Comp. & Maths. with Appls. Vol. 12A. No. 3. pp. 345–351. 1986
Printed in Great Britain.

0886-9553/86 \$3.00 + .00

BENCHMARK FLUID FLOW PROBLEMS FOR CONTINUOUS SIMULATION LANGUAGES

S. THOMPSON† and P. G. TUTTLE
Babcock and Wilcox, Power Generation Division, Lynchburg, Virginia 24505, U.S.A.

(*Received August* 1985)

Communicated by E. Y. Rodin

Abstract—Two benchmark problems for continuous simulation languages are discussed. The use of the Advanced Continuous Simulation Language (ACSL) and the sparse ordinary differential equation solver DSTPGT, which has been incorporated into ACSL, are discussed for the solution of these fluid flow problems. The one-dimensional incompressible Navier–Stokes partial differential equations are discretized spatially using the method of pseudo-characteristics. The resulting sparse system of ordinary differential equations is then solved using the method of lines. A continuous-space-discrete-time solution is also given in order to illustrate the use of the DSTPGT special event detection mechanism (root-finding) in ACSL. The discussions illustrate several important considerations related to the solution of complex fluid problems or, more generally, to sparse systems and/or systems requiring the detection and processing of special events.

NOTATION

ρ Density (kg/m^3).
G Flow rate (kg/m^2 s).
T Temperature (°C).
K Frictional pressure drop coefficient = 10.0.
g_a Gravitational acceleration = 9.80665 (m/s^2).
θ 90°.
Φ Heat flux = 1.1E5 (w/m^2).
P_H Heated perimeter = 7.97318E+2 (m).
A_f Flow area = 3.82760 (m^2).
$\underline{L}$ 1.0 (m).
$\overline{T}$ Absolute temperature = T + 273.15 (°K).
p Pressure [MPa (10^6 Pa)].
v Specific volume (m^3/kg).
h Specific enthalpy (kJ/kg).
s Specific entropy (KJ/Kg-K).
C_p^{-1} Reciprocal of constant pressure specific heat (kg-K/kJ).
κ^{-1} Reciprocal of isothermal compressibility [MPa (10^6 Pa)].
β^{-1} Reciprocal of coefficient volume expansion (K).
a Sound speed (m/s).

$$C_p^{-1} = \left.\frac{\partial T}{\partial h}\right|_p$$

$$\kappa^{-1} = -v\left.\frac{\partial p}{\partial v}\right|_T$$

$$\beta^{-1} = v\left.\frac{\partial T}{\partial v}\right|_p$$

$$a = \left(\left.\frac{\partial p}{\partial \rho}\right|_s\right)^{1/2}$$

INTRODUCTION

There are a number of very good languages for solving continuous simulation problems. One such language is the Advanced Continuous Simulation Language (ACSL)[1]. The well-known ordinary differential equation (ODE) solver DSTPGT[2] has been incorporated in ACSL. DSTPGT

†Present address: Martin–Marietta Energy Systems, Oak Ridge National Laboratory, P. O. Box Y, Bldg. 9207-A, Oak Ridge, TN 37831.

has been used for several years to solve the extremely complex systems that arise in modeling nuclear reactors. Representative problems include core reflood analysis, vapor growth in a time-dependent pressure field[3], coupled heat exchanger performance, anticipated upset transient analysis, subchannel analysis, kinetics problems, pressurizer models, steam generator analysis, and structural mechanics problems[4]. The solution of these difficult, real-world problems represents a truly difficult challenge for simulation languages for several reasons.

Typically, these systems are large and somewhat sparse and are characterized by the presence of discontinuities and special events which must be detected and correctly processed in order to obtain a solution (e.g. opening and closing of relief valves, moving boundaries, system trips and switches, and changes in the underlying equations)[4]. DSTPGT evolved from the excellent GEARS[5] sparse ODE solver. However, it was necessary to extensively modify the original software in order to accommodate the requirements above, to incorporate stiffly stable Adams-type methods[6,7] and to incorporate special debugging aids and other techniques[4]. All results discussed in this paper were obtained using ACSL and DSTPGT. Any other simulation language which contains provisions for solving stiff systems and for event detection may be used in lieu of this combination.

We sometimes receive requests from ODE software developers and proprietors of continuous simulation languages for descriptions of representative problems which may be used for benchmarking their products. Unfortunately, the level of complexity and proprietary restrictions make it impossible to provide compact descriptions. In this article, two benchmark problems which represent worthy challenges to any software with which we are familiar will be formulated and solved using the DSTPGT option in ACSL. Of necessity, these benchmark problems are not as complex as the problems which we must actually solve on a regular basis. On the other hand, they do exhibit many of the difficulties present in everyday models. Furthermore, the solution technique which is described is applicable to more general problems.

The first benchmark problem is obtained by using the method of pseudo-characteristics[8] to discretize spatially a boundary value problem for the one-dimensional Navier–Stokes equations. (The system is a mock-up for the subcooled liquid region for a steam generator model.) An arbitrary guess is used to initialize the system which is then integrated to steady state using ACSL and the method of lines[9]. This problem demonstrates the system sparsity and stiffness that typically arise in fluids problems. It also shows, even for this simple case, a rapid initial transient that approaches steady state.

The second benchmark problem solves the same physical problem. However, in this case the special conditions at steady state are utilized to write a system of ODE's in space. The primary difficulty encountered in this second problem is locating the spatial point at which saturation occurs (i.e. the end of the subcooled region). Locating this point accurately requires the ability to detect and process special events.

In addition to providing useful benchmark problems, these examples illustrate what can be done to get around the pitfalls and difficulties that commonly arise in the simulation of real-world problems. An expanded version of this paper is available from the authors. The expanded version contains complete FORTRAN listings for the model definitions.

FORMULATION OF THE MATHEMATICAL PROBLEM

The following formulation of the one-dimensional Navier–Stokes equations will be considered:

$$\frac{\partial U}{\partial t} + A \cdot \frac{\partial U}{\partial z} = C \quad (0 \leq t, 0 \leq z \leq L) \tag{1}$$

where

$$U = (\rho, G, T)^T \tag{2}$$

$$C = \left(0, \ -KG|G/\rho| - \rho g_a \sin\theta, \ \frac{a^2 \Phi P_H \kappa}{C_p A_f}\right)^T \tag{3}$$

$$A = \begin{pmatrix} 0 & 1 & 0 \\ \dfrac{1}{\rho\kappa} - \dfrac{G^2}{\rho^2} & 2\dfrac{G}{\rho} & \dfrac{\beta}{\kappa} \\ \dfrac{-a^2\beta\overline{T}G}{\rho^2 C_p} & \dfrac{a^2\beta\overline{T}}{\rho C_p} & \dfrac{G}{\rho} \end{pmatrix} \tag{4}$$

and

$$a,\ \kappa,\ \beta,\ C_p = f(T,\ \rho)\ \text{(equation of state)}. \tag{5}$$

The following boundary conditions will be used:

$$\begin{aligned} \rho(0,\ t) &= \rho_0 = 795.521, \\ T(0,\ t) &= T_0 = 255.000, \\ G(L,\ t) &= G_0 = 270.900. \end{aligned} \tag{6}$$

FORMULATION OF THE SPATIALLY DISCRETIZED PROBLEM

The above problem can be solved by substituting finite difference approximations for the spatial (z) derivatives in Eq. (1) and integrating the resulting discretized system of ODE's with respect to time until steady state is achieved. This so-called method of lines[9] approach permits the use of simulation languages such as ACSL for the solution.

A partition $z_0, \ldots, z_M$ with $z_i = Li/M$ $(i = 0, \ldots, M)$ is first defined. After the spatial differences are defined at each spatial node z_i, and the boundary conditions are applied, there results a system containing $3 \cdot M$ ODE's.

The flow equations constitute a first-order quasi-linear hyperbolic system of equations. Consequently, the spatial derivative approximations must be done with care to avoid the formation of shocks which prevent the achievement of steady state. Many commonly used approximations simply will not work for this problem. The approach taken here is to use a pseudo-characteristic formulation[8]. This may be done as follows:

The eigenvalues of A are seen to be

$$G/\rho,\ G/\rho + a,\ \text{and}\ G/\rho - a. \tag{7}$$

In the usual method of characteristics solution, one would first reduce the equations to characteristic form by diagonalizing A. This requires finding a nonsingular matrix **B** for which

$$\mathbf{B}A\mathbf{B}^{-1} = \mathbf{D} \tag{8}$$

where **D** is a diagonal matrix whose diagonal elements are the above eigenvalues. A straightforward derivation shows that one such matrix is:

$$\mathbf{B} = \begin{pmatrix} \beta a^2\overline{T} & 0 & -\rho C_p \\ -G\kappa a + 1 & \rho\kappa a & \rho\beta \\ G\kappa a + 1 & -\rho\kappa a & \rho\beta \end{pmatrix} \tag{9}$$

Multiplying the terms in Eq. (1) by this matrix gives the following characteristic form of the equations:

$$\mathbf{B} \cdot \frac{\partial U}{\partial t} + \mathbf{D} \cdot \mathbf{B} \cdot \frac{\partial U}{\partial z} = \mathbf{B} \cdot C \tag{10}$$

The idea behind the pseudo-characteristic solution for (10) is as follows. At each spatial node, upwind difference approximations[8] are calculated for the spatial derivatives:

$$\rho_{z,0},\ G_{z,0},\ T_{z,0}$$
$$\rho_{z,+},\ G_{z,+},\ T_{z,+}$$
$$\rho_{z,-},\ G_{z,-},\ T_{z,-}$$

(Here, ρ_z means $\partial\rho/\partial z$, for example.) The subscripts 0, + and − indicate that upwind differences are computed with the direction of the differencing dictated by the sign of the local characteristics G/ρ, $G/\rho + a$, $G/\rho - a$, respectively. For each local characteristic, backward differences are used if the characteristic is positive; otherwise forward differences are used (hence the terminology pseudo-characteristic method).

When the resulting values are substituted into the characteristic equation (10), there results a linear system of three equations in the three unknowns:

$$\partial\rho/\partial t,\ \partial G/\partial t, \text{ and } \partial T/\partial t$$

at each node. At node z_i the 3 × 3 system of linear equations to be solved is

$$\mathbf{B} \cdot (\partial\rho_i/\partial t,\ \partial G_i/\partial t,\ \partial T_i/\partial t)^T = \mathbf{E},$$

where **B** in (9) is evaluated at z_i using ρ_i, G_i, and T_i, and the vector **E** is defined in the following manner using C in (3) and the upwind differences:

$$\mathbf{E}_1 = \sum_{j=1}^{3} \mathbf{B}_{1j} \cdot C_j - (G_i/\rho_i) \cdot \{\mathbf{B}_{11}\, \rho_{z,0} + \mathbf{B}_{12}\, G_{z,0} + \mathbf{B}_{13}\, T_{z,0}\},$$

$$\mathbf{E}_2 = \sum_{j=1}^{3} \mathbf{B}_{2j} \cdot C_j - (G_i/\rho_i + a_i) \cdot \{\mathbf{B}_{21}\, \rho_{z,+} + \mathbf{B}_{22}\, G_{z,+} + \mathbf{B}_{23}\, T_{z,+}\}, \tag{11}$$

$$\mathbf{E}_3 = \sum_{j=1}^{3} \mathbf{B}_{3j} \cdot C_j - (G_i/\rho_i - a_i) \cdot \{\mathbf{B}_{31}\, \rho_{z,-} + \mathbf{B}_{32}\, G_{z,-} + \mathbf{B}_{33}\, T_{z,-}\}$$

The solution is $(\partial\rho_i/\partial t,\ \partial G_i/\partial t,\ \partial T_i/\partial t)^T$ which defines the temporal derivatives for the ODE solver.

DISCUSSION OF RESULTS FOR THE SPATIALLY DISCRETIZED PROBLEM

In this section, the results of solving the first benchmark problem using DSTPGT and ACSL will be discussed.

The initial transient for this problem is extremely rapid, as indicated by the graph of the calculated inlet flow rate $G(0, t)$ in Fig. 1, the outlet temperature $T(1, t)$ in Fig. 2, and the outlet density $\rho(1, t)$ in Fig. 3. (The plots are given only for $0 \le t \le 0.1$. The variables in question are virtually constant for $t > 0.1$.)

We emphasize the fact that even though the objective is to obtain a steady-state solution for the problem, the discretized system of ODE's initially has an extremely rapid oscillatory transient. For $M + 1 = 21$ spatial nodes, the transient settles down around $t = 0.01205$. (Note: This problem may also be used to assess the performance on complex problems by codes with automatic stiffness detection and integration method switching.) However, several hundred (in some cases, several thousand) steps are required to track the solution to this time. A steady-state solution is then quickly achieved. The initial guess for the solution was obtained using a linear rise for temperature from $z = 0$ to $z = L$ and using a constant pressure to calculate the corresponding density. Solutions were calculated for several successively refined spatial meshes using two-point spatial differences. Table 1 illustrates the effect of the mesh size for these solutions. The results illustrate the first-order spatial convergence of the solution. Observe that

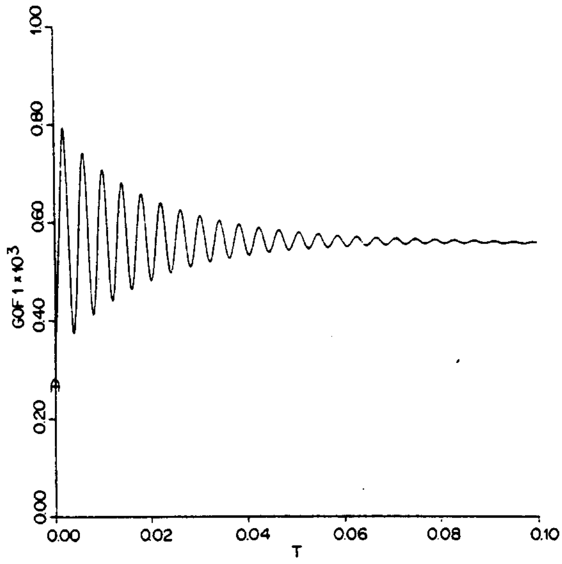

Fig. 1. Inlet flow rate $G(0, t)$.

the problem is not as easy as it might first appear since 41 spatial mesh points (120 ODE's) are required to obtain a maximum relative flow rate error of about 10%; and 81 spatial mesh points (240 ODE's) are required for a relative error of 5%.

Observe that the execution time increases rather dramatically as the spatial mesh size is reduced. The size of the ODE system roughly doubles each time the mesh size is halved. It is well known that the system stiffness also increases. This may be illustrated by considering the eigenvalues of the final Jacobian matrix. Eigenvalues were calculated using the ACSL linear analysis option[1]. For $M = 6$, 11, and 21, the stiffness ratio:

$$\max_i |\mathrm{Re}(\lambda_i)| / \min_i |\mathrm{Re}(\lambda_i)|$$

of the corresponding system is approximately 10,731.3, 24,557.4, and 46,357.8, respectively. Thus, the problem stiffness roughly doubles each time the mesh size is halved. This accounts for the increase in execution time as the number of mesh points increases.

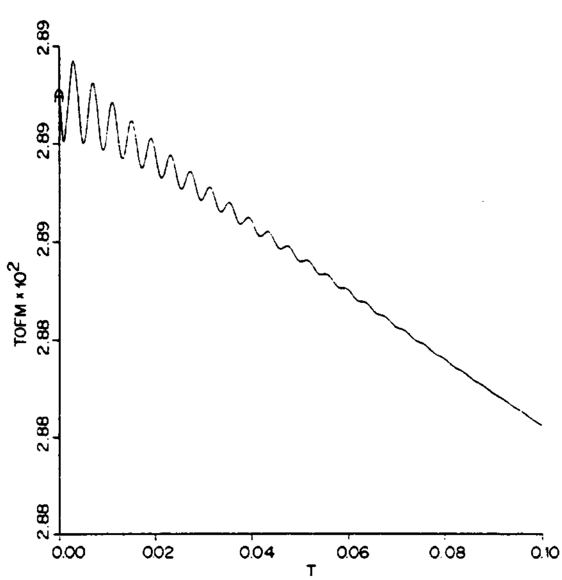

Fig. 2. Outlet temperature $T(1, t)$.

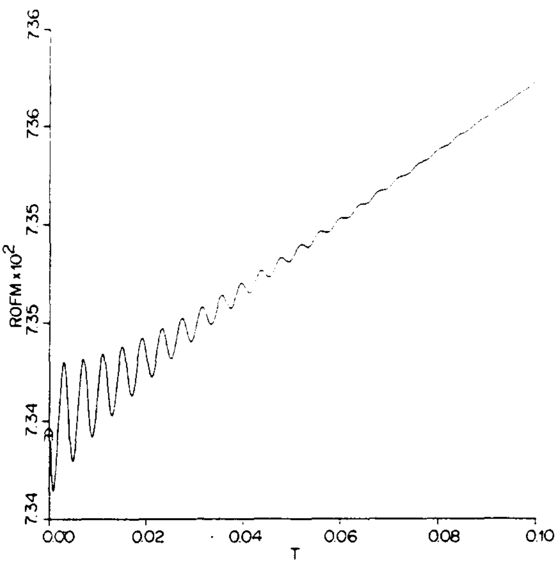

Fig. 3. Outlet density ρ(1, t).

CONTINUOUS-SPACE-DISCRETE-TIME (CSDT) SOLUTION

At steady state, $\partial\rho/\partial t = 0$ implies that $\partial G/\partial z = 0$ so that $G(z)$ is a constant, say G_0. In this case, the defining partial differential equations may be reduced to:

$$\begin{pmatrix} \frac{1}{\rho\kappa} - \frac{G_0^2}{\rho^2} & \frac{\beta}{\kappa} \\ \frac{-a^2\beta\bar{T}G_0}{\rho^2 C_p} & \frac{G_0}{\rho} \end{pmatrix} \begin{pmatrix} d\rho/dz \\ dT/dz \end{pmatrix} = \begin{pmatrix} -KG_0|G_0/\rho| - \rho g_a \sin\theta \\ \frac{a^2\Phi P_H \kappa}{C_p A_f} \end{pmatrix} \tag{12}$$

This constitutes a system of two first-order ODE's with independent variable z. For given values of ρ and T, the corresponding linear system may be solved to obtain $d\rho/dz$ and dT/dz. Given ρ(0) and $T(0)$, the system may, therefore, be integrated in z (i.e. in space) to obtain the steady-state spatial profiles of ρ and T. Note that this continuous-space-discrete-time solution provides a means of obtaining an "exact" solution for the (nondiscretized) benchmark problem (1).

Consider the problem of integrating this system until saturated values of T and ρ are obtained (i.e. until the "end" of the subcooled liquid region is located). The event mechanism in ACSL may be used to accomplish this via the SKEDSE macro[1]. That is, DSTPGT may be used to integrate the system until a root of the equation:

$$0 = g(\rho, T) = \rho_{sat}(T) - \rho$$

Table 1. Selected results for two-point spatial differences

Number of spatial mesh points ($M + 1$)	Maximum error in calculated steady-state flow rates	Execution time (CPU seconds, Cyber 855)
3	252.901	7.5
5	162.011	12.4
11	93.701	26.8
21	53.127	73.2
41	29.231	226.9
81	14.876	855.5
101	12.030	1,270.5

is located where $\rho_{sat}(T)$ is the saturated density corresponding to T. Alternatively, the event equation $g(\rho, T)$ may be used where

$$g(\rho, T) = \begin{cases} -1, & \text{if } \rho \text{ and } T \text{ are not saturated} \\ +1, & \text{if } \rho \text{ and } T \text{ are saturated.} \end{cases}$$

For the initial conditions $\rho(0) = 795.521$ and $T(0) = 255.000$, the value of z at which saturation first occurs is found by DSTPGT to be $z = 2.0953$, at which $\rho = 733.12$ and $T = 289.57$.

Other boundary conditions lead to nonlinear problems to which two-point boundary value techniques may be applied. These problems in fact represent very good benchmark problems for two-point boundary value solvers.

SUMMARY

In this article, two benchmark fluid flow problems were described. These problems provide a means of testing ODE software and simulation languages. They also illustrate the pitfalls and difficulties which practitioners must consider in the solution of complex problems. To illustrate the manner in which the problems may be solved, both were solved using the DSTPGT sparse integration and event detection options in ACSL.

Acknowledgments—The authors gratefully acknowledge the constructive criticisms of three reviewers whose comments significantly improved the exposition of this paper. The first author is also indebted to the stability expert, T. R. Hatcher, for influencing his thinking on techniques for approaching steady state following extremely rapid transients.

REFERENCES

1. *Advanced Continuous Simulation Language (ASCL): User Guide/Reference Manual*. Mitchell and Gauthier Associates, Inc., Concord, Massachusetts (1981).
2. S. Thompson and P. G. Tuttle, Automatic O.D.E. Software for Use in an Industrial Environment, in *Modeling and Simulation in Engineering* (Edited by W. F. Ames, *et al.*), pp. 53–56. IMACS/North-Holland Publishing Company (1983).
3. F. Aguilar and S. Thompson, A non-equilibrium flashing model for rapid pressure transients. *ASME Paper* 81–*AT*–35 (1981).
4. S. Thompson and P. G. Tuttle, "The solution of several stiff problems in an industrial environment: The evolution of an O.D.E. solver," International Conference on Stiff Computation, *Proceedings* 3, Park City, Utah (1982).
5. A. C. Hindmarsh, "Toward a systematized collection of ODE solvers," 10th IMACS Congress on System Simulation and Scientific Computation, *Proceedings* 1, pp. 427–432, Montreal, Canada (1982).
6. D. J. Rodabaugh and S. Thompson, Low-Order A_0-Stable Adams Methods. *J. Comput. Appl. Math.*, 5, No. 3, 225–233 (1979).
7. S. Thompson, Stiffly stable fourth order Adams methods. *J. Comput. Appl. Math.*, 8, No. 4, 253–256 (1982).
8. M. B. Carver, Pseudo-characteristic method of lines solution of first-order hyperbolic equation systems, in *Advances in Computer Methods for Partial Differential Equations III. Proceedings of the Third IMACS International Symposium on Computer Methods for Partial Differential Equations*, pp. 227–230. Bethlehem, Pennsylvania (1979).
9. M. B. Carver, Method of lines solution of partial differential equations: Some standard and devious applications, in *Computational Methods in Nuclear Engineering. Proceedings of the American Nuclear Society Topical Meeting*, Vol. 3, pp. 8.53–8.76. Williamsburg, Virginia (1979).