AIAA JOURNAL
Vol. 31, No. 4, April 1993

Technical Notes

TECHNICAL NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Noniterative Implicit Method for Tracking Particles in Mixed Lagrangian-Eulerian Formulations

T. I-P. Shih* and A. Dasgupta†
Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213

Introduction

MULTIPHASE problems such as particle-laden flows and sprays are often analyzed by using a mixed Lagrangian-Eulerian formulation. In this approach, particles—liquid droplets or solid matter (henceforth referred to as the dispersed phase)—are described from the Lagrangian point of view, and the surrounding fluid (henceforth referred to as the continuum phase) is described from the Eulerian point of view. The governing equations for the dispersed phase is a system of ordinary differential equations (ODEs), and those for the continuum phase are the Navier-Stokes equations. As used here, the Navier-Stokes equations include the continuity, momentum, and energy equations.

The system of ODEs describing the dispersed phase is an initial-value problem (IVP). However, this IVP differs from typical IVPs in that it is coupled to the continuum-phase equations through terms that describe the exchange of mass, momentum, and energy between the dispersed and continuum phases. Typically, the data used to compute the Lagrangian formulation lag behind those in the Eulerian formulation by one time step. This procedure works well if the time-step size used is small when compared with the time scales relevant to the problem.

By lagging behind one time step, any explicit method (e.g., the second- or fourth-order accurate Runge-Kutta methods) can be used to generate solutions to the Lagrangian formulation (see, e.g., Raju and Sirgnano[1]). This is because for explicit methods, data from the Eulerian formulation are needed only at the current time level where the location of each particle or group of particles is known so that those data can be obtained by interpolating the solution of the Eulerian formulation. Though explicit methods can be implemented easily, they can impose severe limitations on the maximum time-step sizes that can be used. As will be shown later, the smaller the particle size, the more severe is this limitation on the time-step size. If the time-step size limitation is too severe, then implicit methods are needed.

For the current IVP, the construction of implicit methods is complicated by the fact that data from the Eulerian formulation are needed at both the current and new time level. Since the location of each particle or group of particles at the new time level is unknown, such data cannot be obtained by interpolation. Despite this difficulty, implicit methods for the current type of IVPs have been constructed (see, e.g., Dukowicz[2]). However, methods constructed so far are complicated and iterative in nature. Also, they are designed to be used with pressure-based algorithms (e.g., SIMPLE[3] and PISO[4]). In this study, a noniterative implicit method is presented that can be used with pressure- as well as density-based algorithms.

Received May 2, 1992; revision received Sept. 9, 1992; accepted for publication Sept. 14, 1992.

*Associate Professor, Department of Mechanical Engineering. Member AIAA.

†Graduate Student, Department of Mechanical Engineering.

Methodology

To illustrate the noniterative implicit method developed in this study, consider a dispersion of noninteractiing solid particles in an isothermal flow in which the added-mass effect can be neglected. For such a dispersion, the following six coupled ODEs govern the position and motion of the ith particle:

$$\frac{d\boldsymbol{r}_i}{dt} = \boldsymbol{v}_i \tag{1}$$

$$\rho_i \vartheta_i \frac{d\boldsymbol{v}_i}{dt} = \boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{v}_i) \tag{2}$$

where

$$\boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{v}_i) = F_x\boldsymbol{i} + F_y\boldsymbol{j} + F_z\boldsymbol{k} = C_{Di}\frac{\rho}{2}(\boldsymbol{V} - \boldsymbol{v}_i)|\boldsymbol{V} - \boldsymbol{v}_i|A_i \tag{3}$$

$$\boldsymbol{r}_i = x_i\boldsymbol{i} + y_i\boldsymbol{j} + z_i\boldsymbol{k} \tag{4}$$

$$\boldsymbol{v}_i = u_i\boldsymbol{i} + v_i\boldsymbol{j} + w_i\boldsymbol{k} \tag{5}$$

In the foregoing equations, ρ_i, $\boldsymbol{v}_i$, and $\boldsymbol{r}_i$ are, respectively, the density, velocity, and position vector of the ith particle; ρ and $\boldsymbol{V}$ are, respectively, the density and velocity of the continuum phase at $\boldsymbol{r}_i$; $\boldsymbol{F}_i$ is the net force acting on the ith particle due to interphase drag [note that $\boldsymbol{F}_i = \boldsymbol{F}_i(\boldsymbol{r}_i, \boldsymbol{v}_i)$ because $\boldsymbol{V} = \boldsymbol{V}(\boldsymbol{r}_i)$]; and C_{Di} is the interphase drag coefficient. Finally, A_i and ϑ_i are, respectively, the cross-sectional area and volume of the ith particle.

The essence of the new implicit method is as follows. By using a second-order accurate in time implicit time-differencing formula, Eqs. (1) and (2) become

$$\frac{\Delta\boldsymbol{r}_i^{n+1}}{\Delta t} = \frac{1}{2}(\boldsymbol{v}_i^n + \boldsymbol{v}_i^{n+1}) \tag{6}$$

$$\rho_i \vartheta_i \frac{\Delta\boldsymbol{v}_i^{n+1}}{\Delta t} = \frac{1}{2}(\boldsymbol{F}_i^n + \boldsymbol{F}_i^{n+1}) \tag{7}$$

where

$$\Delta\boldsymbol{r}_i^{n+1} = \boldsymbol{r}_i^{n+1} - \boldsymbol{r}_i^n \tag{8}$$

$$\Delta\boldsymbol{v}_i^{n+1} = \boldsymbol{v}_i^{n+1} - \boldsymbol{v}_i^n \tag{9}$$

In the foregoing equations, the superscript n denotes the current time level, and the superscript $n + 1$ the new time level. The unknowns in these same equations are $\boldsymbol{r}$ and $\boldsymbol{v}$ at the $(n + 1)$th time level. The difficulty in solving the aforementioned equations is associated with $\boldsymbol{F}^{n+1}$ in Eq. (7) that involve quantities from the continuum phase.

Equations (6) and (7) form a system of six nonlinear algebraic equations. A solution to this nonlinear system can be obtained by using the Newton-Raphson method that requires iteration at each time step. Alternatively, we can linearize this nonlinear system so that iterations are not required.

The term creating the nonlinearities is $\boldsymbol{F}^{n+1}$ in Eq. (7). This term can be linearized in time by using the Taylor series (i.e., $\boldsymbol{F} = \boldsymbol{F} +$

$(\partial F/\partial t)\Delta t + \Delta t^2)$ and noting that $(\partial F/\partial t) = (\partial F/\partial r)(\partial r/\partial t) + (\partial F/\partial v)(\partial v/\partial t)$ to give

$$F_i^{n+1} = F_i^n + \frac{\partial F_i^n}{\partial r_i^n}\Delta r_i^{n+1} + \frac{\partial F_i^n}{\partial v_i^n}\Delta v_i^{n+1} \tag{10}$$

where Δr^{n+1} and Δv^{n+1} are given by Eqs. (8) and (9), and

$$\frac{\partial F_i^n}{\partial v_i^n} = \begin{bmatrix} \frac{\partial F_x}{\partial u} & \frac{\partial F_x}{\partial v} & \frac{\partial F_x}{\partial w} \\ \frac{\partial F_y}{\partial u} & \frac{\partial F_y}{\partial v} & \frac{\partial F_y}{\partial w} \\ \frac{\partial F_z}{\partial u} & \frac{\partial F_z}{\partial v} & \frac{\partial F_z}{\partial w} \end{bmatrix} \tag{11}$$

$$\frac{\partial F_i^n}{\partial r_i^n} = \frac{\partial F_i^n}{\partial V_i^n}\frac{\partial V_i^n}{\partial r_i^n} \tag{12}$$

The linearization just given is second-order accurate in time. Since this is the same order of accuracy as the time-differencing formula, the error incurred by linearization is comparable to that created by time-differencing. This justifies the linearization procedure.

The terms in Eq. (11) can readily be obtained by analytically differentiating Eq. (3) with respect to u, v, and w (the x, y, and z components of the particle velocity). In a similar manner, the partial derivative of F with respect to V that appear in Eq. (12) can easily be obtained. The difficulty is in evaluating the partial derivative of V with respect to r that appear in Eq. (12). This difficulty arises because V is given within the Eulerian framework, whereas r is provided within the Lagrangian framework. Here, this term is evaluated as follows:

$$\frac{\partial V_i^n}{\partial r_i^n} = \frac{v_i^n}{|v_i^n|} \cdot \nabla V_i^n \tag{13}$$

In the foregoing equation, the derivative on the left-hand side is Lagrangian, and the gradient operator on the right is Eulerian. The dot product gives the component of the Eulerian gradient in the direction where the ith particle is moving.

This completes our description of the implicit algorithm developed in this study.

Results

To examine the implicit algorithm developed in this study, numerical solutions were obtained for a dilute particle-laden flow in a passage bounded by one straight wall and one wavy wall in which all particles were spherical and had a finite velocity relative to the continum phase at the inflow boundary so that the right-hand side of Eq. (2) does not vanish. Figure 1a shows the two-dimensional spatial domain and grid system used. The grid system used was generated by an algebraic grid-generation technique based on transfinite interpolation.[5] For this test problem, it was assumed that only the continuum phase can affect the dispersed phase, i.e., the interaction is one-way. As a result, the velocity field for the continuum phase can be obtained independently of the dispersed phase. Here, the velocity field was obtained by the PISO algorithm[4] with QUICK[6] differencing used for the convection terms. The velocity field obtained is shown in Fig. 1b. Note that this velocity field contains regions with recirculating flows.

Solutions for the dispersed phase were obtained by the implicit algorithm described in the previous section and the second-order Runge-Kutta explicit method. The code embodying the explicit method was developed by Raju and Sirignano.[1] Solutions were obtained by this code because it has been tested and can be used to assess the robustness, accuracy, and efficiency of the implicit algorithm developed in this study.

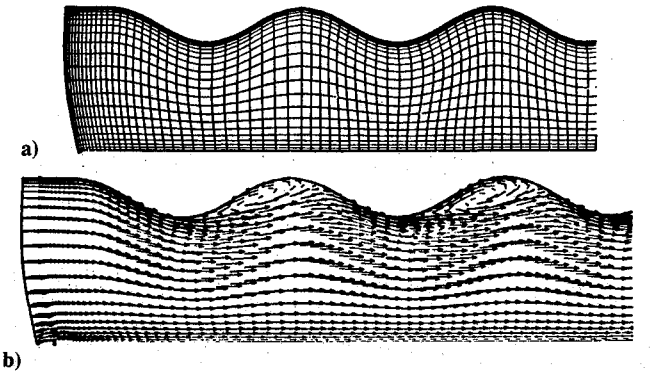

Fig. 1 Test problem: a) spatial domain and grid system, b) velocity field.

When employing the explicit code of Raju and Sirignano for the test problem, the maximum time-step size Δt_{max} that can be used in order to obtain stable numerical solutions was found to be a function of the particle radius r_p as shown here:

r_p (μm):	10.0	1.0	0.1	0.01	0.001
Δt_{max} (s):	10^{-5}	10^{-7}	10^{-9}	10^{-11}	10^{-13}

When the implicit algorithm was used for the test problem, numerical experiments indicated that a time-step size of 2×10^{-4} s can be used with a particle radius as small as 0.001 μm. Thus, the implicit method developed permits much larger time-step sizes.

The reason Δt_{max} decreases with r_p for explicit methods has to do with the stability criterion illustrated next. Suppose C_{Di} in Eq. (3) is given by $64/Re = 32\mu/\rho|V - v_i|r_p$ where μ is the dynamic viscosity of the continuum phase at r_i. For the Euler explicit method that is the first step of the second-order Runge-Kutta method, it can easily be shown that the stability criterion is given by

$$\Delta t \le \frac{\rho_i r_p^2}{16\mu} \tag{14}$$

Equation (14) clearly shows that as r_p decreases, the maximum Δt that can be used decreases. It turns out that the foregoing stability criterion is consistent with the results obtained by using the code of Raju and Sirignano described earlier.

With the robustness of the implicit algorithm established, we now discuss its accuracy. The accuracy of the implicit algorithm was tested by comparing results that it generates with those obtained by the explicit code of Raju and Sirignano. These tests used particles with a radius of 10 μm so that a time-step size of 10^{-5} s can be used by both algorithms. The results of these tests showed that the implicit algorithm gave results within 0.1% of those generated by the explicit code.

The CPU-time requirement for the implicit algorithm was found to be similar to that of the explicit algorithm even though the explicit algorithm does not require the solutions of linear simultaneous equations. This is because the CPU time needed to solve Eqs. (1) to (5) accounts for only a small part of the overall CPU-time requirement. The most expensive parts of the overall algorithm are associated with identifying the cells in which the particles are located and interpolating data from the continuum phase. Thus, the implicit method developed here would be more efficient than explicit methods for problems in which the maximum time-step size based on accuracy exceeds the maximum time-step size based on the stability of the explicit methods. An example of such a problem is liquid sprays in propulsion systems in which droplet sizes vary over a very wide range.

Acknowledgment

This work was partially supported by grant NAG 3-997 from the NASA-Lewis Research Center with Jack McFadden as the grant monitor. The authors are grateful for this support. The authors also thank Suri Raju for giving us a copy of his explicit code.

References

[1]Raju, M. S., and Sirignano, W. A., "Multi-Component Spray Computations in a Modified Centerbody Combustor," *Journal of Propulsion and Power*, Vol. 6, No. 2, 1990, pp. 97–105; also AIAA Paper 88-0638, Jan. 1988.

[2]Dukowicz, J. K., "A Particle-Fluid Numerical Model for Liquid Sprays," *Journal of Computational Physics*, Vol. 35, No. 2, 1980, pp. 229–253.

[3]Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp., Washington, DC, 1980.

[4]Issa, R. I., "Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, Vol. 62, No. 1, 1985, pp. 40–65.

[5]Shih, T. I-P., Bailey, R. T., Nguyen, H. L., and Roelke, R. J., "Algebraic Grid Generation for Complex Geometries," *International Journal for Numerical Methods in Fluids*, Vol. 13, No. 1, 1991, pp. 1–31.

[6]Leonard, B. P., "Adjusted Quadratic Upstream Algorithms for Transient Incompressible Convection," AIAA Paper 79-1469, June 1979.

Comparative Numerical Study of Two Turbulence Models for Airfoil Static and Dynamic Stall

Donald P. Rizzetta* and Miguel R. Visbal†
U.S. Air Force Wright Laboratory,
Wright-Patterson Air Force Base, Ohio 45433

Introduction

WHILE of fundamental interest because of the complex fluid mechanisms involved, both static and dynamic airfoil stall are also important in a variety of applications including high angle-of-attack aerodynamics, maneuvering aircraft, control surface motions, helicopter rotors, wind turbines, and turbomachinery. Moreover, a critical need exists for turbulence models which provide efficient and accurate simulation of such flows for practical numerical computations. The objective of the present work is to examine the adequacy of two turbulence models for the calculation of both static and dynamic airfoil stall flowfields by comparison with each other as well as with experimental data. For this purpose, the commonly used Baldwin-Lomax[1] algebraic model and the two-equation k-ϵ formulation of Launder and Sharma,[2] as generalized by Gerolymos,[3] were selected. The two-equation model requires no predefined turbulence length scales or wall functions, and because it includes low-Reynolds-number terms, both k and ϵ vanish at solid surfaces. Thus the formulation is attractive for the computation of flowfields about complex three-dimensional configurations, or for applications on unstructured meshes.

Results

Experimental conditions of the investigation by Lorber and Carta[4] were chosen to be simulated numerically. This selection was prompted by the comprehensive set of data that are available at high Reynolds number for both steady and unsteady flows. Two Mach numbers were considered, $M_\infty = 0.2$ and $M_\infty = 0.4$, with corresponding chord Reynolds numbers of 2×10^6 and 4×10^6. The wind-tunnel model consisted of a Sikorsky SSC-A09 supercritical section with a 43.9-cm chord.

The governing equations were taken to be the unsteady compressible two-dimensional Navier-Stokes equations written in mass-averaged variables and expressed in nondimensional strong-conservation form. Steady and unsteady solutions to these equations were obtained by an approximately factored Beam-Warming[5] finite difference algorithm. Complete details of the calculations including the governing equations, boundary conditions, numerical method, generation of the (303×131) mesh, and grid resolution study may be found in Ref. 6.

At $M_\infty = 0.2$, 17 cases were considered for angles of attack between 0 and 30 deg. Aerodynamic force coefficients for these cases appear in Fig. 1 where the computations employing both the k-ϵ and Baldwin-Lomax models are compared to the experimental data. Above an angle of attack of 20 deg, no steady solutions were obtainable with the Baldwin-Lomax model. Although the k-ϵ equations fail to predict the abrupt onset of stall that is evident experimentally, they do produce a more favorable comparison with drag and moment than the algebraic model.

Eleven steady solutions employing the k-ϵ equations for $0 \le \alpha \le 20$ deg at $M_\infty = 0.4$ were generated numerically. Once again, with the Baldwin-Lomax model no steady results could be obtained for $\alpha > 9$ deg. Aerodynamic force coefficients for these cases are shown in Fig. 2. The computations generally compare more favorably with the data than was true for $M_\infty = 0.2$. It is also noted that

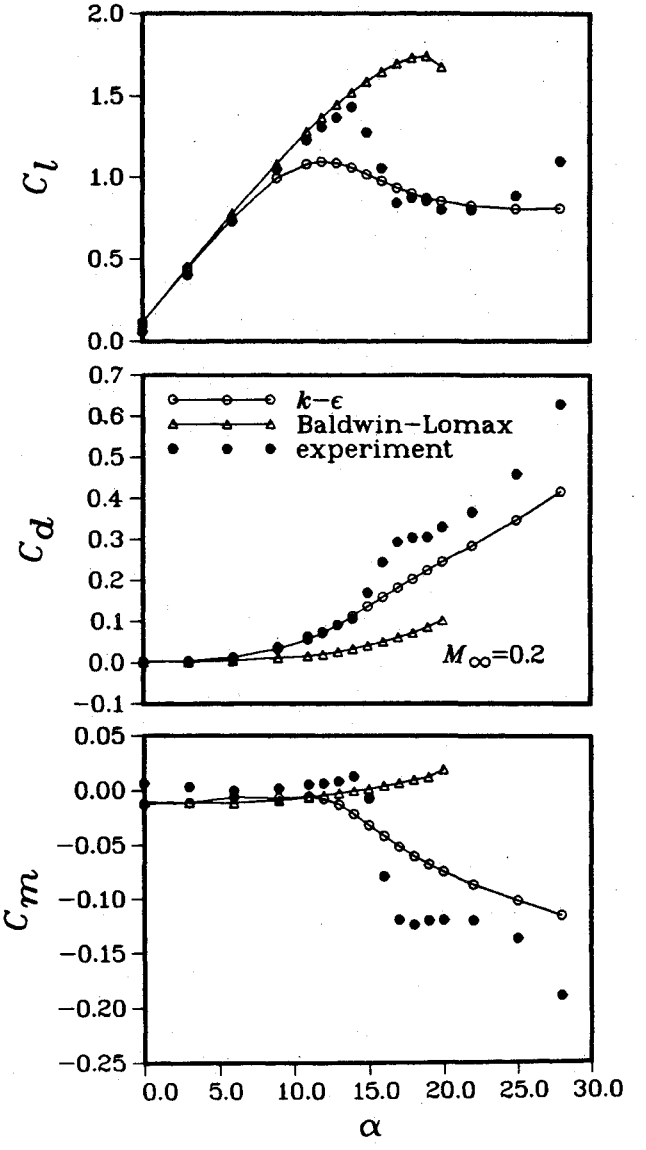

Fig. 1 Steady aerodynamic force coefficients for $M_\infty = 0.2$.

Received May 19, 1992; presented as Paper 92-4649 at the AIAA Atmospheric Flight Mechanics Conference, Hilton Head, SC, Aug. 10–12, 1992; revision received Sept. 19, 1992; accepted for publication Sept. 22, 1992.

*Aerospace Engineer, Flight Dynamics Directorate, Aeromechanics Division, Computational Fluid Dynamics Research Branch. Associate Fellow AIAA.

†Aerospace Engineer, Flight Dynamics Directorate, Aeromechanics Division, Computational Fluid Dynamics Research Branch. Member AIAA.