\tilde{x}(t)$ known, the iteration is completed; the varied function $\tilde{x}(t)$ becomes the nominal function $x(t)$ for the next iteration; that is, return to (a), and iterate the algorithm.

(i) The algorithm is terminated when the stopping condition (23) is satisfied.

3.5. Alternate Scheme. As an alternative to the previous algorithm, the following scheme can be employed:

(a) Assume a nominal initial point $x(0)$ consistent with the prescribed initial conditions (10); integrate Eq. (9) forward to obtain the nominal function $x(t)$, which is such that $\dot{x} - \varphi = 0$ for $0 \le t \le 1$.

(b) For the nominal function, compute the matrix φ_x as a function of the time t; at the final time t = 1, compute the vector ψ and the matrix ψ_x.

(c) Determine the q + 1 particular solutions $A_j(t)$ by forward integration of the differential equation (17-1) subject to the initial conditions (19).

(d) Compute the q + 1 constants k_j by solving the linear algebraic system (21).

(e) Determine the initial value A(0) from (20) applied at t = 0.

(f) Consider the one-parameter family of initial points

$$\tilde{x}(0) = x(0) + \alpha A(0) \tag{35}$$

and integrate Eq. (9) forward subject to (35); determine the stepsize α in such a way that Ineq. (34) is satisfied; to this effect, perform a bisection process on α starting from $\alpha = 1$.

(g) Once the stepsize is known, compute $\tilde{x}(0)$ from (35).

(h) With $\tilde{x}(0)$ known, the iteration is completed; the varied initial point $\tilde{x}(0)$ becomes the nominal initial point x(0) for the next iteration; that is, return to (a), and iterate the algorithm.

(i) The algorithm is terminated when the stopping condition (23) is satisfied.

4. Numerical Examples

In this section, five numerical examples are presented using the algorithm of Section 3.4. For simplicity, the symbols employed here denote scalar quantites. The one-dimensional search to determine the stepsize α was performed on the functional P; a bisection process from $\alpha = 1$ was employed until Ineq. (34) was satisfied. The algorithm was employed iteratively and was terminated when the inequality

$$P < 10^{-16} \tag{36}$$

was satisfied.

All the computations were performed on the Rice University Burroughs B-5500 computer in double-precision arithmetic; the algorithm was programmed in FORTRAN IV; the interval of integration was divided into 50 steps for the first four examples and 200 steps for the fifth example. The differential system (17) was integrated using Hamming's modified predictor-corrector method with a special Runge-Kutta procedure to start the integration routine (Ref. 11). The definite integral P was computed using Simpson's rule.

Example 4.1. Consider the differential equations

$$\dot{x} = y \quad , \quad \dot{y} = -\exp(x) \tag{37}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad x(1) = 0 \tag{38}$$

Assume the nominal functions

$$x(t) = 0 \quad , \quad y(t) = -1 \tag{39}$$

which are consistent with the boundary conditions (38) but not consistent with the differential equations (37). Starting with these nominal functions, we employ the algorithm of Section 3.4. Convergence to the solution is achieved in 3 iterations. The numerical results are presented in Tables 1 and 2, where N denotes the iteration number.[6]

[6] In Tables 1-2 as well as subsequent tables, all data are truncated rather than rounded-off.

Table 1. Performance index (Example 4.1).

N	α	P
0	—	0.2×10^{1}
1	1	0.4×10^{-4}
2	1	0.1×10^{-12}
3	1	0.1×10^{-29}

Table 2. Converged solution (Example 4.1, N = 3).

t	x	y
0.0	0.0000×10^{0}	0.5493×10^{0}
0.1	0.4984×10^{-1}	0.4467×10^{0}
0.2	0.8918×10^{-1}	0.3394×10^{0}
0.3	0.1176×10^{0}	0.2284×10^{0}
0.4	0.1347×10^{0}	0.1148×10^{0}
0.5	0.1405×10^{0}	-0.1508×10^{-9}
0.6	0.1347×10^{0}	-0.1148×10^{0}
0.7	0.1176×10^{0}	-0.2284×10^{0}
0.8	0.8918×10^{-1}	-0.3394×10^{0}
0.9	0.4984×10^{-1}	-0.4467×10^{0}
1.0	-0.1324×10^{-22}	-0.5493×10^{0}

Example 4.2. Consider the differential equations

$$\dot{x} = yz \quad , \quad \dot{y} = -xz \quad , \quad \dot{z} = x - \sin(\frac{\pi}{2}t) \tag{40}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad y(0) = 1 \quad , \quad x(1) = 1 \tag{41}$$

Assume the nominal functions

$$x(t) = t \quad , \quad y(t) = 1 \quad , \quad z(t) = 1 \tag{42}$$

which are consistent with the boundary conditions (41), but not consistent with the differential equations (40). Starting with these nominal functions, we employ the algorithm of Section 3.4. Convergence to the solution is achieved in 13 iterations. The numerical results are presented in Tables 3 and 4, where N denotes the iteration number.

Table 3. Performance index (Example 4.2).

N	α	P
0	—	0.3×10^{0}
1	1	0.4×10^{-2}
2	1	0.2×10^{-3}
3	1	0.1×10^{-4}
4	1	0.1×10^{-5}
5	1	0.6×10^{-7}
6	1	0.4×10^{-8}
7	1	0.2×10^{-9}
8	1	0.1×10^{-10}
9	1	0.9×10^{-12}
10	1	0.6×10^{-13}
11	1	0.3×10^{-14}
12	1	0.2×10^{-15}
13	1	0.1×10^{-16}

Table 4. Converged solution (Example 4.2, N = 13).

t	x	y	z
0.0	0.0000×10^{0}	0.1000×10^{1}	0.1570×10^{1}
0.1	0.1564×10^{0}	0.9876×10^{0}	0.1570×10^{1}
0.2	0.3090×10^{0}	0.9510×10^{0}	0.1570×10^{1}
0.3	0.4539×10^{0}	0.8910×10^{0}	0.1570×10^{1}
0.4	0.5877×10^{0}	0.8090×10^{0}	0.1570×10^{1}
0.5	0.7070×10^{0}	0.7071×10^{0}	0.1570×10^{1}
0.6	0.8089×10^{0}	0.5878×10^{0}	0.1570×10^{1}
0.7	0.8909×10^{0}	0.4540×10^{0}	0.1570×10^{1}
0.8	0.9510×10^{0}	0.3090×10^{0}	0.1570×10^{1}
0.9	0.9876×10^{0}	0.1565×10^{0}	0.1570×10^{1}
1.0	0.1000×10^{1}	0.7479×10^{-4}	0.1570×10^{1}

<u>Example 4.3.</u> Consider the differential equations

$$\dot{x} = 10y \quad , \quad \dot{y} = 10z \quad , \quad \dot{z} = -5xz \tag{43}$$

subject to the boundary conditions

$$x(0) = 0 \quad , \quad y(0) = 0 \quad , \quad y(1) = 1 \tag{44}$$

Assume the nominal functions

$$x(t) = 0 \quad , \quad y(t) = t \quad , \quad z(t) = 0 \tag{45}$$

which are consistent with the boundary conditions (44) but not consistent with the differential equations (43). Starting with these nominal functions, we employ the algorithm of Section 3.4. Convergence to the solution is achieved in 6 iterations. The numerical results are presented in Tables 5 and 6, where N denotes the iteration number.

Table 5. Performance index (Example 4.3).

N	α	P
0	—	0.3×10^{2}
1	1	0.1×10^{1}
2	1/8	0.2×10^{0}
3	1	0.3×10^{-1}
4	1	0.5×10^{-4}
5	1	0.2×10^{-9}
6	1	0.2×10^{-20}

Table 6. Converged solution (Example 4.3, N = 6).

t	x	y	z
0.0	0.0000×10^{0}	0.0000×10^{0}	0.3320×10^{0}
0.1	0.1655×10^{0}	0.3297×10^{0}	0.3230×10^{0}
0.2	0.6500×10^{0}	0.6297×10^{0}	0.2667×10^{0}
0.3	0.1396×10^{1}	0.8460×10^{0}	0.1613×10^{0}
0.4	0.2305×10^{1}	0.9555×10^{0}	0.6423×10^{-1}
0.5	0.3283×10^{1}	0.9915×10^{0}	0.1590×10^{-1}
0.6	0.4279×10^{1}	0.9989×10^{0}	0.2401×10^{-2}
0.7	0.5279×10^{1}	0.9999×10^{0}	0.2201×10^{-3}
0.8	0.6279×10^{1}	0.9999×10^{0}	0.1230×10^{-4}
0.9	0.7279×10^{1}	0.9999×10^{0}	0.4339×10^{-6}
1.0	0.8279×10^{1}	0.1000×10^{1}	0.4208×10^{-8}

Example 4.4. Consider the differential equations

$$\dot{x} = y \quad , \quad \dot{y} = 6y + 12x^2 \quad , \quad \dot{z} = -12x^2 \tag{46}$$

subject to the boundary conditions

$$y(0) = -1 \quad , \quad z(0) = 0 \quad , \quad y(1) = 0 \tag{47}$$

Assume the nominal functions

$$x(t) = 0.5 \quad , \quad y(t) = t-1 \quad , \quad z(t) = t \tag{48}$$

which are consistent with the boundary conditions (47) but not consistent with the differential equations (46). Starting with these nominal functions, we employ the algorithm of Section 3.4. Convergence to the solution is achieved in 4 iterations. The numerical results are presented in Tables 7 and 8, where N denotes the iteration number.

Table 7. Performance index (Example 4.4).

N	α	P
0	—	0.2×10^{2}
1	1	0.5×10^{0}
2	1	0.6×10^{-3}
3	1	0.1×10^{-8}
4	1	0.4×10^{-20}

Table 8. Converged solution (Example 4.4, N = 4).

t	x	y	z
0.0	0.8256×10^{0}	-0.1000×10^{1}	0.0000×10^{0}
0.1	0.7355×10^{0}	-0.8115×10^{0}	-0.7291×10^{0}
0.2	0.6618×10^{0}	-0.6696×10^{0}	-0.1313×10^{1}
0.3	0.6005×10^{0}	-0.5600×10^{0}	-0.1790×10^{1}
0.4	0.5490×10^{0}	-0.4734×10^{0}	-0.2186×10^{1}
0.5	0.5053×10^{0}	-0.4026×10^{0}	-0.2519×10^{1}
0.6	0.4681×10^{0}	-0.3418×10^{0}	-0.2803×10^{1}
0.7	0.4368×10^{0}	-0.2843×10^{0}	-0.3048×10^{1}
0.8	0.4114×10^{0}	-0.2211×10^{0}	-0.3263×10^{1}
0.9	0.3933×10^{0}	-0.1364×10^{0}	-0.3457×10^{1}
1.0	0.3858×10^{0}	-0.1033×10^{-24}	-0.3638×10^{1}

Example 4.5. Consider the differential equations[7]

$$\dot{x} = 13y \;,\quad \dot{y} = 13z \;,\quad \dot{z} = -20.15xz + 1.3y^2 - 13u^2 + 2.6y + 13$$
$$\dot{u} = 13w \;,\quad \dot{w} = -20.15\,xw + 14.3\,yu + 2.6u - 2.6 \tag{49}$$

subject to the boundary conditions

$$x(0) = 0 \;,\quad y(0) = 0 \;,\quad u(0) = 0 \;,\quad y(1) = 0 \;,\quad u(1) = 1 \tag{50}$$

Assume the nominal functions

$$x(t) = 0 \;,\quad y(t) = 0 \;,\quad z(t) = 0 \;,\quad u(t) = t \;,\quad w(t) = 0 \tag{51}$$

which are consistent with the boundary conditions (50) but not consistent with the differential equations (49). Starting with these nominal functions, we employ the algorithm of Section 3.4. Convergence to the solution is achieved in 6 iterations. The numerical results are presented in Tables 9 and 10, where N denotes the iteration number.

[7] This example, which involves unstable differential equations, was considered in Ref. 12.

Table 9. Performance index (Example 4.5).

N	α	P
0	—	0.9×10^{2}
1	1/2	0.3×10^{2}
2	1/2	0.8×10^{1}
3	1	0.4×10^{-1}
4	1	0.6×10^{-4}
5	1	0.5×10^{-10}
6	1	0.3×10^{-22}

Table 10. Converged solution (Example 4.5, N = 6).

t	x	y	z	u	w
0.0	0.0000×10^{0}	0.0000×10^{0}	-0.9663×10^{0}	0.0000×10^{0}	0.6529×10^{0}
0.1	-0.5028×10^{0}	-0.5802×10^{0}	-0.7188×10^{-1}	0.6971×10^{0}	0.4220×10^{0}
0.2	-0.1215×10^{1}	-0.4603×10^{0}	0.1945×10^{0}	0.1100×10^{1}	0.2036×10^{0}
0.3	-0.1631×10^{1}	-0.1744×10^{0}	0.2210×10^{0}	0.1247×10^{1}	0.3249×10^{-1}
0.4	-0.1688×10^{1}	0.7033×10^{-1}	0.1443×10^{0}	0.1213×10^{1}	-0.7189×10^{-1}
0.5	-0.1506×10^{1}	0.1844×10^{0}	0.3000×10^{-1}	0.1093×10^{1}	-0.1002×10^{0}
0.6	-0.1270×10^{1}	0.1602×10^{0}	-0.5755×10^{-1}	0.9815×10^{0}	-0.6490×10^{-1}
0.7	-0.1120×10^{1}	0.6614×10^{-1}	-0.7534×10^{-1}	0.9334×10^{0}	-0.1024×10^{-1}
0.8	-0.1091×10^{1}	-0.1365×10^{-1}	-0.4303×10^{-1}	0.9447×10^{0}	0.2223×10^{-1}
0.9	-0.1133×10^{1}	-0.4258×10^{-1}	-0.1453×10^{-2}	0.9774×10^{0}	0.2352×10^{-1}
1.0	-0.1173×10^{1}	-0.1508×10^{-20}	0.9405×10^{-1}	0.1000×10^{1}	0.1765×10^{-1}

5. Further Numerical Examples

For comparison purposes, the examples of Section 4 were recalculated using the algorithm of Section 3.5. The nominal coordinates at $t = 0$ were identical with those employed in the previous section. In this algorithm, the integration of the differential equation (9) was performed using Hamming's modified predictor-corrector method with a special Runge-Kutta procedure to start the integration routine (Ref. 11). Since $\dot{x} - \varphi = 0$ along each nominal curve, the performance index P contains only the error due to violation of the final conditions.

Computations were also performed by setting the stepsize at the fixed value $\alpha = 1$ in both the algorithm of Section 3.4 and the algorithm of Section 3.5. For the sake of brevity, the detailed results are omitted. Only the number of iterations at convergence is presented in Table 11.

The results show that a definite advantage exists in controlling the stepsize α through the performance index P. For the algorithm of Section 3.4, Example 4.3, the number of iterations for convergence was 6 with stepsize control and 8 without stepsize control. For the algorithm of Section 3.5, Example 4.4, the number of iterations for convergence was 4 with stepsize control and 20 without stepsize control.

Of particular interest is Example 4.5, which involves rather unstable differential equations. For this example, a solution was obtained by means of the algorithm of Section 3.4 with stepsize control included. The algorithm of

Section 3.4 without stepsize control and the algorithm of Section 3.5 with or without stepsize control failed to produce a solution.

For the above reasons as well as the considerations of the previous paragraph, it seems that the algorithm of Section 3.4 with stepsize control included is to be preferred. It is emphasized that the above conclusions were obtained through particular examples and that, consequently, the subject requires further investigation.

Table 11. Number of iterations for convergence.

	Algorithm of Section 3.4		Algorithm of Section 3.5	
	$\alpha \leq 1$	$\alpha = 1$	$\alpha \leq 1$	$\alpha = 1$
Example 4.1	3	3	3	3
Example 4.2	13	13	13	13
Example 4.3	6	8	5	5
Example 4.4	4	4	6	20
Example 4.5	6	–	–	–

6. Discussion and Conclusions

In this thesis, a general technique for solving nonlinear, two-point, boundary-value problems is presented; it is assumed that the differential system has order n and is subject to p initial conditions and q final conditions, where $p + q = n$. First, the differential equations and the boundary conditions are linearized about a nominal function x(t) satisfying the p initial conditions. Next, the linearized system is imbedded into a more general system by means of a scaling factor α, $0 \le \alpha \le 1$, applied to each forcing term. Then, the method of particular solutions (Ref. 2) is employed in order to obtain the perturbation $\Delta x(t) = \alpha A(t)$ leading from the nominal function x(t) to the varied function $\tilde{x}(t)$; this method differs from the adjoint method and the complementary function method in that it employs only one differential system, namely, the nonhomogeneous, linearized system.

As a guide during progression of the algorithm, the performance index P, already introduced in Ref. 10, is employed. This is the cumulative error in the differential equations and the boundary conditions and can be proved to have a descent property with respect to the scaling factor (or stepsize) α. The stepsize α is determined by a one-dimensional search so as to ensure satisfaction of the inequality $\tilde{P} < P$; this can be achieved through a bisection process starting from $\alpha = 1$. In addition, the performance index is also employed as a convergence criterion: the algorithm is terminated when the performance index becomes smaller than some preselected value.

Computationally, the present technique can be employed in two ways: (a) the function $x(t)$ is updated according to $\tilde{x}(t) = x(t) + \alpha A(t)$; or (b) the initial point $x(0)$ is updated according to $\tilde{x}(0) = x(0) + \alpha A(0)$, and the new nominal function $\tilde{x}(t)$ is obtained by forward integration of the nonlinear differential system. In this connection, five numerical examples are presented; they illustrate (i) the simplicity as well as the rapidity of convergence of the algorithm, (ii) the importance of stepsize control, and (iii) the desirability of updating the function $x(t)$ according to Scheme (a).

References

1. GOODMAN, T.R., and LANCE, C.N., The Numerical Integration of Two-Point Boundary-Value Problems, Mathematical Tables and Other Aids to Computation, Vol. 10, No. 54, 1956.

2. MIELE, A., Method of Particular Solutions for Linear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968.

3. BELLMAN, R.E., and KALABA, R.E., Quasilinearization and Nonlinear Boundary-Value Problems, The RAND Corporation, Report No. R-438-PR, 1965.

4. LEE, E.S., Quasilinearization and Invariant Imbedding, Academic Press, New York, 1968.

5. KELLER, H.B., Numerical Methods for Two-Point Boundary-Value Problems, Blaisdell Publishing Company, Waltham, Massachusetts, 1968.

6. BAILEY, P.B., SHAMPINE, L.F., and WALTMAN, P.E., Nonlinear, Two-Point Boundary-Value Problems, Academic Press, New York, 1968.

7. HEIDEMAN, J.C., Use of the Method of Particular Solutions in Nonlinear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 6, 1968.

8. HEIDEMAN, J.C., Use of the Method of Particular Solutions in Quasilinearization, AIAA Journal, Vol. 6, No. 12, 1968.

9. HEIDEMAN, J.C., Application of the Method of Particular Solutions to Boundary-Layer Analyses, Journal of the Astronautical Sciences, Vol. 15, No. 6, 1968.

10. MIELE, A., HEIDEMAN, J.C., and DAMOULAKIS, J.N., The Restoration of Constraints in Holonomic and Nonholonomic Problems, Journal of Optimization Theory and Applications, Vol. 3, No. 5, 1969.

11. RALSTON, A., Numerical Integration Methods for the Solution of Ordinary Differential Equations, Mathematical Methods for Digital Computers, Vol. 1, Edited by A. Ralston and H.S. Wilf, John Wiley and Sons, New York, 1960.

12. ROBERTS, S.M., SHIPMAN, J.S., and ROTH, C.V., Continuation in Quasilinearization, Journal of Optimization Theory and Applications, Vol. 2, No. 3, 1968.