JOURNAL OF COMPUTATIONAL PHYSICS 39, 103–111 (1981)

Solution of Navier–Stokes Equations by Goal Programming

K. Y. K. NG

Operational Research and Analysis Establishment, Department of National Defense, Ottawa, Ontario, Canada

Received June 28, 1979; revised November 2, 1979

Goal programming, a mathematical tool for the analysis of problems involving multiple, conflicting objectives arising in the fields of operations research and systems analysis, is employed to find the numerical solution of certain Navier–Stokes equations. As in the collocation method, the proposed technique involves approximating the unknown solution by a set of trial functions containing unknown coefficients. The technique then minimizes in a weighted residual sense the absolute value deviations of the differential equation residual by the modified pattern search algorithm for nonlinear goal programs. One important feature of this method in solving nonlinear problems is that it does not require the initial programming effort needed to set up a Newton method (or a similar approach) based upon the collocation approximation for the differential equation. Further, this approach is fundamentally more general than the collocation method because the number of undetermined parameters can be less than the number of spatial points. It is shown that the approximate solution to a Navier–Stokes equation with only a low order trial function compares favourably to other methods of weighted residual results.

1. INTRODUCTION

The method of weighted residuals (MWR) has long been considered one of the main techniques for developing approximate solutions to operator equations. In all of these problems, the unknown solution is approximated by a set of trivial functions containing unknown coefficients. These coefficients are chosen by various error criteria to give the "best" approximation for the selected family. Obviously, there is a wide choice in selecting the error criteria. The most commonly used methods in this group are the collocation method, the least squares method, the Galerkin method, the subdomain method and the orthogonal collocation method [1].

It should be noted that except for the collocation methods, the application of other MWR techniques to nonlinear problems often requires the evaluation of integrals of the trial function [1]. In the collocation method, it is only necessary to evaluate the residual at the collocation points. The problem is thus reduced to the solution of a set of nonlinear algebraic equations. Unfortunately, it is well known that the solution of systems of nonlinear equations can be extremely complicated. Thus there is clearly a need for alternative methods of solving nonlinear problems which involve fewer program changes and also reduce the preparation and computation effort.

0021-9991/81/010103–09$02.00/0

The goal programming approach presented here offers a promising alternative to the approximate solutions of nonlinear problems. Goal programming [2], first introduced by Charnes and Cooper [3], is a modification and extension of linear programming. It is a tool for the analysis of problems invoving multiple, conflicting objectives arising in the fields of operations research and systems analysis. The goal programming method for solving nonlinear boundary value problems resembles the collocation technique in that it does not require the evaluation of integrals of the trial function. The criterion minimizes in a weighted residual sense the absolute value deviations of the differential equation residual by the modified pattern search algorithm [2]. It should be mentioned that our method of minimizing the residuals is novel. A similar approach employing Powell's nonlinear least squares minimization method has been reported by Eason and Mote [4]. Several other authors [5–7] have also applied minimization techniques to a least squares differential equation formulation. Our approach differs from these techniques in that the sum of the absolute value deviations of the differential equation residual is minimized (in a weighted residual sense); in [4], the sum of the squared residuals at individual points is minimized instead.

The advantage of our proposed method is that it offers a solution to complicated problems without the initial programming effort required to set up a Newton method (or a similar approach) based upon the collocation approximation for the differential equation. Furthermore, this minimization approach is fundamentally more general than the collocation method because the number of undetermined coefficients can be less than the number of collocation points. Last, the goal programming technique for nonlinear problems can be written as general purpose software, in which problem changes affect only short subprograms and input data.

In Section 2 we present the methodolgy for the solution of nonlinear boundary values problems by means of the goal programming approach. In that section we also discuss problems concerning parameter selection, namely, the distribution of collocation points and the determination of the number of trial functions that will yield a satisfactory solution. In Section 3, we apply the method to finding the numerical solutions of certain Navier–Stokes equations which arise in the description of fluids near stagnation points. It will be demonstrated that even with low order trial functions, the solutions obtained compare favourably to other MWR results.

2. Methodology

Consider the boundary value problem

$$F\left(x, u, \frac{du}{dx}, \frac{d^2u}{dx^2}, \dots, \frac{d^nu}{dx^n}\right) = 0, \tag{2.1}$$

$$Bu = 0 \text{ on boundary,} \tag{2.2}$$

where F is a nonlinear differential operator, and B is a boundary operator. A trial function is taken in the form [1]

$$v = \sum_{i=1}^{N} c_i u_i, \tag{2.3}$$

$$Bu_i = 0 \text{ on boundary,} \tag{2.4}$$

where c_i, $i = 1, 2,\ldots, N$, are the unknown parameters. The trial function (2.3) is substituted into the differential equation (2.1) to form the residuals

$$R(c_i, x) = F\left(x, \sum_{i=1}^{N} c_i u_i, \sum_{i=1}^{N} c_i \frac{du_i}{dx}, \ldots, \sum_{i=1}^{N} c_i \frac{d^n u_i}{dx^n}\right).$$

If the trial functions were the exact solution, the residual $R(c_i, x)$ would be zero. As pointed out earlier, there exist a number of techniques for determining the values of c_i. In this paper we demonstrate the application of goal programming towards the determination of the values c_i.

First we define the weighted integrals of the residual as

$$\int w_j R(c_i, x)\, dx, \tag{2.5}$$

where w_j is the displayed Dirac delta function $w_j = \delta(x - x_j)$. The weighted integral of residual (2.5) then becomes $R(c_i, x_j)$. Instead of setting the weighted residual to zero at N specified collocation points as in the orthogonal collocation method, we choose M collocation points, x_j, $j = 1, 2,\ldots, M$, with $M > N$. We then anticipate the trial function v to approximate the solution of (2.1) as closely as possible at the M chosen collocation points x_j. This is achieved by minimizing the sum of the absolute value derivations $R(c_i, x_j)$ over the chosen M collocation points. In other words, we are to find

$$\min_{\substack{c_i \\ i=1,2,\ldots,N}} \sum_{j=1}^{M} |R(c_i, x_j)|. \tag{2.6}$$

A closer look at the formulation of Eq. (2.6) reveals that we are actually finding the optimum values of c_i with respect to the L_1 norm. Equation (2.6) can be rewritten in the goal programming formulation

$$\min \sum_{j=1}^{M} (n_j + p_j) \tag{2.7}$$

subject to

$$\begin{aligned} R(c_i, x_1) + n_1 - p_1 &= 0, \\ R(c_i, x_2) + n_2 - p_2 &= 0, \\ \cdots\cdots\cdots\cdots &, \\ \cdots\cdots\cdots\cdots &, \\ R(c_i, x_M) + n_M - p_M &= 0, \end{aligned} \tag{2.8}$$

where n_j = deviational variables representing negative deviations from goal level j ($n_j \geqslant 0$); p_j = deviational variables representing positive deviations from goal level j ($p_j \geqslant 0$). That is, in the above formulation, the c_i's are chosen such that the sum of the absolute value deviations from each objective having a goal of value "zero" is minimized. (When solving systems (2.7), (2.8), the minimum is assumed to be attained if the achievement function, Eq. (2.7), is less than a certain tolerance factor.)

In essence, our method requires the choice of the form of v in Eq. (2.3) and a set of collocation points. Then M residuals $|R(c_i, x_j)|$ are determined by substituting the N-term approximation v into (2.1), (2.2). A residual is included for each equation (2.1), (2.2) at every point appropriate to each. A subroutine that calculates $\sum |R(c_i, x_j)|$ for a given initial approximation is all that is necessary to solve (2.7), (2.8) by the modified pattern search technique for the nonlinear goal programs. The pattern search technique iteratively searches for the minimum sum of $\sum |R(c_i, x_j)|$. It increases its search step size if previous searches have been successful and decreases the step size otherwise. The procedure terminates when the convergence criterion is satisfied. The technique so described is robust and efficient because it does not require the evaluation of any derivative. Further, it is a particularly easy method to program. Detailed descriptions of the technique can be found in [2].

In Section 3, we address some problems that arise in the goal programming solution, namely, the choice of collocation points and the determination of the number of trial functions that will yield a satisfactory solution. Experience with the collocation method indicates that the collocation points x_j in (2.5) may be equally spaced, or concentrated in areas where the solution increases rapidly. We will adopt this experience obtained from the collocation method to the goal programming technique. A worked example in Section 3 illustrates this fact.

Because of the absence of rigorous bounds on the error of the approximate solutions for reasonably small, finite N, users of the MWR often assume that small root mean square (rms) residuals are good indicators of accurate solutions. This indicator will be employed as our criterion for the accuracy of the goal programming solution. (It should be mentioned that this is only an indication of the error and not a rigorous bound.) In some problems, it is possible to calculate rigorous bounds on the difference at any point between the actual solution and the approximate solution. Detailed discussions of error bounds can be found in [1, 8, 15]. Last, it is worth mentioning that nonlinear problems often have multiple solutions, which can usually be distinguished from convergence failure by inspecting $\sum |R(c_i, x_j)|$.

Now, we are ready to outline the steps for the determination of the number of trial functions required to provide a satisfactory solution:

Step 1. Assume an educated guess on the number of trial functions N in v, Eq. (2.3). Go to step 2.

Step 2. Solve for c_i by means of the modified pattern search algortihm. Go to step 3.

Step 3. Evaluate the rms residual. If it is less than or equal to a prespecified tolerance factor, then stop and the solution is given by v with the c_i evaluated in step 2. Or else, set $N = N + 1$, and go to step 2.

The above procedure for the determination of the approximate solutions to certain Navier–Stokes equations is demonstrated in Section 3.

3. Examples

In this section we apply the goal programming methodology to find the solution of nonlinear boundary value problems arising in studies of fluid motion.

(i) We consider the numerical solution of the Navier–Stokes equation for the 3-dimensional, axisymmetric case of flow with stagnation. We are interested in obtaining the solution of a problem where a fluid stream impinges on a wall at right angles to it and flows away radially in all directions. Such a case occurs in the neighbourhood of a stagnation point of a body of revolution in a flow parallel to its axis. The governing system of equations is given by [9] as

$$f''' + 2ff'' + 1 - f'^2 = 0 \tag{3.1}$$

with boundary conditions

$$f(0) = f'(0) = 0, \ \lim_{x \to \infty} f'(x) = 1, \tag{3.2}$$

where f is the dimensionless dependent (similarity) variable related to the velocity distribution of the flow at the stagnation point and x is a similarity variable. Let the u_i in (2.3) be given by

$$u_1(x) = -1 + e^{-x} + x,$$
$$u_i(x) = (i-1) - ie^{-x} + e^{-ix}, \qquad i = 2, 3, ...,$$

and

$$v(x) = c_1 u_1(x) + c_2 u_2(x) + c_3 u_3(x) + \cdots . \tag{3.3}$$

It is obvious that v satisfies the boundary conditions (3.2). In this example, we desire that the rms residual (taken over 16 evenly distributed points from 0 to 1.5) be less than 1×10^{-1}. Following the procedure outlined in Section 2 for the determination of the number of trial functions required to yield a satisfactory solution, we assume that the trial function (3.3) consists of only 3 terms. Applying the goal programming technique on (3.3) with 12 suitably chosen collocation points, namely,

$$\begin{aligned} &x_1 = 0.05, x_2 = 0.1, x_3 = 0.2, x_4 = 0.3, x_5 = 0.4, x_6 = 0.5, \\ &x_7 = 0.6, x_8 = 0.7, x_9 = 0.8, x_{10} = 0.9, x_{11} = 1.0, x_{12} = 1.5, \end{aligned} \tag{3.4}$$

initial guess $c_1 = 1.0$, $c_2 = -0.1$, $c_3 = -0.1$, and the tolerance factor for the achievement function (2.7) as 5×10^{-3}, the rms residual obtained is 1.33×10^{-1}. By the procedure described in Section 2, we are to increase the number of independent functions by one. Again with the set of collocation points as given by (3.4), initial guess $c_1 = 0.7$, $c_2 = -0.2$, $c_3 = c_4 = 0.1$, and the tolerance factor for (2.7) as 5×10^{-3}, the approximate solution is given by

$$v(x) = 1.106247u_1(x) - 0.199996u_2(u) - 0.224996u_3(x) + 0.099997u_4(x) \quad (3.5)$$

and the rms residual is equal to 9.712×10^{-2}. The computational time for generating (3.5) on an IBM 370/168 was 5.61 sec. It is interesting to compare the results of the numerical calculations obtained by goal programming and the result obtained by Frössling (see [9]), in terms of the wall shear stress $f''(0)$. The goal programming solution for this case is $f''(0) = 1.312506$ and the solution obtained by Frössling is $f''(0) = 1.312$.

Next, we discuss the effect of the distribution of collocation points on the accuracy of the approximate solution. Again we assume that the trial function $v(x)$ is to consist of 4 terms, with initial guess for c_i as 0.7, –0.2, 0.1, 0.1, tolerance factor for (2.7) as 5×10^{-3}, and the collocation points given by

0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.6, 0.8, 1.0, 1.5.

The goal programming solution obtained has a rms residual equal to 1.784×10^{-1} and $f''(0) = 1.274932$. The explanation of the high rms residual might be the unevenly distributed collocation points.

Finally, we are interested in determining the number of collocation points required to produce a satisfactory solution to (3.1), (3.2). A computer experiment was performed with the trial function consisting of 4 terms and with initial guess and tolerance factor for the achievement function the same as those given above. However, we are to evaluate (2.7) at 7 collocation points only, namely,

0.05, 0.2, 0.4, 0.6, 0.8, 1.0, 1.5.

The goal programming solution obtained has a rms residual 9.728×10^{-2} and $f''(0) = 1.265628$. It is easily noted that if the number of collocation points is decreased, the rms residual increases slightly and the wall shear stress is less accurate.

In concluding our results for this example, the rms residual will decrease as the number of collocation points increases. (Nevertheless, there is a trade-off between the desired accuracy of the approximate solution and the computational time.) Also, in the absence of a theory on the distribution of spatial points that yields a satisfactory solution, evenly distributed collocation points should be preferred. Of greater importance, it is demonstrated that with only 4 terms in the trial function, the wall shear stress obtained by the goal programming technique is accurate to within 3 decimal places compared with the classical solution given by Frössling (see [9]).

(ii) In this example we consider the goal programming solution of a viscoelastic boundary layer equation, which arises in the description of a second order fluid near a two-dimensional stagnation point [10],

$$f''' + ff'' + 1 - f'^2 + k(ff'''' - 2f'f''' + f''^2) = 0, \tag{3.6}$$

$$f(0) = f'(0) = 0. \qquad \lim_{x\to\infty} f'(x) = 1. \tag{3.7}$$

Here f is a dimensionless stream function, k is a non-negative elastic parameter and x is a similarity variable. On physical grounds we also assume that $\lim_{x\to\infty} f''''(x)$ exists and is finite. Thus the evaluation of (3.7) at $x = 0$ gives

$$f'''(0) = -(1 + kf''^2(0)).$$

Throughout this example we will assume k to be equal to 0.2, since we are only interested in the behaviour of a "weakly" viscoelastic fluid. Approximate solutions to the system (3.6)–(3.7) have been obtained by the Karman–Pohlhausen method [11], the perturbation method [12], and the orthogonal collocation method [13]. It is demonstrated in [13] that standard finite difference techniques such as the Runge–Kutta and the predictor–corrector methods are highly unstable when applied to Eqs. (3.6)–(3.7). Hence accurate finite difference solutions have not been obtained.

In order to compare the goal programming solution of (3.6), (3.7) with other MWR results, we follow Serth [13] and choose the trial function to be the set of Laguerre functions which constitutes a complete orthonormal system in $(0, \infty)$ [14], that is,

$$v(x) = -1 + x + \exp(-x) + x^2 \exp\left(-\frac{x}{2}\right) \sum_{i=1}^{N} c_i L_i(x), \tag{3.8}$$

where $L_i(x) = (1/i!) \exp(x)(d/dx)^i \exp(-x)x^i$ is the Laguerre polynomial of degree i [14]. The coefficients c_i are determined by nonlinear goal programming at the set of collocation points given by (3.4). In addition, we let the tolerance factor for the achievement function (2.7) be 2×10^{-4}, the number of terms in the trial function (3.8) be 5, and the initial guess for c_i be $c_1 = c_3 = c_5 = 1.0$, $c_2 = c_4 = -1.0$. The computing time for generating the approximate solution v on an IBM 370/168 was 6.79 sec, and the coefficients c_i are found to be

$$c_1 = 0.378310, \qquad c_2 = -0.470311, \qquad c_3 = 1.089032,$$
$$c_4 = -1.074959, \qquad c_5 = 0.371872.$$

Our results show that the rms residual is 9.646×10^{-2} (taken over 16 uniformly distributed points from 0 to 1.5) and $f''(0) = 1.58788$, using only 5 independent Laguerre functions. By means of the orthogonal collocation method, however, Serth [14] found that using 5 Laguerre functions, $f''(0) = 1.56640$; using 12 Laguerre functions, $f''(0) = 1.58678$; using 16 Laguerre functions, $f''(0) = 1.58800$; and finally, using 24 Laguerre functions, $f''(0) = 1.58719$. It is amazing to find that with the trial function containing only 5 independent functions, the goal programming technique provides a solution that is comparable in accuracy to the orthogonal collocation method employing a trial function involving 12 independent functions.

4. Conclusion

In this paper we have illustrated that goal programming can be applied to find the numerical solution of certain Navier–Stokes equations. The numerical results obtained are extremely encouraging. It can be seen from the worked example in Section 3 that the goal programming solution is notably more accurate and computationally simpler than the orthogonal collocation solution. However, the relatively low computational times for the worked examples are dependent on the initial choice of c_i. It is well known that the better the choice of c_i, the faster the search will converge. Often, an educated guess on the parameters can be made by examining equations that are similar in type. For example, from experience in dealing with the Blasius equation and the Falkner–Skan equation, we would expect the wall shear stress $f''(0)$ in our examples to lie in the range of 1 and 2. A careful algebraic manipulation on the initial guess of c_i so that $f''(0)$ falls in the range of 1 and 2 would definitely save much computational effort.

When applying the method of goal programming, the main difficulty is the choice of trial functions. (In fact this also applies to the method of weighted residuals.) For simple boundary value problems, the choice of trial functions may be quite obvious [1]. However, for certain nonlinear equations, the choice of appropriate approximate functions often requires immense physical intuition as well as linearization techniques. In the event that prior information on the solution profile of the nonlinear equation is lacking, an educated guess on the trial modes can be made by employing the orthogonal polynomials. To justify this statement, a carefully programmed, tuned and tested code on the goal programming methodology with the orthogonal polynomials as the trial functions should be applied to the solution of various nonlinear equations. This is under investigation at present and the results will be discussed in a subsequent paper.

As a last remark, the goal programming formulation is independent of the form of the differential system, so that changing from one problem to another is easy. Both examples were run with the same computer program, requiring only a slight change in a subprogram and the input data. Thus the goal programming technique for solving nonlinear problems is robust and efficient. It definitely warrants more attention and further research.

Acknowledgments

The author wishes to thank the referees for their helpful comments concerning an earlier version of this article.

REFERENCES

1. B. A. FINLAYSON, "The method of Weighted Residuals and Variational Principles," Academic Press, New York/London, 1972.
2. J. P. IGNIZIO, "Goal Programming and Extensions," Heath (Lexington Books), Lexington, Mass., 1976.
3. A. CHARNES AND W. W. COOPER, "Management models and Industrial Applications of Linear Programming," Vols. I, II, Wiley, Sons, New York, 1961.
4. E. D. EASON AND C. D. MOTE, JR., *Int. J. Num. Meth. Eng.* **11** (1977), 641–652.
5. H. J. ECKERT, JR., "Piecewise Polynomials and the Methods of Weighted Residuals for Nonlinear Ordinary Differential Equations," Ph. D. thesis, University of Illinois, 1969; University Microfilms No. 70–839.
6. T. I. ALLEN, *Chem. Phys. Lett.* **13** (1972), 504–506.
7. J. R. PARTIN, "Numerical Techniques for Applying Averaging Methods to Nonlinear Ordinary Differential Equations," Ph. D. thesis, University of Texas, 1965; University Microfilms No. 65–10758.
8. E. D. EASON, *Int. J. Num. Meth. Eng.* **10** (1976), 1021–1046.
9. H. SCHLICHTING, "Boundary-Layer Theory," pp. 91–92, McGraw–Hill, New York, 1968.
10. D. W. BEARD AND K. WALTERS, *Proc. Cambridge Phil. Soc.* **60** (1964), 667.
11. M. H. DAVIES, *Z. Angew. Math. Phys.* **17**(1966), 189.
12. J. PEDDIESON, *AIChE J.* **19** (1973), 377.
13. R. W. SERTH, *J. Eng. Math.* **8**, No. 2 (1974), 89–92.
14. M. ABRAMOWITZ AND I. A. STEGUN, *in* "Handbook of Mathematical Functions," Chaps. 22, 25, National Bureau of Standards, Applied Mathematics Series, No. 55, Nat. Bur. Stand., Washington, D.C., 1966.
15. J. VILLADSEN AND M. L. MICHELSEN, "Solution of Differential Equation Models by Polynomial Approximation," Prentice–Hall, Englewood Cliffs, N.J., 1978.