JOURNAL OF COMPUTATIONAL PHYSICS 12, 403–411 (1973)

On Computer Generated Analytic Solutions to the Equations of Fluid Mechanics. The Case of Creeping Flows

DOMINIQUE BARTHÈS-BIESEL* AND ANDREAS ACRIVOS

Stanford University, Stanford, California 94305

Received August 28, 1972

The purpose of this paper is to illustrate the potential usefulness of symbolic manipulation languages in solving problems of physical interest. As an example, we developed a regular perturbation solution to the equations describing the motion of a liquid droplet, freely suspended in a time dependent shear flow, when both fluids are Newtonian and incompressible. Since the successive equations of the problem become increasingly complicated, the recent language, REDUCE 2, was used to perform automatically most of the routine algebraic computations, thereby extending an existing analysis to higher order in the perturbation parameter.

1. INTRODUCTION

An exciting development in the use of computers which, to date, does no appear to have received the attention it deserves, involves the application of symbolic manipulation languages to obtain analytic (i.e. non-numerical) solutions to many equations arising in fluid mechanics and in other branches of mathematical physics. Recent software developments include MATHLAB [1], SYMBAL [2], FORMAC [3] and more particularly, REDUCE 2,[1] which was the one used in the presented study.

REDUCE is a program completely written in REDUCE language apart from a few LISP primitive functions. It can then easily be modified and extended by those having a knowledge of the system, independently of the type of computer used. Its numerous features render it a very powerful tool for performing general algebraic computations. In particular, its capabilities include arbitrary precision arithmetic, expansion and ordering of rational functions of polynomials, calculation of the greatest common divisor of two polynomials, automatic and user controlled

* Present address, Université de Technologie de Compiegne, Compiegne, France.

[1] A first version was released in 1967, and the new improved one in 1970. Both were developed by Hearn [4], and are available on the IBM 360 series computers, the Digital Equipment Corporation PDP-10 and the CDC 6400 series.

simplification of expressions, calculations with symbolic matrices, and a high energy physics package with spin 1/2 and spin 1 algebra. REDUCE is also provided with a set of general substitution commands, which add greatly to the flexibility and power of the language. For example, those features can be taken advantage of to integrate expressions (provided the values of the indefinite integrals are known), to manipulate trigonometric or special functions whose properties are defined by the user, to taken the Laplace or Fourier transform of expressions, etc. This language can also be used effectively to perform repetitive algebraic computations, for example to evaluate the analytic expression of Legendre polynomials, or to develop a regular perturbation analysis of a nonlinear problem. In both cases, we are limited essentially by computing time or storage considerations.

Perhaps, the most obvious potential applications of REDUCE in fluid mechanics occur when a formal solution to the equations of motion is available, as in the case of inviscid or creeping flows. For example, in the second instance, a general expression for the velocity and stress fields has been derived by Lamb [5], e.g.

$$u_i = \sum_{n=-\infty}^{+\infty} \left[\epsilon_{ijk} \frac{\partial X_n}{\partial x_j} x_k + \frac{\partial \Phi_n}{\partial x_i} + \frac{(n+3)\, r^2}{2(n+1)(2n+3)} \frac{\partial P_n}{\partial x_i} - \frac{n}{(n+1)(2n+3)} x_i P_n\right] \tag{1.1}$$

where ϵ_{ijk} is the permutation symbol. Also, Einstein's summation convention is adopted and X_n, Φ_n, P_n are solid spherical harmonics typically defined as

$$\left.\begin{aligned} \Phi_n &= T^*_{l_1 l_2 \cdots l_n} \frac{\partial^n r^{-1}}{\partial x_{l_1} \partial x_{l_2} \cdots \partial x_{l_n}}\, r^{2n+1}, \\ \Phi_{-n-1} &= T_{l_1 l_2 \cdots l_n} \frac{\partial^n r^{-1}}{\partial x_{l_1} \partial x_{l_2} \cdots \partial x_{l_n}}, \end{aligned}\right\} \quad n > 0$$

with

$$r = (x_l x_l)^{1/2}.$$

The tensors $T_{l_1 l_2 \cdots l_n}$ are chosen to be invariant under any permutation of their indices and to have a zero contradiction with respect to any two indices. Thus, the expression of the harmonics becomes

$$\Phi_n = (-1)^n (2n-1)(2n-3) \cdots 3.1\; T^*_{l_1 l_2 \cdots l_n} x_{l_1} x_{l_2} \cdots x_{l_n},$$

$$\Phi_{-n-1} = (-1)^n (2n-1)(2n-3) \cdots 3.1\; T_{l_1 l_2 \cdots l_n} \frac{x_{l_1} x_{l_2} \cdots x_{l_n}}{r^{2n+1}}.$$

To complete then the analysis of a creeping flow problem, whose solution can be represented by the linear combination of spherical harmonics shown in (1.1), it is necessary to determine the foregoing tensorial coefficients by satisfying the boundary conditions. This process, which often involves tedious algebraic manipulations can be efficiently handled using REDUCE.

We chose as an example the motion, under creeping flow conditions, of a single droplet freely suspended in a time dependent linear shear field, when the fluids involved are Newtonian and incompressible. The mathematical formulation of the problem and the method of analysis will now be briefly described and be followed by an outline of the program that was designed to obtain the solution.

2. The Physical Problem

The Reynolds number of the disturbance due to the drop is assumed to be small so that inertia effects can be neglected. Three boundary conditions are imposed at the surface of the droplet: continuity of the velocities and the tangential stresses, with the difference in normal stresses balanced by surface tension.

The flow problem is thus defined by the following nondimensional equations, referred to a set of axes moving with the center of the drop. Equation of the surface is

$$r = (x_i x_i)^{1/2} = 1 + \epsilon f\left(\frac{x_1}{r}, \frac{x_2}{r}, \frac{x_3}{r}\right). \tag{2.1}$$

Equations of motion are

$$\frac{\partial^2 u_i^*}{\partial x_k^2} = \frac{\partial p^*}{\partial x_i}, \qquad \frac{\partial u_k^*}{\partial x_k} = 0 \qquad \text{for} \quad 0 \leqslant r \leqslant 1 + \epsilon f, \tag{2.2}$$

$$\frac{\partial^2 u_i}{\partial x_k^2} = \frac{\partial p}{\partial x_i}, \qquad \frac{\partial u_k}{\partial x_k} = 0 \qquad \text{for} \quad r > 1 + \epsilon f. \tag{2.3}$$

Boundary conditions are

$$\left.\begin{aligned} u_i - u_i^* &= 0, && (2.4)\\ u_k n_k &= K\epsilon \frac{\partial f}{\partial t}, && (2.5)\\ (p_{ij} - \lambda p_{ij}^*)\, n_j &= n_i k \left(\frac{1}{R_1} + \frac{1}{R_2}\right) \end{aligned}\right\} \quad \text{at } r = 1 + \epsilon f$$

$$u_i \to e_{il} x_l + \tfrac{1}{2}\epsilon_{ikl}\omega_k x_l \qquad \text{as} \quad r \to \infty, \tag{2.7}$$

where

$$K = \left| \frac{\partial(r - \epsilon f)}{\partial x_i} \right|^{-1},$$

$$k = \frac{\sigma}{\mu_0 G a}, \qquad \lambda = \frac{\mu^*}{\mu_0}.$$

The notation of Frankel and Acrivos [6] is adopted here and is summarized below: μ_0, viscosity of the suspending fluid; μ^*, viscosity of the drop; σ, surface tension; a, radius of the undeformed drop; G, magnitude of the shear rate; e_{ij}, ω_i, rate of strain, vorticity of the undisturbed flow field; p_{ij}, u_i, stress, velocity in the flow field; n_i, unit vector directed along the outer normal to the surface of the drop;

R_1, R_2, principal radii of curvature of the surface of the drop; δ_{ij}, Kronecker symbol. All starred quantities refer to the discrete phase.

The velocities and stresses are given by Lamb's general solution (e.g. Eq. (1.1)). By requiring the velocity to be finite at both $r = 0$ and $r \to \infty$, this solution simplifies further in that it involves only positive order harmonics inside the droplet and negative order harmonics in the suspending medium (except for a contribution from χ_1 and Φ_2 which follows readily from the boundary condition (2.7) at infinity).

Correspondingly, the function f, describing the surface of the drop, is expanded in terms of surface spherical harmonics:

$$f = \sum_{n=0}^{\infty} \psi_n . \tag{2.8}$$

Because the unknown tensorial coefficients are very difficult to determine in the general case, only small deformations of the drop will be considered here. The problem can then be linearized by application of a standard regular perturbation technique. This procedure has alrealy been used, among others, by Chaffey, Brenner, and Mason [7], by Cox [8], and by Frankel and Acrivos [6], who have provided details of the method. The parameter ϵ, appearing in the equation of the surface, is supposed to be much smaller than unity, and, as shown by Cox, is $O(k^{-1})$ when $k \gg 1$ and λ is $O(1)$, or $O(\lambda^{-1})$ when $\lambda \gg 1$ and k is $O(1)$. All quantities of interest are expanded in terms of ϵ, hence the function f, describing the surface of the drop, becomes

$$f = f^{(0)} + \epsilon f^{(1)} + \epsilon^2 f^{(2)} + O(\epsilon^3),$$

where $f^{(0)}$, $f^{(1)}$, and $f^{(2)}$ are sums of surface harmonics. Similar expansions apply for the remaining variables, e.g. u_i and p_{ij}.

At each step of the perturbation analysis, a linear system involving the coefficients of the harmonics results when Lamb's expressions for the velocities and stresses inside and outside the drop are inserted into the boundary equations. For example, the matching of the velocities yields the following equation in the $O(\epsilon)$ approximation:

$$\begin{aligned} \epsilon_{lim} x_m (B_l + B'_{lqp} x_q x_p) + A_{lm} x_l x_m x_i + A'_{il} x_l + C_{lmpq} x_l x_m x_p x_q x_i \\ + C'_{ilmp} x_l x_m x_p = F_{lm}(D_{pq} x_p x_q x_i + D'_{ip} x_p)\, x_l x_m , \end{aligned} \tag{2.9}$$

where the A's, and B's and C's are linear combinations of the unknown $O(\epsilon)$ tensors, whereas the D's and F's are linear combinations of $O(1)$ tensors for which a solution has already been obtained. In this form, however, (2.9) is not readily amenable to solution, because it is evaluated at $r = 1$ and the x's are, thus, not independent. Frankel and Acrivos, therefore, developed a method for transforming such local relationships into algebraic equations. It consists chiefly of suitably integrating these expressions over a unit sphere, S, and taking advantage of the orthogonality conditions

$$\int_S x_i x_j \, d\Omega = \frac{4\pi}{3} \delta_{ij}, \qquad \int_S x_i x_j x_l x_m \, d\Omega = \frac{4\pi}{15} (\delta_{ij}\delta_{lm} + \delta_{il}\delta_{jm} + \delta_{jl}\delta_{im}), \text{ etc.,} \tag{2.10}$$

where $d\Omega$ is a solid angle. Thus, when (2.9) is first multiplied by $x_j \, d\Omega$, then integrated over S, it becomes, by virtue of (2.10),

$$\epsilon_{lij} B_l + \frac{2}{5} A_{ij} + A'_{ij} = \frac{8}{35} Sd(F_{il} D_{lj}) + \frac{2}{5} F_{jl} D'_{li} + \text{isotropic terms.} \tag{2.11}$$

No higher order tensors are involved in the above because they appear only in the form $C_{lmpq} \, \delta_{il} \, \delta_{jm} \, \delta_{pq}$ which reduces to C_{ijpp} and, therefore, vanishes as a consequence of the zero contraction requirement. Terms multiplied by δ_{ij} are called isotropic, and the symmetric deviator of a second order tensor is defined by

$$Sd(A_{ij}) = \tfrac{1}{2}(A_{ij} + A_{ji} - \tfrac{2}{3} \delta_{ij} A_{ll}).$$

Similarly, multiplication of (2.9) by $x_i x_a x_b \, d\Omega$ followed by integration, results in another equation of the form

$$A_{ab} + A'_{ab} = (4/7)\, Sd(F_{al} D_{lb}) + (4/7)\, Sd(F_{al} D'_{lb}).$$

To obtain a relationship between the fourth-order tensors, the same procedure is applied except that the multiplication factors are changed to $x_j x_a x_b \, d\Omega$ and $x_i x_j x_a x_b x_c \, d\Omega$. This yields, respectively,

$$\begin{aligned} &\epsilon_{lij} B'_{lab} + \epsilon_{lia} B'_{lbj} + \epsilon_{lib} B'_{lja} + (4/3)\, C_{ijab} + 3C'_{ijab} \\ &\quad = (4/9)\, Sd_4(F_{ij} D_{ab}) + F_{ab} D'_{ij} + F_{aj} D'_{ib} + F_{bj} D'_{ai} + \text{isotropic terms,} \end{aligned} \tag{2.12}$$

and

$$3(C_{cjab} + C'_{cjab}) = Sd_4(F_{jc} D_{ab}) + Sd_4(F_{jc} D'_{ab}),$$

where to any fourth-order tensor A_{ijab} is associated its symmetric deviator, a quantity which is perfectly symmetric, has zero contraction and is, thus, defined as

$$\begin{aligned} Sd_4(A_{ijab}) = (1/8)\{&A_{ijab} + A_{iabj} + 22 \text{ other terms} \\ &- (2/7)[\delta_{ab}(A_{ijll} + A_{ilјl} + 10 \text{ other terms}) + 5 \text{ other terms}] \\ &+ (8/35)(\delta_{ij}\, \delta_{ab} + \delta_{ia}\, \delta_{bj} + \delta_{ib}\, \delta_{ja})(A_{llmm} + A_{lmlm} + A_{lmml})\}. \end{aligned}$$

Continuing in this fashion and applying all the remaining boundary conditions leads then to a set of linear algebraic equations for the tensor coefficients appearing in the general solution.

3. An Outline of the REDUCE Program

As is evident from the above, the proposed method of solution, although straightforward in principle, can become rather impractical if the various algebraic equations relating the tensorial coefficients, e.g. (2.12) must be derived analytically by hand, since it is humanly impossible to avoid making mistakes, for example sign errors, which are hard to detect. It was decided, therefore, to perform most of the algebra involved in the successive steps of the solution using REDUCE. Our aims were twofold: first, to eliminate the human error element from tedious routine calculations such as those involving the sum and product of rational fractions; and, more importantly, to check the power and capabilities of such symbolic languages when tested against real physical problems.

The program was run on a PDP-10 time-shared computer, in a totally interactive mode. The tensorial coefficients of the various harmonics were considered as symmetric functions of their indices. The special properties of the Kronecker symbol could be easily implemented by defining a set of general substitution commands whereby $A_{il}\,\delta_{lj}$ was systematically replaced by A_{ij}.

The program began with the evaluation of the velocity and stress fields from Lamb's general solution, using the differentiation and simplification capabilities of REDUCE to compute the derivatives appearing in (1.1). Next these values were substituted into the appropriate boundary conditions. The $O(1)$ problem was solved first, and the computed values of the corresponding $O(1)$ quantities were inserted into the $O(\epsilon)$ equations. These were solved in turn, yielding expressions for the $O(\epsilon)$ tensors. Finally the $O(1)$ and $O(\epsilon)$ terms were replaced by their values in the $O(\epsilon^2)$ equations which were then solved to yield the $O(\epsilon^2)$ tensors.

An important feature of REDUCE, namely its ability to perform substitutions in a wide variety of forms, was found very useful for the purpose of integrating the various boundary conditions. Indeed, since all the integrals are already known from (2.10), it is possible to simply reduce the integration step to a series of substitution commands. For example, $F_{lm}D_{pq}x_l x_m x_p x_q x_i x_j\, d\Omega$ can be replaced by $(4\pi/105)\, F_{lm}D_{pq}(\delta_{lm}\,\delta_{pq}\,\delta_{ij} + \cdots)$ which can then be simplified and evaluated. Unfortunately, this straightforward procedure tends to generate enormous intermediate expressions because the number of terms increases exponentially with the order of the integrals. To avoid the resulting waste of storage space and computing time, it was noted that the symmetric/antisymmetric part of the integrated equations could be obtained by considering separately the symmetric/antisymmetric part of each individual term. Accordingly, each integral was evaluated by hand, partitioned into a symmetric and an antisymmetric part, and expressed in terms of deviators and isotropic terms. Therefore, in the above example, $F_{lm}D_{pq}x_l x_m x_p x_q x_i x_j \, d\Omega$ is directly replaced by

$$(4\pi/105)[8Sd(F_{il}D_{lj}) + (14/3)\,\delta_{ij}F_{lm}D_{lm}]$$

or by zero, depending on whether the symmetric or the antisymmetric part of the corresponding equation is computed. The resulting system of linear equations was of rank 4 only, and the matrix inversion routine could thus be used to solve it, with no substantial loss of efficiency. Throughout the program, the coefficients of the various tensorial quantities were complicated rational fractions, sometimes involving polynomials of degree 7, with integer coefficients of order 10^{12}. An automatic simplification of those expressions could be easily obtained by taking advantage of the greatest common divisor routine. To illustrate the type of results obtained from this program, we present here as an example the differential equation for the coefficient F_{ij} of the harmonic ψ_2 which appears in (2.8), the equation for the shape of the drop, for the case $\epsilon = k^{-1} \ll 1$ and $\lambda = O(1)$:

$$
\begin{aligned}
F_{ij} &+ \frac{(2\lambda+3)(19\lambda+16)}{40(\lambda+1)}\, k^{-1}\left[\frac{\partial F_{ij}}{\partial t} + \frac{\omega_p}{2}\,(\epsilon_{ipl}F_{lj} + \epsilon_{jpl}F_{li}\right] \\
&\left[\frac{19\lambda+16}{24(\lambda+1)} - \frac{(11172\lambda^4 + 18336\lambda^3 + 17440\lambda^2 + 3499\lambda - 7572)}{980(2\lambda+3)^2\,(\lambda+1)}\right. \\
&\left.\times\, k^{-2}F_{lm}F_{lm}\right] e_{ij} \\
&- \left[\frac{(\lambda-1)(22344\lambda^3 + 52768\lambda^2 + 45532\lambda + 19356\,)}{980(2\lambda+3)^2\,(\lambda+1)}\, F_{lm}e_{lm}\right. \\
&\left. + \frac{6C_7(\lambda)\, F_{lm}F_{lm}}{245(2\lambda+3)^2\,(19\lambda+6)^2\,(10\lambda+11)(17\lambda+16)(\lambda+1)}\right] k^{-2}F_{ij} \\
&+ \frac{(4\lambda-9)(19\lambda+16)}{28(2\lambda+3)(\lambda+1)}\, k^{-1}Sd(e_{il}F_{lj}) \\
&+ \frac{36(137\lambda^3 + 624\lambda^2 + 741\lambda + 248)}{35(2\lambda+3)(19\lambda+16)(\lambda+1)}\, k^{-1}Sd(F_{il}F_{lj}) \\
&+ \frac{6(\lambda-1)(2793\lambda^3 + 7961\lambda^2 + 8474\lambda + 3522)}{245(2\lambda+3)^2\,(\lambda+1)}\, k^{-2}Sd(e_{il}F_{lm}F_{mj}) \\
&- \frac{10(43\lambda^2 + 79\lambda + 53)}{3(2\lambda+3)(\lambda+1)}\, k^{-2}F_{ijlm}e_{lm} \\
&+ \frac{2C_5(\lambda)\, k^{-2}F_{ijlm}F_{lm}}{(2\lambda+3)(19\lambda+16)(10\lambda+11)(17\lambda+16)(\lambda+1)} + O(k^{-3}),
\end{aligned}
$$

where

$$C_7(\lambda) = 2127976\lambda^7 - 16341920\lambda^6 - 38494964\lambda^5 + 122942551\lambda^4 + 474068311\lambda^3 + 591515680\lambda^2 + 332123136\lambda + 71700480,$$

and

$$C_5(\lambda) = 405260\lambda^5 + 2366960\lambda^4 + 9142173\lambda^3 + 8595967\lambda^2 + 3334160\lambda + 693760.$$

All the terms in the above equations are dimensionless, and F_{ijlm} is the coefficient of the ψ_4 harmonic which obeys a similar relation. The expression for all the $O(\epsilon)$ tensors, and those for the $O(\epsilon^2)$ tensors of rank two are given by Barthès-Biesel [9].

Thus, using REDUCE, an analytic solution was obtained to a problem which, due to its complexity, appeared quite untractable if done by hand. Of course, no claim is made that this is the first time a symbolic language has been used to solve a physical problem. In fact, we should mention the contribution of Campbell and Hearn [10] in quantum electrodynamics and the work of Howard and Tashjian [11], who used FORMAC to generate automatically the equations of mathematical physics (e.g. the Navier–Stokes equations) in any curvilinear coordinate system.

In conclusion, it seems that many physical problems could be successfully solved using REDUCE, or FORMAC, or some other general purpose symbolic language. For example, the same problem which was considered here could be studied for the case of potential flow. Another interesting application of REDUCE would consist in generating automatically a power series solution (e.g. Blasius series) to the boundary layer equations [12], for a variety of bodies and outer velocity profiles.

Thus, it appears that, before too long, REDUCE or some suitable modification thereof, will evolve into a standard tool for performing most of the routine analytic manipulations that, to date, have been commonly carried out by hand for many problems in mathematical physics and in fluid mechanics.

Acknowledgments

This work was supported in part by a grant from NASA-Ames on Numerical Fluid Mechanics. The authors wish to acknowledge the precious help of A. C. Hearn who not only developed the original version of REDUCE but who was also kind enough to expand the program to facilitate tensorial operations. The authors are further indebted to John McCarthy for making available to them the computer of the Artificial Intelligence Center at Stanford.

References

1. C. Engelman, 1965 MATHLAB: a program for on-line assistance in symbolic computations. Proceeding 1965 FJCC, Spartan Books, Washington, D. C.

2. M. Engeli, 1968 User's manual for the formula manipulation language SYMBAL. The University of Texas at Austin Computation Center, Austin, Texas.
3. R. G. Tobey *et al.*, 1967 PL/I FORMAC Interpreter, User's Reference Manual. IBM contributed program library, 360 D03.3004, Hawthorn, New York.
4. A. C. Hearn, 1970 REDUCE 2 User's Manual, Stanford Artificial Intelligence Project, Memo AIM-133.
5. H. Lamb, "Hydrodynamics," 6th edition, Cambridge University Press, London/New York, 1932.
6. N. A. Frankel and A. Acrivos, The constitutive equation for a dilute emulsion, *J. Fluid Mech.* **44** (1970), 65–78.
7. C. E. Chaffey, H. Brenner, and S. G. Mason, Particle motions in sheared suspensions XVIII. Wall migration (theoretical), *Rheol. Acta*, **4** (1965), 64–72; Correction, *Rheol. Acta*, **6** (1967), 100.
8. R. G. Cox, The deformation of a drop in a general time-dependent fluid flow, *J. Fluid Mech.*, **37** (1969), 601–623.
9. D. Barthès-Biesel, Ph.D. Dissertation, Stanford University, 1972.
10. J. A. Campbell and A. C. Hearn, Symbolic analysis of Feynman diagrams by computer, *J. Comp. Phys.* **5** (1970), 280.
11. J. C. Howard and H. Tashjian, An algorithm for deriving the equations of mathematical physics by symbolic manipulation, *Comm. ACM* **11** (1968), 814–819.
12. H. Schlichting, "Boundary-Layer Theory," 6th edition, McGraw-Hill, New York, 1955.