Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
International Journal of Multiphase Flow 30 (2004) 901–937
International Journal of Multiphase Flow
www.elsevier.com/locate/ijmulflow

A pseudocompressibility method for the numerical simulation of incompressible multifluid flows

R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous *

Center for Risk Studies and Safety, University of California, Santa Barbara, CA 93106, USA

Received 4 September 2003; received in revised form 21 March 2004

This paper is dedicated to honor George Yadigaroglu for his significant contributions in multiphase flow applications to nuclear reactor safety

Abstract

This paper presents an explicit characteristics-based, conservative, finite-difference method for the simulation of incompressible multiphase flows. The method is based on the artificial compressibility concept, extended to variable-density, and uses a time stretching procedure to relieve the acoustic constrain. We take advantage of the algorithmic simplicity and hyperbolicity provided by the artificial compressibility to develop a flow solver that is numerically robust, accurate and effective for massively parallel computations of incompressible multifluid flows. The resulting method, named Numerical Acoustic Relaxation or NAR, is a combination of the AC concept with the Level Set method for interface-capturing and the Ghost-Fluid method to compute flows with multiple, arbitrary density variation, free or stationary interfaces. In this paper we demonstrate convergence and accuracy of the solver by computing such standard test problems as the ''Lid-Driven Cavity'' and ''Doubly Periodic Shear Layer''. Competitiveness with approximate projection, vorticity stream function, pseudospectral, and Lattice Boltzmann Equation is also discussed. In addition, we demonstrate the interface-capturing features of NAR by means of the ''simple'' ''Rayleigh–Taylor'' and ''Water Column Collapse'' problems.

Keywords: Incompressible flow; Artificial compressibility; Pseudocompressibility method; Numerical acoustic relaxation; Characteristics-based treatment; Level set; Ghost fluid; Multifluid flow

* Corresponding author. Tel.: +1-805-893-3892; fax: +1-805-893-4927.
E-mail address: theo@theo.ucsb.edu (T.G. Theofanous).

0301-9322/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmultiphaseflow.2004.03.010

1. Introduction

This paper is concerned with direct numerical simulations (DNS) of incompressible multifluid flow at high density ratios. In such tasks, the two main enabling components are a Navier–Stokes equations (NSE) solver, and a method for interface treatment. Over past two decades, significant progress has been made in developing techniques for interface treatment (tracking or capturing), while Navier–Stokes equations solver can be taken from a panoply of schemes developed for single-phase flows. The methods based on Poisson equation formulations (e.g., Approximate Projection) have been the preferred choice due to their excellent performance for incompressible single-phase flow DNS. Notwithstanding the advances, it has remained a challenge to perform large-scale (highly resolved) DNS for multifluid flow at large density ratios, as needed, for example, when inertia effects are important. A major obstacle toward this goal lies in the prohibitively low efficiency of solving the Poisson equation's variable-coefficient linear algebra.

The method presented in this paper was inspired by our current efforts on Compressible Multihydrodynamics (CM) (Nourgaliev et al., in press), in combination with some recent work on reviewing and assessing of the Lattice Boltzmann Equation (LBE) method (Nourgaliev et al., 2003a). From the LBE we borrow the pseudocompressibility feature. In concept very similar to the artificial compressibility (AC) method (Chorin, 1967), iterations are applied to dissipate density/pressure "waves" until the solenoidal (incompressibility) condition is satisfied to a sufficient degree everywhere. While theoretically not as well founded as classical incompressible flow (elliptic) solvers, both (LBE and AC) methods offer great advantages in algorithmic (programming) simplicity, both are explicit and hence natural for massively parallel computations, and both are having a significant impact in practice. The LBE is known now to be especially well suited for incompressible flows in complex, fine-featured domains, such as porous media, and disperse (multiparticle, multibubble) two-phase flow systems; see e.g. Ladd and Verberg (2001); Chen and Doolen (1998) as well as review in Nourgaliev et al. (2003a). The main current limitations on the latter class are that the fluid-to-fluid density ratio has to remain below $\sim$10, and that the viscosities are the same. The AC method, while considerably older, received rather modest attention, and only for single-phase flows, yet it continues to remain relevant; see e.g. Rogers and Kwak (1990), Rogers and Kwak (1991) and references therein; also Kwak and Kiris (2003).

In this work our point of departure is the AC method of Chorin (1967). The choice allows us, like the LBE, to bypass the Poisson equation, the source of great advantage for both methods. Unlike the LBE, it is not subject to the density ratio, and viscosity limitation, mentioned above. The price is that now we have to "capture" interfaces, and to do so at the appropriate level of fidelity, we must maintain the interface sharp and match the proper boundary conditions from each side (and at each point along the interfaces). We use the Level Set (LS) (Osher and Sethian, 1988) and Ghost Fluid (GF) (Fedkiw et al., 1999) methodologies for this purpose. The formulation leads to a hyperbolic system, which allows for high-order treatments, by Godunov characteristics-based schemes (Nourgaliev et al., 2003b). It turns out the computational machinery developed for fully-compressible multiphase flows (Characteristics-Based Matching "CBM", Nourgaliev et al., in press) is of value to the method developed in the present study as well. Thus the NAR method can be put into an overall perspective, as illustrated in Fig. 1.

The purpose of this paper is to present the formulation (Section 2) of NAR, to describe key algorithmic details (Sections 3 and 4), to demonstrate convergence and accuracy characteristics of

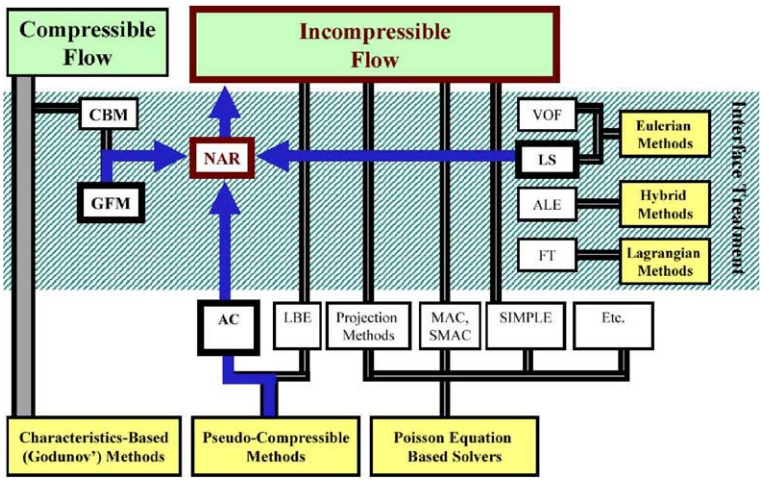

Fig. 1. The numerical scheme presented in this paper (NAR, AC, LS, GFM and CBM) and relations to other methods of incompressible, compressible and multiphase flow. LS: Level Set (Osher and Sethian, 1988); VOF: volume of fluid, (Hirt and Nichols, 1981); FT: front tracking (Unverdi and Tryggvason, 1992); ALE (Arbitrary Lagrangian Eulerian, Hirt et al., 1974).

the method, and to illustrate the interface-capturing features of it, by means of simple test problems. For the former we use the "Lid-Driven Cavity", the "Doubly Periodic Shear Layer", and "Decaying Travelling Wave" problems (Sections 5.1, 5.2 and 6). For the latter we consider "Rayleigh–Taylor Instability" and a "Water Column Collapse" (Sections 5.3 and 5.4).

2. The 'Numerical Acoustic Relaxation (NAR)' method

Artificial compressibility (AC). The numerical algorithms based on the artificial compressibility (AC) concept originated from the classical paper by Chorin (1967). The main idea of Chorin's concept is to add a time derivative of pressure into the continuity equation, thus providing a direct coupling between pressure and flow velocity and hence avoiding the need to solve the elliptic Poisson equation. It is noted that Chorin's (1967) paper was devoted to a steady-state incompressible flow calculation. In fact, the idea of using time stabilization to solve a steady flow problem has its origin in von Newmann and Richtmyer's work (1950) on transonic flow, where they first used the physical time to formulate a mathematically homogeneous, hyperbolic problem in both supersonic and subsonic regions. After Chorin's work, the artificial compressibility concept was adopted for transient flow calculations: on each physical time step, sub-iterations are used to drive the velocity divergence to zero (see Rogers and Kwak, 1990; Rogers and Kwak, 1991 and references therein). The resulting procedure resembles time pre-conditioning strategy developed for compressible flows (e.g., Turkel, 1987; van Leer et al., 1991). Recently, Kwak and Kiris (2003) reported broad and successful applications of the AC method for computing incompressible single-phase flow problems of interest to NASA (Kwak and Kiris, 2003). The AC method had not been previously extended to variable-density flow and multifluid flow problems. Such an extension is accomplished in the present NAR method (see Section 2.2).

2.1. Formulation

Let us consider isothermal Newtonian fluid dynamics described by a system of four partial differential equations with five unknown (ρ, u, v, w, P) (Landau and Lifschitz, 1988):

$$\begin{aligned} &\partial_t \rho + \partial_j \rho u_j = 0 \\ &\partial_t \rho u_i + \partial_j \rho u_i u_j = -\partial_i P + \partial_j \mathscr{T}_{ij} + \rho g_i \end{aligned} \tag{1}$$

where a viscous stress tensor has the following form (Aris, 1962):

$$\mathscr{T}_{ij} = \mu(\partial_j u_i + \partial_i u_j) + \underbrace{\left(\xi - \frac{2}{3}\mu\right)}_{\text{'bulk' viscosity, } \lambda} \partial_k u_k \cdot \delta_{i,j} \tag{2}$$

μ and ξ are the 'first' and the 'second' fluid viscosities; g_i is a body force (e.g., gravity). In a general case of compressible flow, one needs to provide an equation of state linking P and ρ to close the system. For incompressible flow, the fluid density is a constant, the equation of state is trivial, and the momentum equations must be solved subject to the solenoidal constraint for the velocity field. From a numerical standpoint, this "loss" of a pressure–velocity coupling creates difficulties in that the whole field must be simultaneously coupled (i.e., by a Poisson equation) at a very high tolerance level.

Artificial fluid. Let us introduce an *artificial* compressible fluid, A-fluid, which has the following equation of state:

$$P = \rho c_s^2 \tag{3}$$

where c_s is a pseudo-sound-speed in the A-fluid. The A-fluid density can be given by

$$\rho = \rho_0 + \delta_\rho \tag{4}$$

where ρ_0 is the given reference density; and δ_ρ is a small variation of density:

$$\frac{\delta_\rho}{\rho_0} \ll 1 \tag{5}$$

Substituting Eqs. (3) and (4) into Eq. (1), and assuming that $\xi = \frac{2}{3}\mu$ (Landau and Lifschitz, 1988), we obtain:

$$\begin{aligned} &\partial_t P + \partial_j (c_s^2 \rho_0 u_j) = \underbrace{-\partial_j c_s^2 \delta_\rho u_j}_{\Gamma_\rho} \\ &\partial_t \rho_0 u_i + \partial_j (\rho_0 u_i u_j + P\delta_{i,j}) = \partial_j [\mu(\partial_i u_j + \partial_j u_i)] + \rho_0 g_i + \underbrace{\partial_t \delta_\rho u_i + \partial_j \delta_\rho u_i u_j + \delta_\rho g_i}_{\Gamma_{u_i}} \end{aligned} \tag{6}$$

Given condition Eq. (5), terms associated with density variation in Eq. (6), namely Γ_ρ and Γ_{u_i} are negligibly small, i.e. $\Gamma_\rho \approx 0$ and $\Gamma_{u_i} \approx 0$. For a steady-state problem of incompressible flow such as those considered in the original work of Chorin (1967), the transient pressure term gradually

decreases and eventually disappears at the established steady-state. In other words, numerical acoustics has enough time to propagate and dissipate over the whole computational domain for all the waves to die out, rendering the introduced acoustics irrelevant for the steady-state solution.

Pseudotime formalism. In the present work, we are interested in transient incompressible flows. Therefore, momentum transient terms of Eq. (6) must be retained, whilst the transient pressure term must be effectively suppressed. This is achieved by pressure-relaxation procedure, applied on each and every calculation time step, δt. For time integration of the pressure equation, we introduce a *"stretched pseudotime"* τ defined as

$$\begin{aligned} &\mathrm{d}\tau \equiv \mathscr{N}\,\mathrm{d}t \\ &\tau(t) = \int_0^t \mathscr{N}\,\mathrm{d}t \end{aligned} \tag{7}$$

with $\mathscr{N}$ being a "time stretching" parameter. The acoustic waves are assumed to "travel" and dissipate in the "stretched-time–space" τ. Neglecting minor contributions by the density variation terms Γ_ρ, the continuity equation equivalent of Eq. (6) in "τ-*time–space*" can now be rewritten as

$$\partial_\tau P + \partial_j\left(c_s^2 \rho_0 u_j\right) = 0 \tag{8}$$

so that the effect of "relaxation(/stretching)" translates into the introduction of time stretching parameter $\mathscr{N}$ into the divergence term of the first equation of system (6). Thus, the resulting equivalent system of governing equations becomes

$$\partial_t P + \partial_j\left(\mathscr{N} c_s^2 \rho_0 u_j\right) = 0 \tag{9}$$

$$\partial_t \rho_0 u_i + \partial_j(\rho_0 u_i u_j + P\delta_{i,j}) = \partial_j[\mu(\partial_i u_j + \partial_j u_i)] + \rho_0 g_i \tag{10}$$

For $\mathscr{N} = 1$, Eq. (9) is similar to Chorin's formulation for a steady-state problem (Chorin, 1967).

Scaling analysis. To non-dimensionalize Eqs. (9) and (10), we introduce characteristic length scale L and velocity scale U_0. Using the following notation for components of the position and velocity vectors

$$\begin{aligned} &\mathbf{r} = (x_1, x_2, x_3) = (x, z, y) \\ &\mathbf{u} = (u_1, u_2, u_3) = (u, v, w) \end{aligned}$$

non-dimensional variables and parameters are defined as

$$\begin{array}{ll} \text{Time:} & \hat{t} = t\frac{\mathsf{U}_0}{\mathsf{L}} \\ \text{Length:} & \hat{\mathbf{r}} = \frac{\mathbf{r}}{\mathsf{L}} \\ \text{Velocity:} & \hat{\mathbf{u}} = \frac{\mathbf{u}}{\mathsf{U}_0} \\ \text{Kinematic viscosity/Reynolds number:} & \hat{\nu} = \frac{1}{\mathsf{Re}} = \frac{\nu}{\mathsf{U}_0\mathsf{L}} \\ \text{Body force/Froude number:} & \hat{g} = \frac{1}{\mathsf{Fr}} = \frac{g\mathsf{L}}{\mathsf{U}_0^2} \\ \text{Numerical Mach number:} & \mathsf{N}^{(\mathsf{M})} = \frac{\mathsf{U}_0}{\mathsf{c}_s} \\ \text{Effective numerical Mach number:} & \mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}} = \frac{\mathsf{U}_0}{\mathsf{c}_s\sqrt{\mathscr{N}}} = \frac{\mathsf{N}^{(\mathsf{M})}}{\sqrt{\mathscr{N}}} \end{array} \tag{11}$$

The above-introduced "numerical Mach number" is an artificial construction—a numerical discretization parameter (see Section 3.4) and should not be confused with the physical Mach number of compressible fluid dynamics. Given the above definitions and $\rho_0 = \text{const}$, the governing equations of A-fluid (9) can be cast into the following non-dimensional form:

$$\begin{aligned} &\partial_{\hat{j}}\hat{u}_j = \underbrace{-\frac{[\mathsf{N}^{(\mathsf{M})}]^2}{\mathscr{N}}\partial_{\hat{t}}\hat{P}}_{\mathscr{D}_{\mathsf{A}}} \\ &\partial_{\hat{t}}\hat{u}_i + \partial_{\hat{j}}(\hat{u}_i\hat{u}_j) = -\frac{1}{\rho_0}\partial_{\hat{i}}\hat{P} + \frac{1}{\mathsf{Re}}\partial_{\hat{j}}[(\partial_{\hat{i}}\hat{u}_j + \partial_{\hat{j}}\hat{u}_i)] + \frac{\hat{a}_i}{\mathsf{Fr}} \end{aligned} \tag{12}$$

and $\hat{a}_i$ is a unit vector, specifying the direction of the body force. Solution of the original incompressible flow problem emerges from Eq. (12) as the divergence source term $\mathscr{D}_{\mathsf{A}}$ approaches zero. Theoretically, at the steady-state of $\partial_{\hat{t}}\hat{P} = 0$, a solenoidal velocity field is completely recovered $\partial_{\hat{j}}\hat{u}_j = 0$. Practically, we have two parameters to control the "divergence" term, $\mathscr{D}_{\mathsf{A}}$, first by decreasing the numerical Mach number $\mathsf{N}^{(\mathsf{M})}$ (i.e., to increase c_{s}); and second by increasing the pseudotime "stretching", $\mathscr{N} \uparrow$.

2.2. Variable-density extension

The formulation of the NAR concept is extendable to multiphase incompressible flow that consists of m incompressible fluids topologically separated by fluid–fluid interfaces. Each mth fluid has its own, but constant, density, $\rho(\mathbf{r}) = \rho_m = \text{const}$. Jump in fluid density and other properties occurs across an infinitely thin interfacial layer. Using the notion:

$$\begin{aligned} &\text{"Scaled density"}: && \tilde{\rho} = \tfrac{\rho(\mathbf{r})}{\rho_0} \\ &\text{"Scaled phasic velocity"}: && \tilde{\mathbf{u}} = \tfrac{\rho(\mathbf{r})\mathbf{u}}{\rho_0} = \tilde{\rho}\mathbf{u} \end{aligned} \tag{13}$$

(ρ_0 is a "reference" density), Eq. (9) can be rewritten in a "variable-density" formulation as follows:

$$\begin{aligned} &\partial_t P + \partial_j(\mathscr{N} c_{\mathrm{s}}^2 \rho_0 \tilde{u}_j) = 0 \\ &\partial_t \tilde{u}_i + \partial_j\left(\tilde{u}_i u_j + \frac{P}{\rho_0}\delta_{i,j}\right) = \partial_j\left[\frac{\mu}{\rho_0}(\partial_i u_j + \partial_j u_i)\right] + \tilde{\rho} g_i \end{aligned} \tag{14}$$

with four unknown variables being $(P, \tilde{u}, \tilde{v}, \tilde{w})$. In order to obtain solution for the system Eq. (14) of multiphase A-fluid flow, local density $\rho(\mathbf{r})$ (and viscosity) must be determined. This requires knowledge of the interface position, that can be computed by using an interface-capturing method. Implementation of such a method for NAR is detailed in Section 4.

3. Numerical implementation of NAR

Governing equations (14) of the A-fluid can be written in the following form

$$\mathbf{U}_t + \underbrace{[\mathbf{F}]_x + [\mathbf{G}]_z + [\mathbf{H}]_y}_{\vec{\mathscr{F}}} = \mathbf{S} \tag{15}$$

with **U**, **F**, **G**, **H** and **S** being vectors of conservative variables, fluxes and source terms, respectively:

$$\mathbf{U}=\begin{bmatrix} P \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix};\quad \mathbf{F}=\begin{bmatrix} \tilde{c}_s^2\rho_0\tilde{u} \\ \tilde{u}u+\frac{P}{\rho_0} \\ \tilde{v}u \\ \tilde{w}u \end{bmatrix};\quad \mathbf{G}=\begin{bmatrix} \tilde{c}_s^2\rho_0\tilde{v} \\ \tilde{u}v \\ \tilde{v}v+\frac{P}{\rho_0} \\ \tilde{w}v \end{bmatrix};$$
$$\mathbf{H}=\begin{bmatrix} \tilde{c}_s^2\rho_0\tilde{w} \\ \tilde{u}w \\ \tilde{v}w \\ \tilde{w}w+\frac{P}{\rho_0} \end{bmatrix};\quad \mathbf{S}=\begin{bmatrix} 0 \\ \partial_x\mathscr{T}_{xx}+\partial_z\mathscr{T}_{xz}+\partial_y\mathscr{T}_{xy}+g_{(x)} \\ \partial_x\mathscr{T}_{zx}+\partial_z\mathscr{T}_{zz}+\partial_y\mathscr{T}_{zy}+g_{(z)} \\ \partial_x\mathscr{T}_{yx}+\partial_z\mathscr{T}_{yz}+\partial_y\mathscr{T}_{yy}+g_{(y)} \end{bmatrix} \tag{16}$$

where $\tilde{c}_s=c_s\sqrt{\mathscr{N}}$, and $\mathscr{T}_{ij}=\frac{\mu}{\rho_0}(\partial_i u_j+\partial_j u_i)$.

Understanding of mathematical behavior and advanced numerical schemes for such systems have been developed in the past (Lax, 1973; Le Veque, 1992; Dafermos, 2000). This knowledge is used in this work to develop a high-order-accurate characteristics-based finite-difference scheme for solving the NAR A-fluid equations, and constitutes a major advancement as compared to previous realizations of the artificial compressibility method.

3.1. Characteristic analysis

The Jacobian matrices for fluxes **F**, **G** and **H** in Eq. (15) are:

$$\mathbf{J}_{(\mathbf{F})}=\begin{bmatrix} 0 & \rho_0\tilde{c}_s^2 & 0 & 0 \\ \frac{1}{\rho_0} & 2u & 0 & 0 \\ 0 & v & u & 0 \\ 0 & w & 0 & u \end{bmatrix};\quad \mathbf{J}_{(\mathbf{G})}=\begin{bmatrix} 0 & 0 & \rho_0\tilde{c}_s^2 & 0 \\ 0 & v & u & 0 \\ \frac{1}{\rho_0} & 0 & 2v & 0 \\ 0 & 0 & w & v \end{bmatrix};\quad \mathbf{J}_{(\mathbf{H})}=\begin{bmatrix} 0 & 0 & 0 & \rho_0\tilde{c}_s^2 \\ 0 & w & 0 & u \\ 0 & 0 & w & v \\ \frac{1}{\rho_0} & 0 & 0 & 2w \end{bmatrix} \tag{17}$$

with the following eigenvalues:

$$\Lambda_{(\mathbf{F})}=\begin{bmatrix} u+\sqrt{\tilde{c}_s+u^2} & 0 & 0 & 0 \\ 0 & u-\sqrt{\tilde{c}_s+u^2} & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{bmatrix}$$

$$\Lambda_{(\mathbf{G})} = \begin{bmatrix} v+\sqrt{\tilde{c}_s+v^2} & 0 & 0 & 0 \\ 0 & v-\sqrt{\tilde{c}_s^2+v^2} & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & v \end{bmatrix}$$

$$\Lambda_{(\mathbf{H})} = \begin{bmatrix} w+\sqrt{\tilde{c}_s^2+w^2} & 0 & 0 & 0 \\ 0 & w-\sqrt{\tilde{c}_s^2+u^2} & 0 & 0 \\ 0 & 0 & w & 0 \\ 0 & 0 & 0 & w \end{bmatrix} \tag{18}$$

and corresponding "left-" and "right-" eigenvectors:

$$\mathbf{R}_{(\mathbf{F})} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ \frac{u+\sqrt{\tilde{c}_s^2+u^2}}{\rho_0\tilde{c}_s^2} & \frac{u-\sqrt{\tilde{c}_s^2+u^2}}{\rho_0\tilde{c}_s^2} & 0 & 0 \\ \frac{v\left[u\left(u+\sqrt{\tilde{c}_s^2+u^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+u^2)} & \frac{v\left[u\left(u-\sqrt{\tilde{c}_s^2+u^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+u^2)} & 1 & 0 \\ \frac{w\left[u\left(u+\sqrt{\tilde{c}_s^2+u^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+u^2)} & \frac{w\left[u\left(u-\sqrt{\tilde{c}_s^2+u^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+u^2)} & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}_{(\mathbf{F})} = \begin{bmatrix} \frac{\sqrt{u^2+\tilde{c}_s^2}-u}{2\sqrt{\tilde{c}_s^2+u^2}} & \frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+u^2}} & 0 & 0 \\ \frac{\sqrt{u^2+\tilde{c}_s^2}+u}{2\sqrt{\tilde{c}_s^2+u^2}} & -\frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+u^2}} & 0 & 0 \\ -\frac{v}{\rho_0(\tilde{c}_s^2+u^2)} & -\frac{uv}{\tilde{c}_s^2+u^2} & 1 & 0 \\ -\frac{w}{\rho_0(\tilde{c}_s^2+u^2)} & -\frac{uw}{\tilde{c}_s^2+u^2} & 0 & 1 \end{bmatrix} \tag{19}$$

$$\mathbf{R}_{(\mathbf{G})} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ \frac{u\left[v\left(v+\sqrt{\tilde{c}_s^2+v^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+v^2)} & \frac{u\left[v\left(v-\sqrt{\tilde{c}_s^2+v^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+v^2)} & 1 & 0 \\ \frac{v+\sqrt{\tilde{c}_s^2+v^2}}{\rho_0\tilde{c}_s^2} & \frac{v-\sqrt{\tilde{c}_s^2+v^2}}{\rho_0\tilde{c}_s^2} & 0 & 0 \\ \frac{w\left[v\left(v+\sqrt{\tilde{c}_s^2+v^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+v^2)} & \frac{w\left[v\left(v-\sqrt{\tilde{c}_s^2+v^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+v^2)} & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}_{(\mathbf{G})} = \begin{bmatrix} \frac{\sqrt{v^2+\tilde{c}_s^2}-v}{2\sqrt{\tilde{c}_s^2+v^2}} & 0 & \frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+v^2}} & 0 \\ \frac{\sqrt{v^2+\tilde{c}_s^2}+v}{2\sqrt{\tilde{c}_s^2+v^2}} & 0 & -\frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+v^2}} & 0 \\ -\frac{u}{\rho_0(\tilde{c}_s^2+v^2)} & 1 & -\frac{uv}{\tilde{c}_s^2+v^2} & 0 \\ -\frac{w}{\rho_0(\tilde{c}_s^2+v^2)} & 0 & -\frac{vw}{\tilde{c}_s^2+v^2} & 1 \end{bmatrix} \tag{20}$$

and

$$\mathbf{R}_{(\mathbf{H})} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ \frac{u\left[w\left(w+\sqrt{\tilde{c}_s^2+w^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+w^2)} & \frac{u\left[w\left(w-\sqrt{\tilde{c}_s^2+w^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+w^2)} & 1 & 0 \\ \frac{v\left[w\left(w+\sqrt{\tilde{c}_s^2+w^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+w^2)} & \frac{v\left[w\left(w-\sqrt{\tilde{c}_s^2+w^2}\right)+\tilde{c}_s^2\right]}{\rho_0\tilde{c}_s^2(\tilde{c}_s^2+w^2)} & 0 & 1 \\ \frac{w+\sqrt{\tilde{c}_s^2+w^2}}{\rho_0\tilde{c}_s^2} & \frac{w-\sqrt{\tilde{c}_s^2+w^2}}{\rho_0\tilde{c}_s^2} & 0 & 0 \end{bmatrix}$$

$$\mathbf{L}_{(\mathbf{H})} = \begin{bmatrix} \frac{\sqrt{w^2+\tilde{c}_s^2}-w}{2\sqrt{\tilde{c}_s^2+w^2}} & 0 & 0 & \frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+w^2}} \\ \frac{\sqrt{w^2+\tilde{c}_s^2}+w}{2\sqrt{\tilde{c}_s^2+w^2}} & 0 & 0 & -\frac{\rho_0\tilde{c}_s^2}{2\sqrt{\tilde{c}_s^2+w^2}} \\ -\frac{u}{\rho(\tilde{c}_s^2+w^2)} & 1 & 0 & -\frac{uw}{\tilde{c}_s^2+w^2} \\ -\frac{v}{\rho_0(\tilde{c}_s^2+w^2)} & 0 & 1 & -\frac{vw}{\tilde{c}_s^2+w^2} \end{bmatrix} \tag{21}$$

These explicit formulae are necessary for the characteristic-based treatment by Algorithm A. The discretized fluxes and diffusion terms are described in Sections 3.2 and 3.3 below.

Algorithm A. Loop over all j and k, from $\mathsf{j}_{\mathsf{beg}}$ to $\mathsf{j}_{\mathsf{end}}$ and from $\mathsf{k}_{\mathsf{beg}}$ to $\mathsf{k}_{\mathsf{end}}$:

1. Calculate $\mathbf{U}_{(i,j,k)}$ and $\mathbf{F}_{(i,j,k)}$ (see Eq. (16)) for all i (from $\mathsf{i}_{\mathsf{beg}}$ to $\mathsf{i}_{\mathsf{end}}$).
2. ...Start loop for i from $\mathsf{i}_{\mathsf{beg}}$ to $\mathsf{i}_{\mathsf{end}}$ ...
   (i) Calculate $\left[u^L_{(i+\frac{1}{2},j,k)}; u^R_{(i+\frac{1}{2},j,k)}\right]$; $\left[v^L_{(i+\frac{1}{2},j,k)}; v^R_{(i+\frac{1}{2},j,k)}\right]$; and $\left[w^L_{(i+\frac{1}{2},j,k)}; w^R_{(i+\frac{1}{2},j,k)}\right]$, using the WENO_5 algorithm.
   (ii) Using Eqs. (18)–(21), calculate $\Lambda_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}}$; $\mathbf{L}_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}}$; and $\mathbf{R}_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}}$, utilizing the "arithmetic mean" formulation $\phi_{(i+\frac{1}{2},j,k)} = \frac{\phi^L_{(i+\frac{1}{2},j,k)} + \phi^R_{(i+\frac{1}{2},j,k)}}{2}$, where $\phi = u, v$ and w.
   (iii) Calculate local 'numerical viscosity' $\varpi_{i,j,k}$ as $\max\left(|\Lambda_{(\mathbf{F})_{(i,j,k)_{s,r}}}| \cdot \delta_{s,r}; |\Lambda_{(\mathbf{F})_{(i+1,j,k)_{s,r}}}| \cdot \delta_{s,r}\right)$, where $\delta_{s,r}$ is a Kroenecker symbol.
   (iv) In the neighborhood of point (i,j,k); put the vectors of variables and fluxes into the "characteristic field" as $\mathbf{U}^{(\mathrm{ch})}_{(i+q,j,k)} = \mathbf{L}_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}} \mathbf{U}_{(i+q,j,k)}\Big|_{q=-2,\ldots,3}$, $\mathbf{F}^{(\mathrm{ch})}_{(i+q,j,k)} = \mathbf{L}_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}} \mathbf{F}_{(i+q,j,k)}|_{q=-2,\ldots,3}$.
   (v) Calculate local "characteristic" fluxes using the "Local Lax–Friedrichs" (LLF) flux splitting formula:

$\mathbf{F}^{(\mathrm{ch}),\pm}_{(i+q,j,k)} = \mathbf{F}^{(\mathrm{ch})}_{(i+q,j,k)} \pm \frac{\varpi_{(i,j,k)}}{2}\mathbf{U}^{(\mathrm{ch})}_{(i+q,j,k)}\Big|_{q=-2,\ldots,3}$.

(vi) Calculate "edge" fluxes in the "characteristic field", using WENO_5 algorithm:

$$\mathbf{F}^{(\mathrm{ch}),+}_{(i+q,j,k)}\Big|_{q=-2,\ldots,3} \overset{\mathsf{WENO}_5}{\rightarrow} \vec{\mathsf{F}}^{(\mathrm{ch}),L}_{(i+\frac{1}{2},j,k)} \quad \text{and} \quad \mathbf{F}^{(\mathrm{ch})-}_{(i+q,j,k)}\Big|_{q=-2,\ldots,3} \overset{\mathsf{WENO}_5}{\rightarrow} \vec{\mathsf{F}}^{(\mathrm{ch}),R}_{(i+\frac{1}{2},j,k)}$$

(vii) "Retrieve" the "edge" fluxes from the "characteristic field" using the right eigenvector matrix: $\vec{\mathsf{F}}_{(i+\frac{1}{2},j,k)} = \left(\vec{\mathsf{F}}^{(\mathrm{ch}),L}_{(i+\frac{1}{2},j,k)} + \vec{\mathsf{F}}^{(\mathrm{ch}),R}_{(i+\frac{1}{2},j,k)}\right)\mathbf{R}_{(\mathbf{F})_{(i+\frac{1}{2},j,k)}}$.

3. ... Finish loop i ...

3.2. Discretization of fluxes

The discretization of the flux terms in Eq. (15) at a computational node (i,j,k) is based on "*Characteristics-Based Conservative Finite-Difference*" approach (Fedkiw et al., 1998):

$$\vec{\mathscr{F}}^{(\diamondsuit)}_{(i,j,k)} = \frac{\vec{\mathsf{F}}^{(\diamondsuit)}\left(x_{(i+\frac{1}{2},j,k)}\right) - \vec{\mathsf{F}}^{(\diamondsuit)}\left(x_{(i-\frac{1}{2},j,k)}\right)}{x_{(i+\frac{1}{2},j,k)} - x_{(i-\frac{1}{2},j,k)}} + \frac{\vec{\mathsf{G}}^{(\diamondsuit)}\left(z_{(i,j+\frac{1}{2},k)}\right) - \vec{\mathsf{G}}^{(\diamondsuit)}\left(z_{(i,j-\frac{1}{2},j,k)}\right)}{z_{(i,j+\frac{1}{2},k)} - z_{(i,j-\frac{1}{2},k)}} + \frac{\vec{\mathsf{H}}^{(\diamondsuit)}\left(y_{(i,j,k+\frac{1}{2})}\right) - \vec{\mathsf{H}}^{(\diamondsuit)}\left(y_{(i,j,k-\frac{1}{2})}\right)}{y_{(i,j,k+\frac{1}{2})} - y_{(i,j,k-\frac{1}{2})}} \tag{22}$$

where $(\diamondsuit)$ denotes the point of time discretization (n or m, see "Time discretization" Section 3.4 below); while $\vec{\mathsf{F}}_{(i\pm\frac{1}{2},j,k)}$, $\vec{\mathsf{G}}_{(i,j\pm\frac{1}{2},k)}$ and $\vec{\mathsf{H}}_{(i,j,k\pm\frac{1}{2})}$ are *numerical fluxes* at the edges of the computational cell, evaluated by high-order characteristic upwind differencing. Conservative finite-difference algorithm used to compute numerical fluxes $\vec{\mathsf{F}}$ at the cell edges $(i\pm\frac{1}{2},j,k)$ i.e. $\vec{\mathsf{F}}_{(i\pm\frac{1}{2},j,k)}$, is described in Algorithm A (see also (Jiang and Shu, 1996) for detail description). Numerical fluxes in the other directions, $\vec{\mathsf{G}}_{(i,j\pm\frac{1}{2},k)}$ and $\vec{\mathsf{H}}_{(i,j,k\pm\frac{1}{2})}$ can be obtained similarly.

3.3. Discretization of viscous diffusion term

All spatial gradients, present in the viscous diffusion term, i.e. $\partial_i u_j$ and $\partial_j \mathscr{T}_{ij}$, are calculated using the fourth-order-accurate central difference scheme. For example, derivatives in x-direction are:

$$\partial_x \psi|^{(\diamondsuit)}_{i,j,k} = \frac{-\psi^{(\diamondsuit)}_{i+2,j,k} + 8\psi^{(\diamondsuit)}_{i+1,j,k} - 8\psi^{(\diamondsuit)}_{i-1,j,k} + \psi^{(\diamondsuit)}_{i-2,j,k}}{12\delta_x} + \mathrm{O}\left(\delta_x^4\right) \tag{23}$$

where $\psi = u$, v, w, $\mathscr{T}_{xx}$, $\mathscr{T}_{xz}$, $\mathscr{T}_{xy}$, $\mathscr{T}_{zx}$, $\mathscr{T}_{zz}$, $\mathscr{T}_{zy}$, $\mathscr{T}_{yx}$, or $\mathscr{T}_{yz}$, $\mathscr{T}_{yy}$, and $(\diamondsuit)$ denotes the points of time discretization, $(\diamondsuit) = n$ or m. Similarly, one can derive expressions for $\partial_z \psi|^{(\diamondsuit)}_{i,j,k}$ and $\partial_y \psi|^{(\diamondsuit)}_{i,j,k}$.

3.4. Time discretization and NAR (Algorithm B)

For time discretization of the A-fluid momentum conservation equations (10), we implement an *Implicit Trapezoidal* (*IT*) scheme as follows:

$$\frac{\partial \psi}{\partial t} = \mathscr{S}_{\psi} \overset{\text{"IT" discretization}}{\longrightarrow} \frac{\begin{array}{l}\text{Iterate for } m = n, n+1, \ldots, n+\mathscr{N}: \\ \psi^{(m+1)}_{(i,j,k)} = \psi^{(n)}_{(i,j,k)} + \frac{\delta_t}{2}\left(\mathscr{S}^{(n)}_{\psi_{i,j,k}} + \mathscr{S}^{(m)}_{\psi_{(i,j,k)}}\right)\end{array}}{\text{In the end, set } \psi^{(n+1)}_{(i,j,k)} = \psi^{n+\mathscr{N}}_{(i,j,k)}} \tag{24}$$

where $\psi = \tilde{u}, \tilde{v}, \tilde{w}$; and $\mathscr{S}_{\psi}$ is a source term for corresponding transport equation, including the fluxes, viscous diffusion and body force terms. The advantage of the IT scheme is that it provides the second-order accuracy for time discretization, and is known for its numerical stability property, particularly important in solving stiff systems (Oran and Boris, 1987).

Pressure equation (9) is discretized using an "implicit" iterative procedure:

$$\frac{\partial P}{\partial t} = -\mathscr{F}_P \overset{\text{Discretization}}{\longrightarrow} \frac{\begin{array}{l}\text{Iterate for } m = n, n+1, \ldots, n+\mathscr{N}: \\ P^{(m+1)}_{(i,j,k)} = P^{(m)}_{(i,j,k)} - \delta_t \mathscr{F}^{(m)}_{P(i,j,k)}\end{array}}{\text{In the end, set } P^{(n+1)}_{(i,j,k)} = P^{n+\mathscr{N}}_{(i,j,k)}} \tag{25}$$

where $\mathscr{F}_P$ is a flux term of the pressure evolution equation (i.e., $\mathscr{F}_1$ in the vector defined by Eq. (22)).

The calculation procedure for time advancement from $t^{(n)}$ to $t^{(n+1)}$ is given by Algorithm B. Note: Step 4(iii) in Algorithm B is concerned with the treatment of multifluid interfaces and discussed separately in Section 4.

Since the IT scheme is utilized for time discretization, to calculate the numerical flux terms $\vec{\mathscr{F}}^{(\diamondsuit)}_{(i,j,k)}$, one has to employ the sound speed c_s instead of $\tilde{c}_s$ in Eqs. (16)–(21). This implies that the numerical acoustic waves propagate with speed c_s in the "stretched-time–space".

Algorithm B.

1. Calculate the stretching parameter $\mathscr{N}$ using Eq. (27).
2. Calculate source terms (including all fluxes, viscous diffusion and body force terms), based on the currently available values at the time point $t^{(n)}$: $\mathscr{S}^{(n)}_{\psi_{(i,j,k)}}$; $\psi = \tilde{u}, \tilde{v}, \tilde{w}, P$; at all computational nodes (i,j,k).
3. Initialize "IT" procedure by setting $m = n$ and $\psi^{(m)}_{(i,j,k)} = \psi^{(n)}_{(i,j,k)}$; $\psi = \tilde{u}, \tilde{v}, \tilde{w}, P$; at all computational nodes (i,j,k).
4. Iterate for $m = n, n+1, \ldots, n+\mathscr{N}$:
   (i) Calculate source terms (including all fluxes, viscous diffusion and body force terms), based on the currently available values at the "stretched-time point" $t^{(m)}$: $\mathscr{S}^{(m)}_{\psi_{(i,j,k)}}$; $\psi = \tilde{u}, \tilde{v}, \tilde{w}, P$; at all computational nodes (i,j,k).
   (ii) Advance "conservation variables" in the "stretched time" as:

$$\begin{cases} P^{(m+1)}_{(i,j,k)} = P^{(m)}_{(i,j,k)} - \delta_t \cdot \mathscr{F}^{(m)}_{P_{(i,j,k)}} \\ \psi^{(m+1)}_{(i,j,k)} = \psi^{(n)}_{(i,j,k)} + \frac{\delta_t}{2} \cdot \left(\mathscr{S}^{(n)}_{\psi_{(i,j,k)}} + \mathscr{S}^{(m)}_{\psi_{(i,j,k)}}\right) \end{cases}$$

   where $\psi = \tilde{u}$, $\tilde{v}$ and $\tilde{w}$, at all computational nodes (i,j,k).
   (iii)
      (a) If $m = n + \mathscr{N}$, advance in time the Level Set function: $\varphi^{(n)}_{(i,j,k)} \xrightarrow{\delta_t} \varphi^{(n+1)}_{(i,j,k)}$.
      (b) Set boundary conditions on both domain boundaries and fluid–fluid interfaces (using the "GFM" approach).
5. Finish "IT" procedure by setting $\psi^{(n+1)}_{(i,j,k)} = \psi^{(n+\mathscr{N})}_{(i,j,k)}$; $\psi = \tilde{u}, \tilde{v}, \tilde{w}, P$; at all computational nodes (i,j,k).

Time step and numerical sound speed. The "numerical sound speed" is chosen as

$$c_{\mathrm{s}} = \mathsf{CFL}_{\mathsf{NA}} \frac{\tilde{\delta}_x}{\delta_t} \tag{26}$$

where $\tilde{\delta}_x$ is the smallest grid size in the computational domain; δ_t is the calculation time step; and $\mathsf{CFL}_{\mathsf{NA}}$ is a "Courant–Friedrichs–Levy" number for "numerical acoustic wave" propagation. [1]As a rule, numerical stability is ensured with $\mathsf{CFL}_{\mathsf{NA}} < 1$. However, the value of $\mathsf{CFL}_{\mathsf{NA}}$ used in calculation depends on concrete schemes for numerical discretization of the hyperbolic terms. The NAR method is found stable with $\mathsf{CFL}_{\mathsf{NA}}$ up to 0.9.

"*Time stretching*" *parameter*, $\mathscr{N}$. This parameter is calculated at the beginning of each time step $t^{(n+1)}$ (see item 1 in Algorithm B) to define the number of iterations required for the numerical acoustic wave to dissipate:

$$\mathscr{N} = \max\left[1, (\mathsf{int})\left(\frac{\mathsf{N}^{(\mathsf{M})}_{\mathsf{max}}}{\mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}}}\right)^2\right] \tag{27}$$

where $\mathsf{N}^{(\mathsf{M})}_{\mathsf{max}} = \frac{|\mathbf{u}_{\max}|}{c_{\mathrm{s}}}$; and $\mathbf{u}_{\max}$ is the maximum material velocity in the computational domain $\mathscr{D}$ at $t = t^{(n)}$. Value of $\mathscr{N}$ in a flow problem depends on a specified "effective numerical Mach number", $\mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}}$. On the one hand, the computational expense for each time step decreases with the increase of $\mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}}$. On the other hand, $\mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}}$ must be small enough to suppress the undesirable "compressibility" effects and limit the velocity divergence source term $\mathscr{D}_{\mathsf{A}}$ in Eq. (12). As shown in Section 5, calculations using $\mathsf{N}^{(\mathsf{M})}_{\mathsf{eff}} = 10^{-1}$ give results competitive with, or superior than, other Poisson-equation-based solvers.

3.5. Treatment of domain boundary conditions

In his original paper (Chorin, 1967), Chorin used a finite-difference method to solve the system of pseudocompressible Navier–Stokes equations, by adopting a "leap-frog" scheme for inertia and pressure terms (central difference in both time and space), and using "Dufort–Frankel"

[1] The "numerical acoustic CFL" number $\mathsf{CFL}_{\mathsf{NA}}$ in NAR resembles the "numerical sound speed" in LBE (Nourgaliev et al., 2003a)

$$c_{\mathrm{s}} = \sqrt{\frac{\Upsilon^{(4)}}{\Upsilon^{(2)}}}\frac{\delta_x}{\delta_t}$$

where $\Upsilon^{(4)}$ and $\Upsilon^{(2)}$ are the "symmetry parameters" of the chosen lattice. In a commonly-used LBE lattice geometry, $\mathsf{D}_2\mathsf{Q}_9$, $\frac{\Upsilon^{(4)}}{\Upsilon^{(2)}} = \frac{1}{3}$, rendering $\mathsf{CFL}_{\mathsf{NA}} \approx 0.57$.
At the same time, the LBE "numerical sound speed" is coupled with fluid kinematic viscosity by

$$\nu = \left(\tau - \frac{1}{2}\right)\delta_t c_{\mathrm{s}}^2$$

where τ is a "relaxation time" in LBE. This condition makes the LBE simulations of high-Re-number flows computationally expensive (see for details Nourgaliev et al., 2003a). No such limitations are present in NAR. However, it is yet to be seen what would be the Reynolds number limitations on NAR due to the pseudocompressibility.

pattern for discretization of viscous dissipation terms. At the wall boundary, the "pseudodensity" equation is discretized using explicit finite-difference formulas for velocity divergence, based on the velocity components available in the internal and boundary points of the computational mesh. In so doing, Chorin achieved a robust treatment of boundary conditions in a steady-state single-phase flow problem.

In the present work, we utilize a boundary condition treatment, that is an extension of the Ghost-Fluid concept (Section 4). Specifically, we consider the domain boundary as an interface separating a *real fluid* inside the computational domain from its *ghost* outside the domain. The boundary conditions are formulated by appropriately setting the value of variables at the *ghost* points. Similar approach for boundary treatment has also been implemented for the Characteristics-Based Matching method developed by the present authors for the simulation of compressible flow in complex geometry and compressible multiphase flow; for details see Nourgaliev et al. (2003b) and Nourgaliev et al. (in press).

In a viscid formulation, "Dirichlet" or "no-slip" boundary conditions $\mathbf{u}_{\mathsf{BC}}$ are imposed for all velocity components at the fluid–solid wall interface. Values of velocity in the *ghost* points at any iteration (m) are given by

$$\mathbf{u}_{\mathsf{G}}^{(m)} = 2\mathbf{u}_{\mathsf{BC}} - \mathbf{u}_{\mathsf{R}}^{(m)} \tag{28}$$

where G and R are the ghost and corresponding real points outside and inside of the computational domain $\mathscr{D}$. In the inviscid formulation, "reflection" boundary conditions are provided as:

$$u_{\mathsf{G}}^{\mathrm{T},(m)} = u_{\mathsf{R}}^{\mathrm{T},(m)}; \quad u_{\mathsf{BP}}^{\mathrm{N},(m)} = 0; \quad u_{\mathsf{G}}^{\mathrm{N},(m)} = -u_{\mathsf{R}}^{\mathrm{N},(m)} \tag{29}$$

where T and N denote the velocity's tangential and normal components, respectively.

In the absence of body forces, pressure in the ghost points are simply extrapolated from pressure values at the *boundary* points $P_{\mathsf{BP}}^{(m)} \to P_{\mathsf{G}}^{(m)}$ ("Neumann" boundary conditions). Given a body force, such as due to the gravitational acceleration g, the "extrapolated" pressure is defined as $P_{\mathsf{G}}^{(m)} = P_{\mathsf{BP}}^{(m)} + (\mathbf{r}_{\mathsf{G}} - \mathbf{r}_{\mathsf{BP}}) \cdot \mathbf{i} \cdot \rho_{\mathsf{BP}} \cdot g$, where $\mathbf{r}_{\mathsf{G}}$, $\mathbf{r}_{\mathsf{BP}}$ are the position-vectors of the *ghost* and *boundary* points, respectively; $\mathbf{i}$ is the vector specifying the direction of body force g; and ρ_{BP} is a fluid density at the *boundary* point.

4. Treatment of multifluid interfaces

Significant progress has been made over the past decade in the area of front/interface tracking and capturing. The eminent developments include the "volume-of-fluid (VOF)" method (Hirt and Nichols, 1981); immersed boundary method (Peskin, 1977); front-tracking approaches (Unverdi and Tryggvason, 1992; Glimm et al., 1998) and the "Level Set approach (LSA)" (Osher and Sethian, 1988). In the present work, we use the Level Set approach. A physical (e.g., fluid–fluid) interface $\mathbb{I}$ is described by a zero position of a so-called Level Set function $\varphi(t)$, which represents a signed distance function to the interface. The evolution of the surface $\mathbb{I}$, moving in the direction normal to itself with a speed function $\mathbb{F}$, can be described by the following Hamilton–Jacobi equation (Osher and Sethian, 1988)

$$\partial_t \varphi + \mathbb{F}|\nabla \varphi| = 0 \tag{30}$$

Subsequently, one may use a "regularized Heaviside" function to smear fluid/material properties (e.g., density and viscosity) over a few computational nodes in the vicinity of the interface (Chang et al., 1996). On the one hand, such a "smearing" procedure, also called diffusive interface approach, helps to suppress numerical instabilities of solution in a variable-density domain—a crucial issue in the implementation of the incompressible flow solvers based on Poisson equation. On the other hand, the discretization/resolution needed becomes increasingly more demanding with increasing velocity and/or density gradients. An alternative can be found in the recently introduced Ghost-Fluid method (GFM) (Fedkiw et al., 1999). It can accurately describe the necessary jump conditions within one node, and, having been demonstrated for compressible flows, it is naturally adaptable to our purposes here. The details are provided in the following.

Level Set algorithm implementation. In the present work, the speed function $\mathbb{F}$ is defined as follows

$$\mathbb{F} \equiv u_j \cdot \partial_j \varphi \tag{31}$$

which at the equation level is equivalent to what was utilized in Mulder et al. (1992):

$$\partial_t \varphi + \mathbf{u} \cdot \nabla \varphi = 0 \tag{32}$$

To solve Eq. (30), the third-order Runge–Kutta TVD scheme (Fedkiw et al., 1998) is employed for time discretization. For space discretization, we apply the fifth-order WENO_5 scheme (Jiang and Peng, 2000).

As the interface and flow evolve, it is necessary to reinitialize the Level Set function φ, to maintain it as the signed distance; see Chen et al. (1991) and Evans and Spruck (1991). Such a procedure of reinitialization (i) replaces $\varphi(t, \mathbf{r})$ by another function $d(\mathbf{r})$ that has the same zero level as φ but satisfying the property $|\nabla d(\mathbf{r})| = 1$; and, then, (ii) takes this new function $\varphi(t, \mathbf{r}) = d(\mathbf{r})$ as an initial condition for calculation until the next reinitialization. For this purpose, we apply a Partial Differential Equation-based (or "PDE-based") reinitialization approach introduced in Sussman et al. (1994) and further developed in Peng et al. (1999). The distance function satisfying condition $|\nabla d| = 1$ is obtained by solving to steady-state the following "pseudotransient" Hamilton–Jacobi equation:

$$\begin{cases} d_\tau + \mathbb{P}(d)(|\nabla d| - 1) = 0 \\ d(\mathbf{r}, \tau = 0) = d_0(\mathbf{r}) = \varphi(\mathbf{r}, t) \end{cases} \tag{33}$$

with

$$\mathbb{P}(d) = \frac{d}{\sqrt{d^2 + |\nabla d|^2 \Delta h^2}} \tag{34}$$

where $\Delta h = \min(\delta_x, \delta_z, \delta_y)$. To solve "pseudotransient" equation (33), Runge–Kutta TVD schemes are used; for details see Peng et al. (1999) and Sussman et al. (1994). Pseudotime step is given by $\Delta\tau = \mathsf{CFL}_{\text{reinit}} \cdot \Delta h$, where $\mathsf{CFL}_{\text{reinit}} < 1$ is a Courant number.

The speed function $\mathbb{F}$ is calculated using the PDE-based "extension" approach (Peng et al., 1999):

$$\psi_t + \underbrace{[\mathbb{Q}(\varphi)\cdot\vec{\mathbb{N}}]\cdot\nabla\psi}_{\mathscr{H}_{\mathscr{J}}(\nabla\psi,\mathbf{x},t)} = 0 \tag{35}$$

where $\psi = \mathbb{F}$; and $\mathbb{Q}(\varphi)$ is a signature function of φ if defined as

$$\begin{array}{ll}
\text{Extrapolate to both sides } |\varphi| \geqslant 0 & \mathbb{Q}(\varphi) = \begin{cases} -1 & \text{if } \varphi < 0 \\ 0 & \text{if } \varphi = 0 \\ +1 & \text{if } \varphi > 0 \end{cases} \\
\text{Extrapolate to the "positive" side } \varphi \geqslant 0 & \mathbb{Q}(\varphi) = \begin{cases} 0 & \text{if } \varphi \leqslant 0 \\ +1 & \text{if } \varphi > 0 \end{cases} \\
\text{Extrapolate to the "negative" side } \varphi < 0 & \mathbb{Q}(\varphi) = \begin{cases} 0 & \text{if } \varphi \geqslant 0 \\ -1 & \text{if } \varphi < 0 \end{cases}
\end{array} \tag{36}$$

Gradients of the Level Set function, necessary for evaluation of the normal vector $\vec{\mathbb{N}} = \frac{\nabla\varphi}{|\nabla\varphi|}$, are calculated using a second-order-accurate central differencing scheme. If the central differencing produces zero normal vector, the degeneracy is avoided by using the second-order-accurate upwinding (backward or forward).

The Hamiltonian $\mathscr{H}_{\mathscr{J}}$ of Eq. (35) is calculated using a first-order-accurate upwinding scheme (Peng et al., 1999):

$$\begin{aligned}
\mathscr{H}_{\mathscr{J}}|_{i,j,k} = {} & \max\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(x)}_{i,j,k}, 0\right)\frac{\psi_{i,j,k}-\psi_{i-1,j,k}}{\delta_x} + \min\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(x)}_{i,j,k}, 0\right)\frac{\psi_{i+1,j,k}-\psi_{i,j,k}}{\delta_x} \\
& + \max\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(z)}_{i,j,k}, 0\right)\frac{\psi_{i,j,k}-\psi_{i,j-1,k}}{\delta_z} + \min\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(z)}_{i,j,k}, 0\right)\frac{\psi_{i,j+1,k}-\psi_{i,j,k}}{\delta_z} \\
& + \max\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(y)}_{i,j,k}, 0\right)\frac{\psi_{i,j,k}-\psi_{i,j,k-1}}{\delta_y} + \min\left(\mathbb{Q}_{i,j,k}\mathbb{N}^{(y)}_{i,j,k}, 0\right)\frac{\psi_{i,j,k+1}-\psi_{i,j,k}}{\delta_y}
\end{aligned} \tag{37}$$

No additional treatment for boundary conditions is required, because characteristics of Eq. (35) flow out of the domain (Peng et al., 1999).

Ghost-Fluid method implementation. The main idea of the GFM is to introduce two sets ("fields") of variables, for each separate fluid modeled. Next, for each "field", zero level set separates *real* and *ghost* zones (Fig. 2). That is, the real fluid 1 coexists with the ghost fluid 2, and vice versa. Consequently, the task of implementing the GFM boils to the developing algorithms that "populate" field variables in the *ghost* zones of each fluid during each time step so that numerical solutions of separate field conservation equations are correctly coupled. Two algorithms, namely the 'soft' GFM coupling and the 'stiff' GFM coupling, were adopted here.

Soft GFM coupling. In this algorithm, both pressure and velocity are considered to be "continuous" functions across the fluid–fluid interface. Thus, the *ghost* points are populated by copying the values from the *real* fluid. Other properties, such as density and kinematic

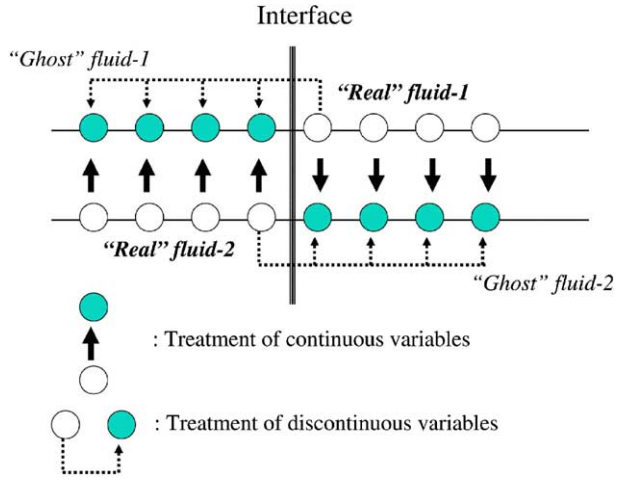

Fig. 2. On the formulation of the "Ghost-Fluid method" in one dimension.

viscosity, are discontinuous functions, which are "extrapolated" from the *real* nodes to the *ghost* zone.

However, it is noted that when the fluids involved differ significantly in density (e.g., water and air), small numerical errors in pressure calculation may lead to large errors in velocity for the lighter fluid. As a result, the 'soft' coupling algorithm may fail. The problem is akin to an issue of coupling of a "soft" equation of state (say, γ-gas) and a "stiff" equation of state (say, water) in compressible flow simulation (Fedkiw, 2002). In this work, we adopt the following 'stiff' coupling algorithm, first proposed by Fedkiw for the GFM treatment of multimaterial compressible media.

Stiff GFM coupling. It is natural to think of a high density fluid (e.g., water) as a "stiff" fluid and low density fluid (e.g., air) as a "soft" fluid. Pressure in the *ghost* region of the stiff fluid is copied from the *real* soft fluid. Pressure in the *ghost* region of the soft fluid is extrapolated from the *real* soft region using a PDE-based technique (see Eqs. (35) and (36)). Total velocity in the *ghost* zone of the stiff fluid is extrapolated from its *real* zone, while velocity in the *ghost* zone of the soft fluid is directly copied from the *real* stiff fluid.

In numerical exercises presented in Section 5, the "soft" coupling algorithm is used for multiphase flows with density ratio less than 10:1, and the "stiff" coupling algorithm was found to perform excellently in multiphase situations with density ratio of 1000:1. It is important to emphasize that "stiff" GFM coupling is essential for the successful simulation of high-density-ratio flows.

In addition, it may be worth pointing out that the Ghost-Fluid treatment is important to eliminate divergence errors due to density jump across the interface. The first equation of the system (14) can be written as

$$\partial_j u_j = \mathscr{D}_{\mathsf{A}} - u_j \partial_j \tilde{\rho}$$

where, in addition to the divergence errors of the artificial compressibility $\mathscr{D}_{\mathsf{A}}$, the system experiences density ratio errors of the order $\mathrm{O}(\frac{\delta\rho}{\Delta x})$. The GF treatment effectively eliminates the density jump which is seen by each fluid. As a result, nearly divergence-free states are achieved on either

side of the interface. As it is shown in Sections 5.3 and 5.4, the total divergence errors are insignificant.

5. Numerical results and comparative analysis

In the following, the "Numerical Acoustic Relaxation" method performance is examined on four flow tests. Section 5.1 describes the NAR simulation of the "Lid-Driven Cavity flow" LDC test (steady-state single-phase problem), and compares its numerical results with that of obtained by "vorticity-stream-function based" and "approximate projection based" methods. Section 5.2 presents the NAR application to "Doubly Periodic Shear Layer Flow" ('SL') (transient single-phase problem). The NAR results are compared with those obtained by a "pseudospectral method". In Section 5.3, the NAR method is applied to a multiphase flow problem—a low density ratio (Atwood number) "Rayleigh–Taylor Instability", and compared to results by a "Lattice Boltzmann Equation" method and analytical results. Finally, in Section 5.4, the NAR method is applied for the simulation of "Collapse of Water Column", a high density ratio free-interface problem, and the results are compared to the "Moving Particle Semi-Implicit" (MPS) method.

5.1. Lid-Driven Cavity flow

Problem formulation. The 'Lid-Driven Cavity' ('LDC') flow has been established as a standard "benchmark" test for numerical methods of incompressible fluid dynamics. In Ghia et al. (1982), Ghia et al. employed finite-difference method, using $(\omega - \psi)$ formulation of incompressible flow, to obtain solutions in a wide range of Re numbers. Using grids with high resolution (129×129 and 257×257) and 'coupled strongly implicit multigrid' method, Ghia et al. provided "reference" data for comparison of velocity profiles at the vertical and horizontal centerlines of the square cavity, with no-slip vertical and bottom walls, and moving top wall.

NAR realization. The NAR calculations of the "LDC" test are performed for two Re numbers, 400 and 1000, using computational grids 50×50, 100×100 and 128×128. Since the comparison is sought for the steady-state solution, "numerical Mach number" does not present any limitation ($\mathsf{N}_{\mathsf{eff}}^{(\mathsf{M})} = 1$). As the dimensionless time reaches $t = 30\ldots40$, all "numerical acoustic waves" have essentially decayed, providing a completely divergence-free velocity field, $\mathscr{D}_{\mathsf{A}} \approx 0$, Eq. (12).

Calculation results. The NAR-calculated streamline patterns for cases $\mathsf{Re} = 400$ and 1000 are shown on Fig. 3. Qualitatively, it can be seen that the NAR solution captures primary, secondary and additional corner vortices, with flow structures similar to those given in Ghia et al. (1982). Quantitative comparison is given in Figs. 4 and 5. An excellent agreement between the NAR result and the "$\omega - \psi$-based" method result is evident.

Figs. 6 and 7 present the calculated velocity profiles at the cavity's vertical and horizontal centerlines. Even on a coarse grid resolution (50×50), the NAR method produces results that

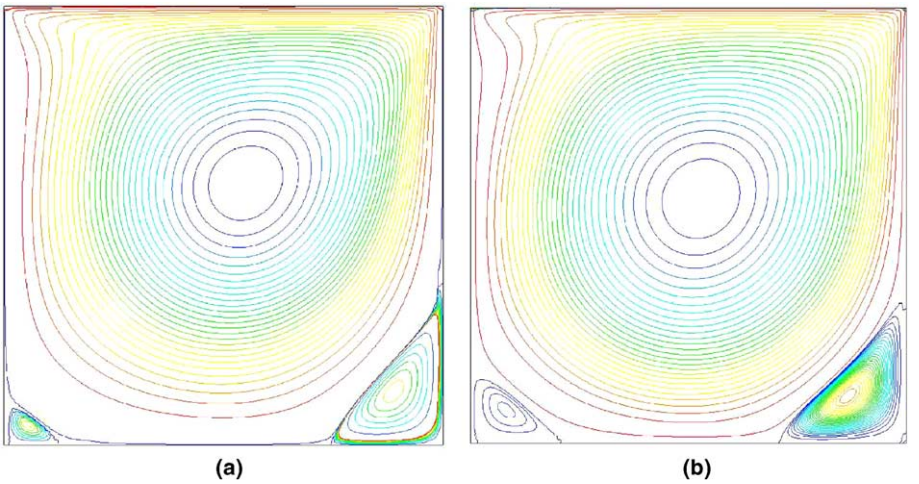

Fig. 3. Lid-Driven Cavity test. Streamline pattern for "LDC" test. NAR calculation. (a) Re = 400. Grid resolution: 100×100; (b) Re = 1000. Grid resolution: 128×128.

agree well with the reference result of Ghia et al. using $(\omega-\psi)$-formulation on the grid 128×128. In addition, Figs. 6 and 7 also show results obtained by the present authors using an Approximate Projection (AP) method, "$\mathrm{WENO}^{5}_{\mathrm{CB-CFD}}$-AP" (conservative finite-difference WENO_5 scheme for inertia terms, and the "vertex-centered" divergence). Notably, due to the NAR high accuracy, the NAR results on 50×50 grid are as good as the solution by the AP scheme on 100×100 grid.

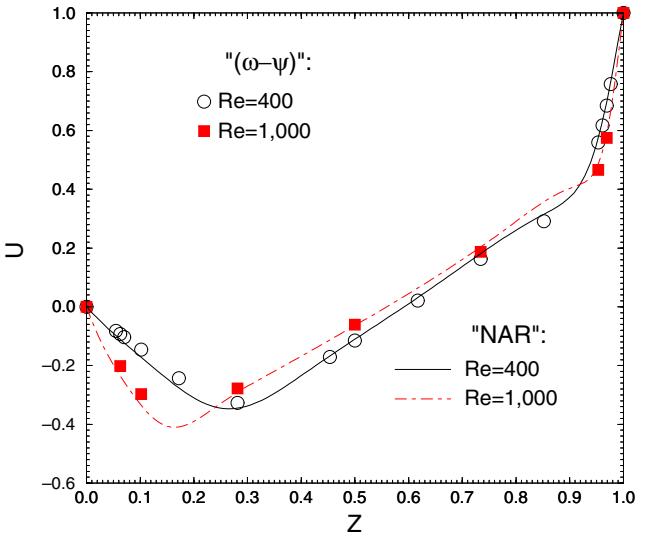

Fig. 4. Lid-Driven Cavity test. Comparison of the velocity profile at the cavity's vertical centerline. Re = 400 and 1000. Computational data obtained using (a) $(\omega-\psi)$ formulation (Ghia et al., 1982); (b) NAR (present work).

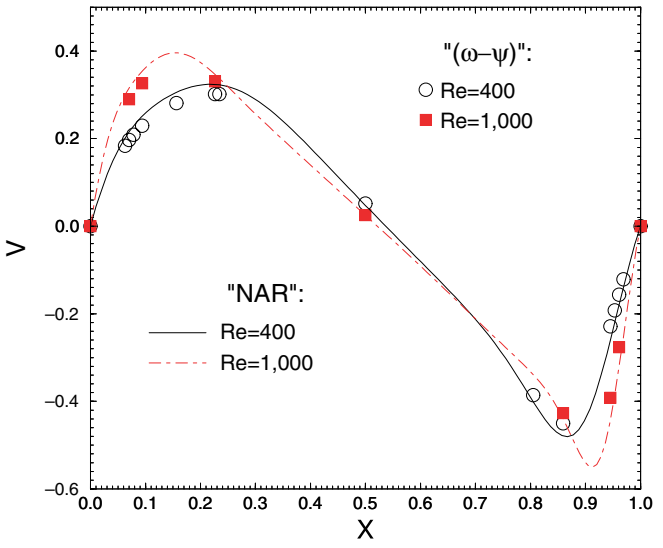

Fig. 5. Lid-Driven Cavity test. Comparison of the velocity profile at the cavity's horizontal centerline. Re = 400 and 1000. Computational results obtained using (a) $(\omega - \psi)$ formulation (Ghia et al., 1982); (b) NAR (present work).

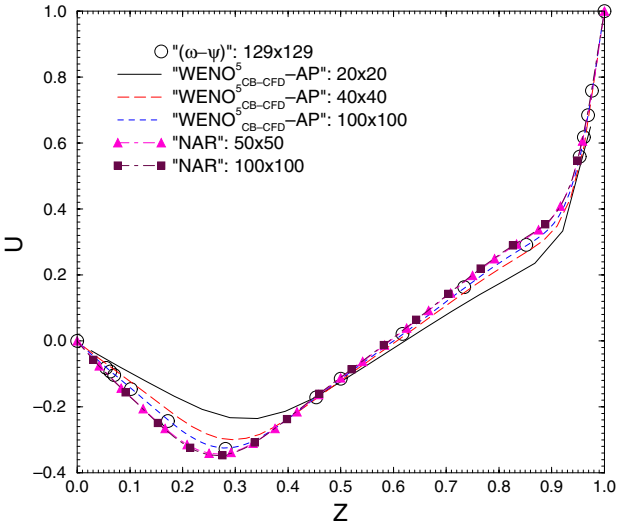

Fig. 6. Lid-Driven Cavity test. Velocity profiles at the cavity's vertical centerline, Re = 400. Results obtained using (a) $(\omega - \psi)$ formulation (Ghia et al., 1982); (b) characteristics-based conservative finite-difference scheme with WENO_5 for inertia terms and vertex-centered approximate projection "$\mathrm{WENO}^5_{\mathrm{CB-CFD}}$–AP" (by present authors); (c) NAR (present work).

5.2. Doubly Periodic Shear Layer problem

Problem formulation. Introduced by Bell et al. (1989) and examined more recently in Minion and Brown (1997), the present test considers two jets in a doubly periodical domain of size $[1 \times 1]$,

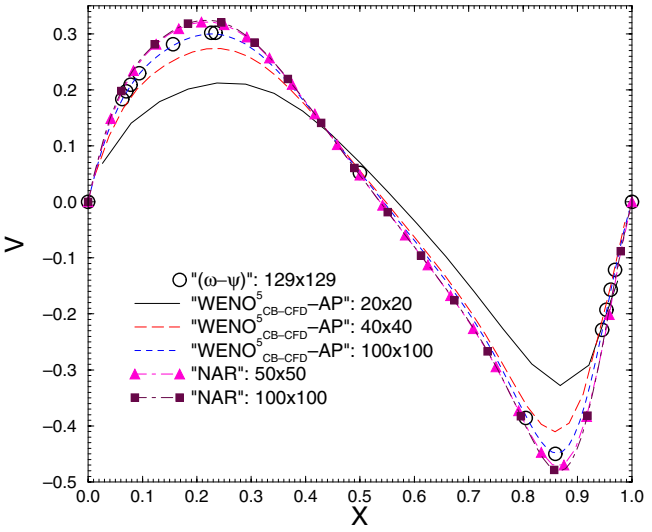

Fig. 7. Lid-Driven Cavity test. Velocity profiles at the cavity's horizontal centerline.

to which a sinusoidal perturbation perpendicular to the plane of motion is imposed at the lowest wavenumber supported by the computational mesh. The initial pressure field is uniform $P = 0$ and initial velocity fields are given by:

$$u = \begin{cases} \tanh(\vartheta(z - 0.25)), & \text{for } z \leqslant 0.5 \\ \tanh(\vartheta(0.75 - z)), & \text{for } z > 0.5 \end{cases} \qquad (38)$$
$$v = \gamma \sin(2\pi(x + 0.25))$$

where ϑ is the shear layer width parameter and γ is the strength of the initial perturbation. In the absence of any additional perturbations, each of the shear layers rolls up in a single vortex as the flow evolves.

Two width parameters of the shear layer, $\vartheta = 30$ and 80, are employed in the calculation, corresponding to a "thick" and "thin" layer configuration. The perturbation strength $\gamma = 0.05$ was unchanged. The NAR calculations are performed on 32×32, 64×64, 128×128 and 256×256 grids.

Calculation results. For a sufficiently thick shear layer ($\vartheta = 30$) the NAR result (Fig. 8) obtained on a coarse grid of 64×64 is comparable to that of (Rider, 1994) by an Approximate Projection method.

For thin layer configurations (e.g., $\vartheta = 80$ and $\mathrm{Re} = 10,000$), the grid resolution was found to have a substantial effect on the vorticity field, Fig. 9, leading to formation of spurious vortices. It can be seen in the figure that these spurious vortices disappear with a better grid resolution. As shown by Minion and Brown (1997), this feature is common for all numerical methods of incompressible flow. There, they examined the "pseudospectral", the "Godunov Projection", the "centered $(\omega - \psi)$-based", the upwind $(\omega - \psi)$-based", the "Pressure-Poisson-based", and the "ENO-Projection" methods (Minion and Brown, 1997). Fig. 10 shows the NAR transient solu-

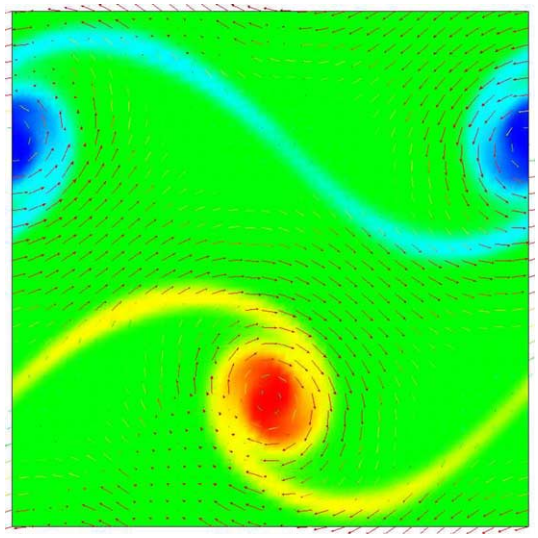

Fig. 8. Doubly Periodic Shear Layer test. Vorticity and velocity fields for a "thick shear layer test". NAR algorithm with $\mathsf{N}_{\mathrm{eff}}^{(\mathsf{M})} = 0.1$. Grid resolution: 64×64. Calculational results for $t = 1$.

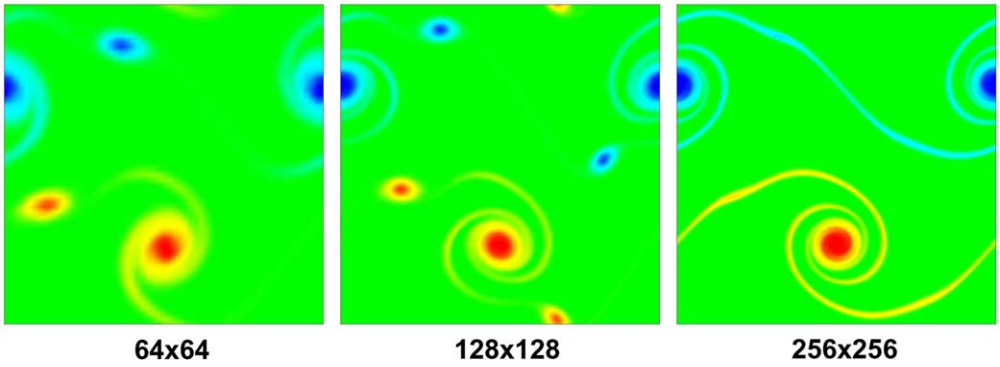

Fig. 9. Doubly Periodic Shear Layer test. Effect of grid resolution on vorticity field. NAR algorithm with $\mathsf{N}_{\mathrm{eff}}^{(\mathsf{M})} = 0.1$. Grid resolution 64×64, 128×128, and 256×256.

tion on 256×256 grid, that is qualitatively comparable with high-resolution results given in Minion and Brown (1997).

Quantitatively, the comparison favors the NAR performance. Fig. 11 shows the history of total enstrophy, $\sum_{i,j}^{\mathfrak{D}} \frac{\omega_{(i,j)}^2}{2} \frac{\Delta \mathsf{A}_{i,j}}{\mathsf{A}}$ ($\Delta \mathsf{A}_{i,j}$ and A are the local cell and total area, respectively), as obtained by NAR and by the pseudospectral method. It can be seen that the NAR solution on 256×256 grid is as accurate as the pseudospectral method solution on 768×768 grid. Note, the results on a low-resolution grid show a slightly accelerated decay of the total enstrophy. The same trend is observed with the increase of the effective numerical Mach number.

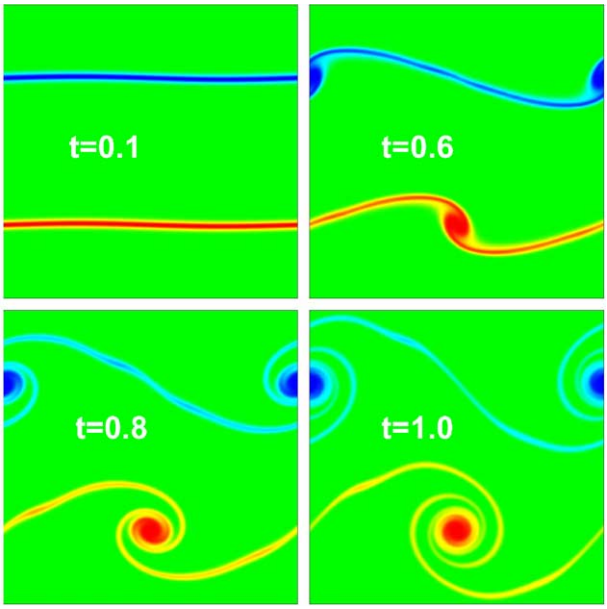

Fig. 10. Doubly Periodic Shear Layer test. Dynamics of vorticity field for thin shear layer. NAR algorithm with $N_{eff}^{(M)} = 0.1$. Grid resolution 256×256.

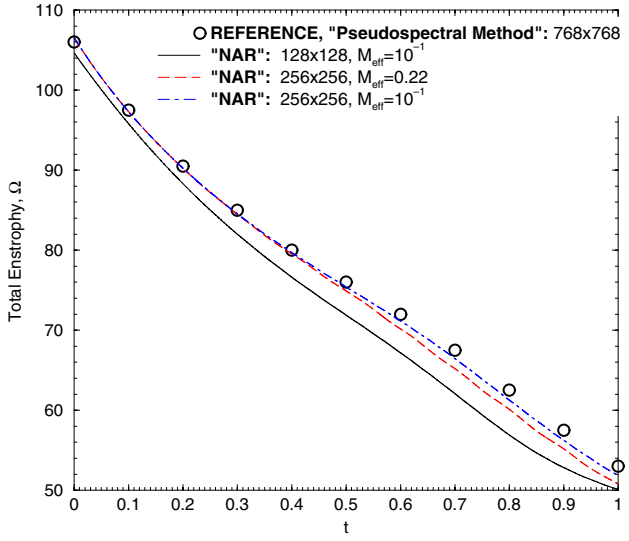

Fig. 11. Doubly Periodic Shear Layer test. Total enstrophy decay for the thin shear layer test. Comparison of the NAR results with solution by "pseudospectral method" (Minion and Brown, 1997).

Fig. 12 shows a decay of the flow's total kinetic energy, $\sum_{i,j}^{\mathfrak{D}} \frac{|\mathbf{u}|^2_{(i,j)}}{2} \frac{\Delta A_{i,j}}{A}$, as obtained by NAR and other methods (Minion and Brown, 1997). In general, the total kinetic energy decays faster on poor grids. Also, under elevated numerical Mach number, oscillations appear and "$\mathscr{D}_A$-errors" increase. It can be seen that the "upwind $(\omega - \psi)$-based" method produces a significantly faster

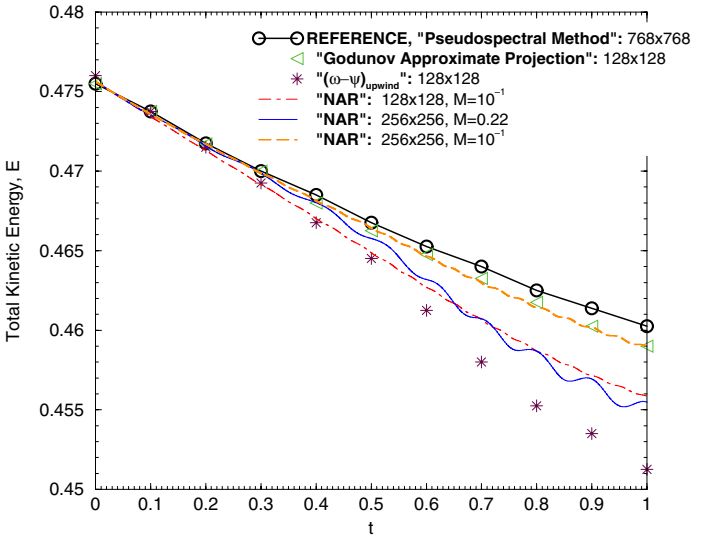

Fig. 12. Doubly Periodic Shear Layer test. Total kinetic energy decay for the thin shear layer test. Comparison of the NAR results with solution by "pseudospectral", "Godunov Approximate Projection" and "$(\omega - \psi_{\text{upwind}})$" methods (Minion and Brown, 1997).

decay. At the same time, both the "Godunov Approximate Projection" method and the NAR with a sufficient time stretching $\left(\mathsf{N}_{\text{eff}}^{(\mathsf{M})} = 0.1\right)$ give accurate results, that approach the pseudo-spectral solution on 768×768 grid.

5.3. Viscous Rayleigh–Taylor instability

The objective of this test is to examine the performance of the NAR for simulation of multi-phase flow.

Problem formulation. We consider a two-dimensional Rayleigh–Taylor instability problem in a domain of size $[1 \times 2]$. A heavy fluid (density ρ_{h}) is placed on a top of a layer of a lighter fluid (density ρ_{l}), rendering the initial planar interface (at $z = 1$) unstable under the gravitational acceleration. It is known that in such situations, any small perturbation will lead to the heavy fluid moving downwards as "spikes", the light fluid moving upwards as "bubbles" (Sharp, 1984). Initially, both heavy and light fluids are motionless, and the initial pressure field is prescribed by the local hydraulic head. At $t = 0$, a small local perturbation

$$\mathsf{h}(x) = \mathfrak{A} \cos\left(\frac{2\pi}{\mathsf{W}} x\right)$$

is added to the interface, with W and $\mathfrak{A}$ being the width of the computational domain and the amplitude of the initial perturbation, respectively. Periodic boundary conditions are imposed on the side boundaries, while no-slip boundary conditions are used for the top and bottom boundaries of the computational domain.

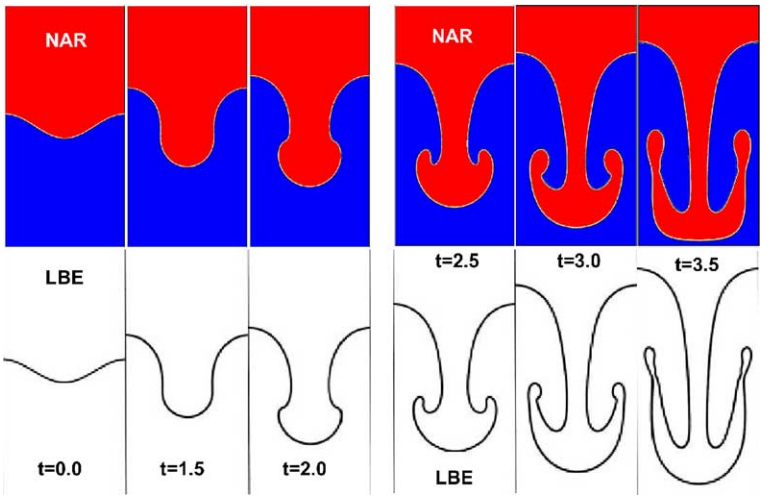

Fig. 13. Rayleigh–Taylor Instability test. Comparison of the NAR and "He–Chen–Zhang" LBE (He et al., 1999) methods. Parameters of the test case (NAR/LBE): At = 0.5; Re = 256; single-mode initial perturbation with amplitude $\mathfrak{A} = 0.1 \cdot W$; aspect ratio—$(1{:}2)_{NAR}/(1{:}4)_{LBE}$; grid resolution—$(128 \times 256)_{NAR}/(256 \times 1056)_{LBE}$. For NAR: "top" fluid has density $\rho = 3$; while "bottom" fluid has density $\rho = 1$.

Calculation results. Fig. 13 depicts the calculated dynamics of the interface. A computational grid 128×256 is used. The results obtained by a Lattice Boltzmann Equation method (He et al., 1999) are also shown in the figure. Fig. 14 shows the evolution of the leading front positions of spike and bubble. One can see the agreement between the LBE and NAR results. Fig. 15 shows

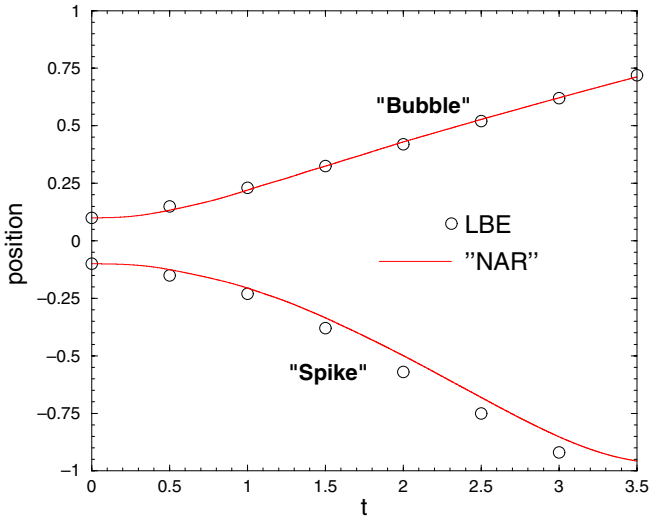

Fig. 14. Rayleigh–Taylor Instability test. Positions of the "bubble" and "spike" fronts versus time. At = 0.5; Re = 256. Comparison of the NAR calculation (grid: 128×256; aspect ratio: 1:2; $N_{eff}^{(M)} = 0.1$) with the result by LBE approach (He et al., 1999) (grid: 256×1024; aspect ratio: 1:4).

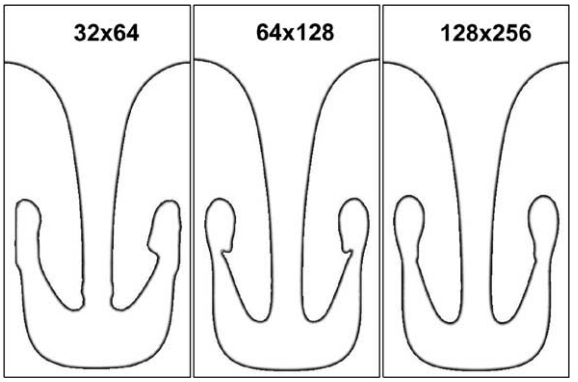

Fig. 15. Rayleigh–Taylor Instability test. Effect of grid resolution on the position of the interface. Calculations performed on grid 32×64, 64×128 and 128×256 for $\mathrm{At}=0.5$; $\mathrm{Re}=256$. Aspect ratio: 1:2. $\mathrm{N}_{\mathrm{eff}}^{(\mathrm{M})}=0.1$, $t=3.5$.

the NAR calculated results obtained on different grids, 32×64, 64×128 and 128 x 256, at $t=2.5$. The convergence is evident.

Next, we examine the NAR results against the prediction by the linear stability theory. During the initial stages of the Rayleigh–Taylor instability, the perturbation amplitude is significantly smaller than the wavelength, the equations can be linearized, and the perturbation of the fluid–fluid interface exhibits an exponential growth (Chandrasekhar, 1981):

$$\mathrm{h}(t)=\mathrm{h}_0\mathrm{e}^{\hat{\alpha}t} \tag{39}$$

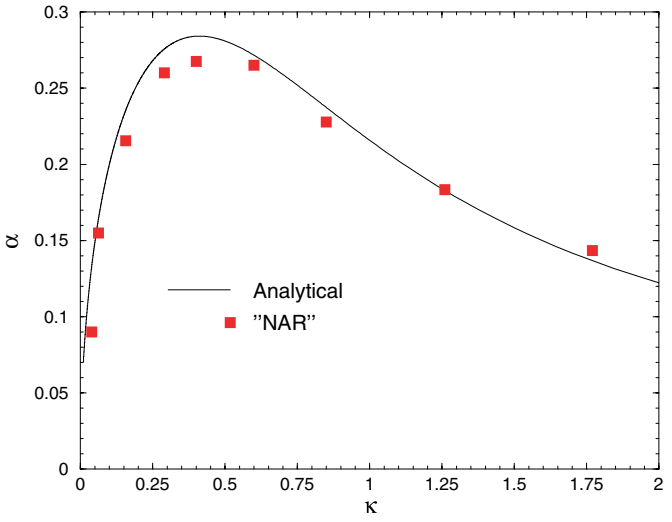

Fig. 16. Rayleigh–Taylor Instability test. The dependence of the growth rate, α (measured in units of $(\frac{g^2}{\nu})^{1/3}$), of a disturbance on its wavenumber, κ (measured in units of $(\frac{g}{\nu^2})^{1/3}$). $\mathrm{At}=0.5$. Comparison with analytical solution by Chandrasekhar (1981).

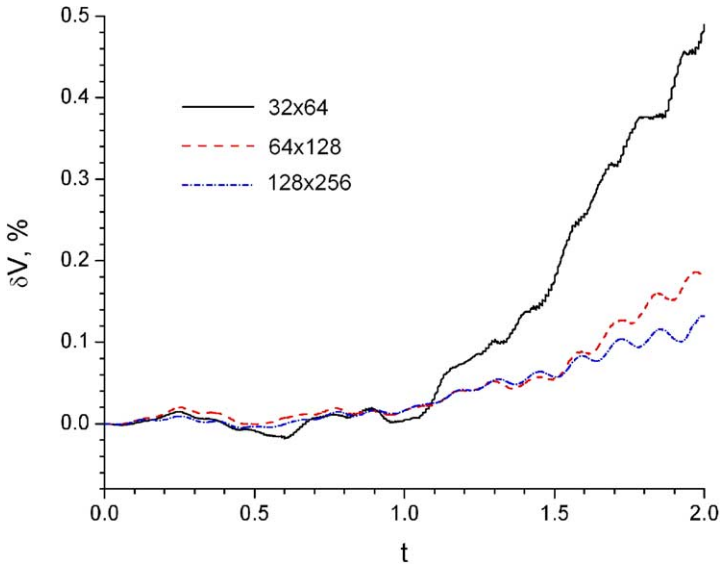

Fig. 17. Volume conservation errors for Rayleigh–Taylor Instability test.

where $\mathsf{h}(t)$ is the perturbation amplitude at time t; h_0 is the amplitude of the initial perturbation, and $\hat{\alpha}$ is the growth rate. Fig. 16 shows an analytical solution obtained by Chandrasekhar (1981) for the dependence of the dimensionless growth rate $\alpha = \frac{\hat{\alpha}}{[g^2/\nu]^{1/3}}$ on the disturbance's wavenumber $\kappa = \frac{2\pi}{\mathsf{W}[g/\nu^2]^{1/3}}$. Numerically, we can vary κ by changing the fluid kinematic viscosity ν in the NAR simulations. In all cases shown here, a single-mode perturbation with an initial perturbation

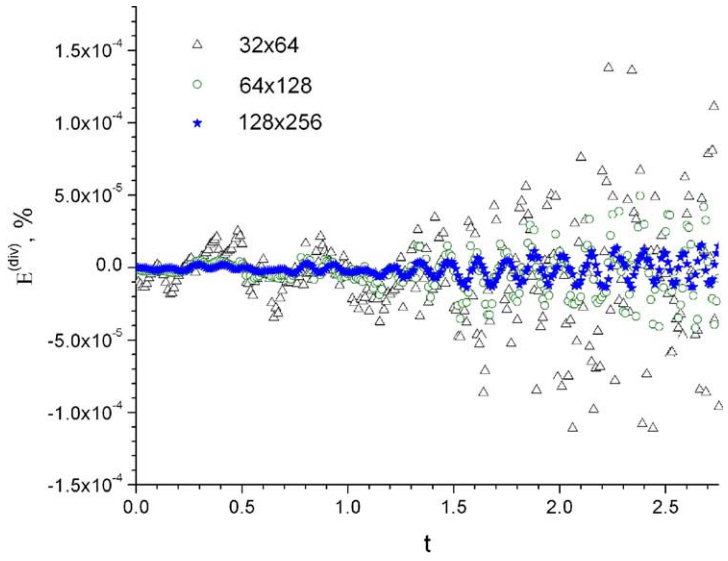

Fig. 18. Mass conservation (divergence) errors for Rayleigh–Taylor Instability test.

amplitude of $\mathfrak{A} = 0.05 \cdot \mathsf{W}$ was used. Simulations are performed on 64×128 grid. The measurement of the growth rate, $\alpha(t) = \frac{\ln \frac{\mathsf{h}(t-t_0)}{\mathsf{h}_0(t_0)}}{t-t_0}$, is reinitialized every 200 computational steps by re-setting t_0 and correspondent h_0. As it can be seen from Fig. 16, the NAR results agree excellently with the linear stability analytical prediction.

Conservation errors. Finally, we present mass and volume conservation errors in Figs. 17–20 and Table 1. The volume of the light fluid was computed at each time step using a piecewise-linear reconstruction of the interface based on cross-sections of zero level set with grid lines. As it can be seen from Fig. 17, the volume conservation errors are negligibly small (<0.5%). Moreover, they diminish when computational grid is refined.

The Level Set method is known for not strictly enforcing volume conservation. However, from our experience, the actual loss/gain of volume happens when the interface curvature R_i becomes of the order of the grid size, $\delta R_i \equiv \frac{R_i}{\Delta h} \to 1$. With a better grid $\Delta h \to 0$ and $\delta R_i \uparrow$, the volume conservation errors are going down, see Fig. 17.

The divergence errors are computed as:

$$\delta M_{(i,j,k)} = \left[\frac{\mathsf{F}^{(P)}_{i+\frac{1}{2},j,k} - \mathsf{F}^{(P)}_{i-\frac{1}{2},j,k}}{\Delta x} + \frac{\mathsf{G}^{(P)}_{i,j+\frac{1}{2},k} - \mathsf{G}^{(P)}_{i,j-\frac{1}{2},k}}{\Delta z} + \frac{\mathsf{H}^{(P)}_{i,j,k+\frac{1}{2}} - \mathsf{H}^{(P)}_{i,j,k-\frac{1}{2}}}{\Delta y} \right] \frac{\Delta t \Delta x \Delta z \Delta y}{c_s^2}, \ [\mathrm{kg}] \tag{40}$$

and

$$\begin{aligned} E^{\mathrm{div}} &= \frac{\sum_{i,j,k}^{\text{all nodes}} \delta M_{(i,j,k)}}{M_{\mathrm{tot}}} \cdot 100, \ [\%] \\ \mathscr{L}_1^{\mathrm{div}} &= \frac{\sum_{i,j,k}^{\text{all nodes}} \frac{|\delta M_{(i,j,k)}|}{M_{(i,j,k)}}}{N_{\mathrm{tot}}} \\ \mathscr{L}_2^{\mathrm{div}} &= \sqrt{\frac{\sum_{i,j,k}^{\text{all nodes}} \frac{(\delta M_{(i,j,k)})^2}{M_{(i,j,k)}}}{N_{\mathrm{tot}}}} \end{aligned} \tag{41}$$

where $\mathsf{F}^{(P)}$, $\mathsf{G}^{(P)}$ and $\mathsf{H}^{(P)}$ are flux terms for pressure equation (16); while N_{tot}, M_{tot} and $M_{(i,j,k)}$ are the total number computational nodes, the total mass of fluid, and local mass of the cell (i,j,k), respectively.

The histories of these divergence errors are plotted in Figs. 18–20. It can be seen that the mass losses are negligibly small. Furthermore, with a refined grid under the same numerical Mach number, the divergence errors are going down. The effects of artificial compressibility are evaluated in Table 1 and Fig. 20. It can be seen that errors can be controlled by appropriate selection of the numerical Mach number.

5.4. Collapse of Water Column

The objective of this test is to examine the performance of the NAR method in a multiphase flow problem with very large density ratio—a domain where traditional Poisson equation-based

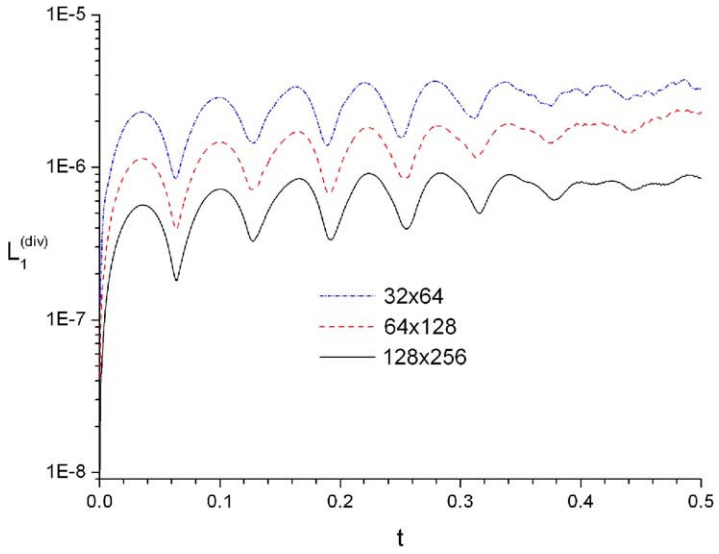

Fig. 19. Mass conservation (divergence) errors for Rayleigh–Taylor Instability test (continued).

methods perform poorly, because of necessity to solve Poisson equations with significantly varied coefficients.

Problem formulation. A water column of size $L \times 2L$ is initially positioned in a closed box of size $4L \times 2.2L$, where $L = 14.6$ cm, Fig. 21. The water column collapses in the gravity field $g = 9.8$ m/s^2. Water and air densities are 1000 and 1 kg/m^3, respectively. The initial pressure field is generated according to the local hydraulic head.

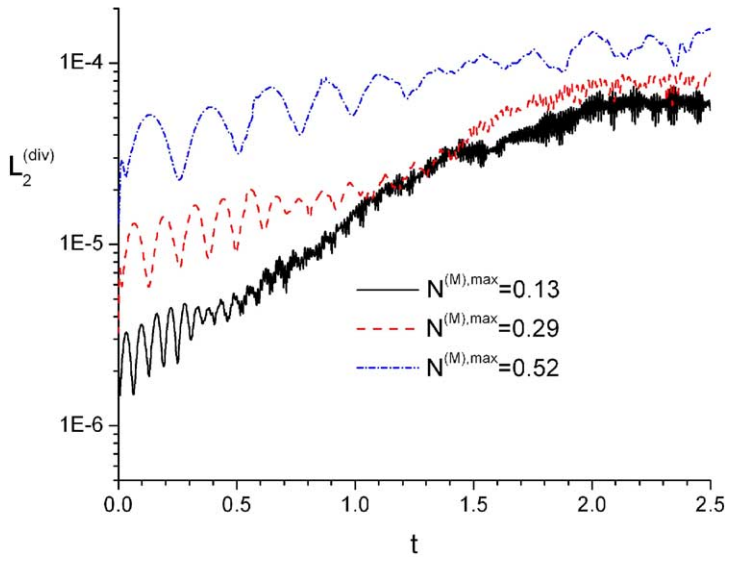

Fig. 20. The effect of numerical Mach number on $\mathscr{L}_2$ norm of divergence errors for Rayleigh–Taylor Instability test.

Table 1
On mass conservation (divergence) errors for Rayleigh–Taylor Instability test

$N^{(M)}$	0.13	0.29	0.52
$E^{(\mathrm{div}),\max}$, %	1.588×10^{-4}	2.9×10^{-4}	1.48×10^{-3}
$\mathscr{L}_1^{(\mathrm{div}),\max}$	5.6×10^{-5}	5.7×10^{-5}	1.18×10^{-4}
$\mathscr{L}_2^{(\mathrm{div}),\max}$	8.7×10^{-5}	8.8×10^{-5}	1.83×10^{-4}

The effect of numerical Mach number value chosen for the computation. Maximum divergence errors are taken over transient time from $t = 0$ to 3.

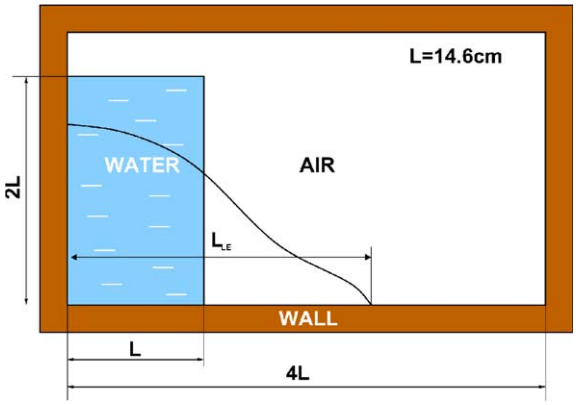

Fig. 21. Water Column Collapse test. Problem formulation.

NAR realization. No viscous and surface tension effects are included. Slip ("velocity reflection") boundary conditions are imposed at the box's walls. At the water–air interface, 'stiff coupling' is applied: from the air ('soft' fluid) side, the water is seen as a piston and the air's ghost nodes take the value of the local velocity of water. The air pressure in the ghost nodes is extrapolated from the air' real zone side. For water (stiff fluid), the pressure in the water ghost nodes takes the local value of the calculated air pressure. Velocity of the water ghost nodes is extrapolated from the water' real zone side.

Calculational results. Fig. 22 shows the calculated dynamics of the water–air interface and the velocity field. A grid of 110×200 was used. The NAR results are in a good agreement with experimental observations (Koshizuka and Oka, 1996). In Fig. 23, the calculated leading edge position is compared with the experimental data of Koshizuka and Oka (1996) and Martin and Moyce (1952). The numerical results by other methods (MPS, Koshizuka and Oka, 1996; SOLA-VOF, Hirt and Nichols, 1981) are also shown. A slightly faster motion of the calculated leading edge is due to ignoring the wall friction. As shown in Fig. 24, the mass conservation errors are insignificant. Also, the volume conservation is maintained within 4%.

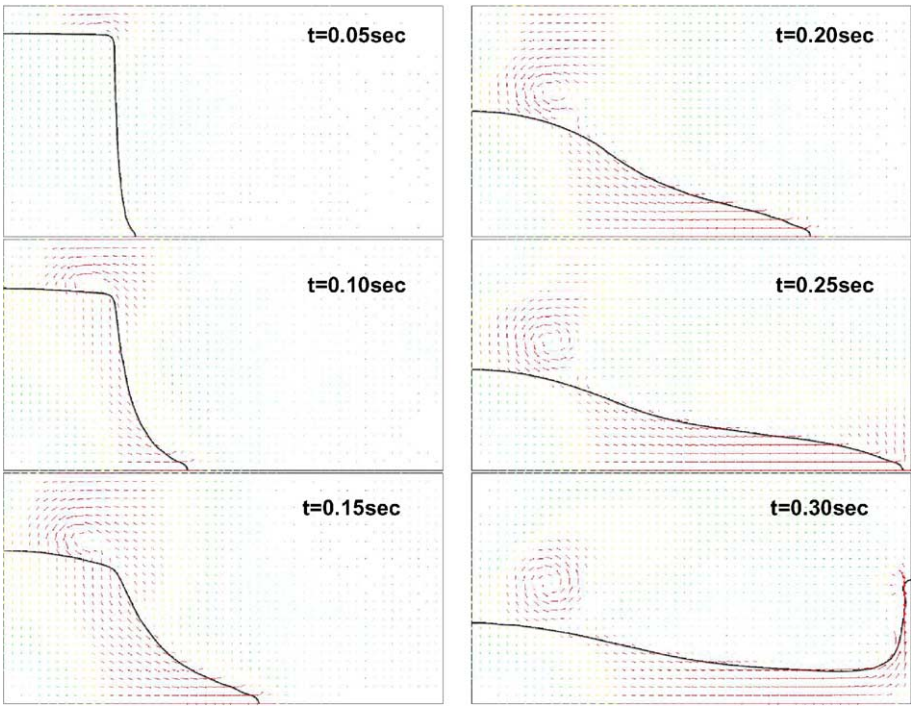

Fig. 22. Water Column Collapse test. Dynamics of water column collapse.

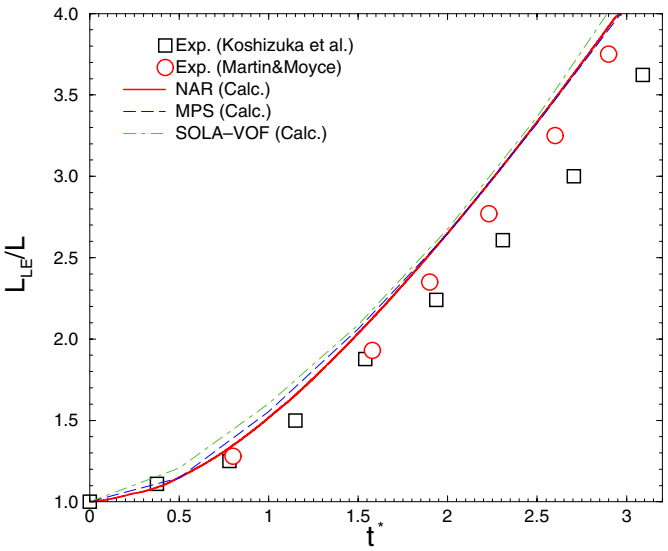

Fig. 23. Water Column Collapse test. Motion of leading edge in water column collapse. Dimensionless time defined as $t^* = t\sqrt{\frac{2g}{L}}$.

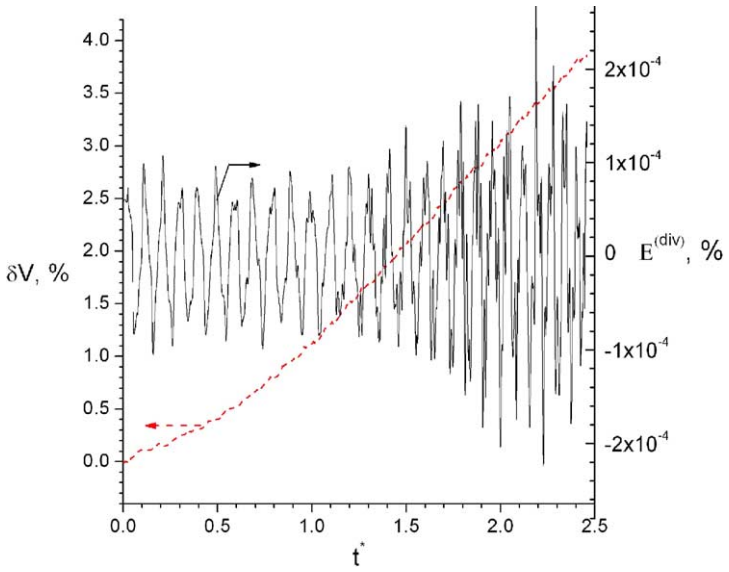

Fig. 24. Volume and mass conservation (divergence) errors for Water Column Collapse test.

6. Discussion

The Numerical Acoustic Relaxation (NAR) method exhibits an excellent performance in all four test problems examined in the preceding sections. Here, we will discuss pros and cons of the NAR, relative to other methods for incompressible single and multifluid dynamics.

Accuracy. The use of the high-order-accurate characteristics-based scheme for the treatment of hyperbolic terms gives the NAR method remarkable accuracy and robustness. Importantly, the discrete divergence is calculated by using the fifth-order-accurate WENO_5 upwind differencing discretization scheme (Jiang and Shu, 1996). However, there always remain errors due to the velocity divergence terms associated with numerical acoustic waves, $\mathscr{D}_{\mathsf{A}} \ll 1$. The NAR's "time stretching" procedure is designed to effectively increase the influence zone of the numerical acoustics in each real-time step. This allows the "background" pressure waves to propagate far enough, and the fluid's velocity to relax to a *nearly divergence-free state*. The total error is determined by the dominance between finite-difference discretization and the numerical acoustic errors.

Velocity divergence discretization errors exist even in the implicit (Poisson-based) methods. In this case, the divergence source errors appear after the predictor step. The correction step is designed to eliminate these errors (by solving Poisson equation), which requires a linear algebra solver. Solvability properties of the linear algebra often dictate that the exact projection, based on the central differencing

$$\partial_j u_j \overset{\text{"Exact" Projection}}{\rightarrow} \frac{u_{(i+1,j,k)} - u_{(i-1,jk)}}{2\delta_x} + \frac{v_{(i,j+1,k)} - v_{(i,j-1,k)}}{2\delta_z} + \frac{w_{(i,j,k+1)} - w_{(i,j,k-1)}}{2\delta_y} + \mathrm{O}\left(\delta_x^2, \delta_z^2, \delta_y^2\right)$$

is substituted by "approximate projections" (Rider, 1994). This is realized by de-collocating grids for pressure and velocity components, which sacrifices the "idempotent" property of the projection operator and introduces stabilizing numerical smearing effects. In our examples, we utilized the Approximate Projection (AP) method ("$\mathrm{ENO}_{\mathrm{CB-CFD}}$-AP"), in which the discrete velocity divergence is formulated using the following "pressure-vertex-centered" finite-difference formula:

$$\partial_j u_j \overset{\text{"Approximate" Projection}}{\rightarrow} \frac{u_{(i+\frac{1}{2},j+\frac{1}{2})} + u_{(i+\frac{1}{2},j-\frac{1}{2})}}{2\delta_x} - \frac{u_{(i-\frac{1}{2},j+\frac{1}{2})} + u_{(i-\frac{1}{2},j-\frac{1}{2})}}{2\delta_x} + \frac{v_{(i+\frac{1}{2},j+\frac{1}{2})} + v_{(i-\frac{1}{2},j+\frac{1}{2})}}{2\delta_z} - \frac{v_{(i+\frac{1}{2},j-\frac{1}{2})} + v_{(i-\frac{1}{2},j-\frac{1}{2})}}{2\delta_z}$$

Additional errors in representing the incompressibility condition are due to the formulation of the velocity divergence-free conditions at the cell's corners, whereas the "conserved" variables are defined in the cell's center. In the numerical examples shown in this paper, the discretization errors of the Approximate Projection method and the NAR method are found to be comparable, when the numerical Mach number in NAR was sufficiently small, $\mathsf{N}_{\mathrm{eff}}^{(\mathsf{M})} < 10^{-1}$. Numerical acoustics errors will be discussed below.

Convergence. That the NAR scheme is able to provide accurate solutions was demonstrated in Section 5. Here, we will study convergence rate of the NAR. For this, we will consider a flow problem called "Travelling Wave", that has a known analytical solution. Again, we refer to Minion and Brown (1997), where the performance of several implicit methods on this incompressible flow problem were examined.

The computational domain of $[1\times 1]$ is doubly periodic, with an initially generated periodical vortex structures convected by a flow field (Fig. 25) and exponentially decaying due to the viscous dissipation. In the present test, the viscosity is chosen as $\nu = 10^{-2}$ (see Minion and Brown (1997) for definition of dimensionless variables). The exact solution for this problem is:

$$\begin{aligned} u(x,z,t) &= 1 + 2\cos(2\pi(x-t))\sin(2\pi(z-t))\mathrm{e}^{-8\pi^2\nu t} \\ v(x,z,t) &= 1 - 2\sin(2\pi(x-t))\cos(2\pi(z-t))\mathrm{e}^{-8\pi^2\nu t} \\ p(x,z,t) &= -(\cos(4\pi(x-t)) + \cos(4\pi(z-t)))\mathrm{e}^{-16\pi^2\nu t} \end{aligned} \tag{42}$$

Fig. 26 depicts L_1 norm of the error for the u velocity component at $t = 0.7$, in comparison with the results of reference Minion and Brown (1997). It can be seen that, for the travelling wave problem, the NAR exhibits a high convergence rate on 32×32 and 64×64 grids, and a slowdown for 128×128. Specifically, the convergence rate reached 4.63 before it dropped to 1.36. Note that, for the travelling wave problem, very small numerical errors are already achieved with NAR on 64×64 grid. Consequently, the slow down of convergence at higher grid resolutions is not surprising, as discretization errors become small and comparable with the errors due to the artificial compressibility. The grid size at which the total numerical errors start to level off can be increased by decreasing the numerical Mach number, however, this is apparently at the cost of a higher computational expense needed to advance the solution in each physical time step.

Fig. 26 also shows the L_1 norm of the solution obtained by other methods, including the popular Godunov Approximate Projection method and other more sophisticated schemes. It can be seen that the NAR has significantly better convergence rates than that of the Godunov

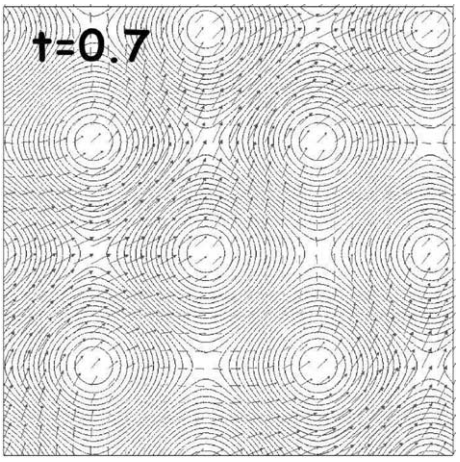

Fig. 25. Travelling Wave test. Pressure and velocity field calculated by NAR on 128×128 grid.

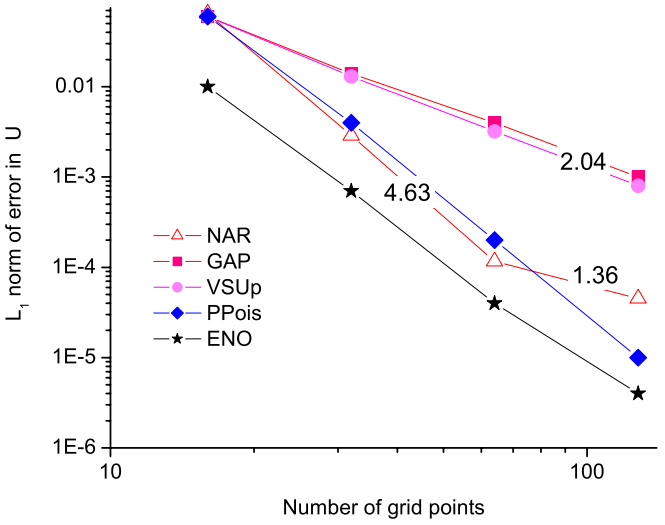

Fig. 26. Travelling Wave test. Convergence result: L_1 norm of error. NAR: present work; GAP, VSUp, PPois and ENO: are due to Minion and Brown (1997). GAP: Godunov Approximate Projection method; VSUp: upwind vorticity stream function; PPois: pressure Poisson, ENO: Essentially Non-Oscillatory.

Approximate Projection and the upwind vorticity stream-function methods. The NAR also has a slightly better convergence rate than that of the fourth-order central difference pressure Poisson method and the ENO projection method for grids up to 64×64. The ENO projection offers a better accuracy than NAR, as do the centered vorticity stream-function and the pseudospectral methods (not shown in the figure). Combined with $\mathscr{L}_1$ norm shown in Figs. 18 and 19 for the departure from divergence-free condition (the NAR accuracy improves with grid refinement), the

above comparison demonstrates that the NAR convergence performance is comparable with the performance of existing methods for incompressible flow.

Efficiency. The incompressible flow methods based on the Poisson equation require linear algebra solvers, which to a large extent determine the efficiency of the overall computational procedure. Notably, even best available linear algebra solvers significantly deteriorate with the increase of the computational grid and geometrical complexity (Bernsdorf et al., 1999). Another important area, where the linear algebra presents a liability, is the flow with large density variations such as in multiphase flows. The NAR method does not suffer such limitations imposed by the linear algebra solvers.

From the performance standpoint, one of NAR's time stretching iterative step can arguably be viewed as a counterpart of one sweep in the Poisson solver's iterative procedure. However, one iteration step of NAR is computationally more expensive than one iteration sweep of the Poisson solver. The NAR iteration requires re-calculation of all source terms that include hyperbolic terms, viscous diffusion and body forces. Therefore, in flow simulations with simple domain geometry and small number of computational nodes, the NAR is inferior to implicit schemes, whose efficiency is high in these cases when a small number of iteration sweeps is needed to converge linear algebra. However, the Poisson solvers exhibit very slow convergence in simulations of single-phase flow in complex domain or multiphase flow with large density variations, and with large number of computational nodes. There exists a "break-even-point", when the NAR method becomes more efficient than the 'implicit' Poisson-based method. For example, in the Rayleigh–Taylor instability problem (fluid density ratio 1:7) the break-even-point occurs for grid 128×512, when the NAR simulation utilizes $\mathsf{M}_{\mathsf{eff}} = 10^{-1}$ and the implicit method is based on the "$\mathsf{ENO}_{\mathsf{CB-CFD}}$-AP" scheme and utilizes the "variable-density Algebraic MultiGrid (AMG)" method (Alcouffe et al., 1981) as a linear algebra solver. The "break-even-point" for larger density ratios happens under smaller computational grid. As an aside, while we could get very good results with AP scheme, high-grid-resolution calculations for density ratios greater than 10 turn out to be so hard to converge, as to be impractical. This underlines the computational challenge when resolution requirements increase (finer and finer grid) in the presence of large density ratios, such as encountered in breakup of interfaces under conditions that inertia forces cannot be neglected.

Parallelizability. The NAR algorithm is highly suitable for the parallelization strategy using domain decomposition strategy. That is, the computational domain is subdivided into subdomains, corresponding to the number of processors available. The problem solution is carried out for each subdomain on a separate processor and data communication between the processors is via a Message Passing Interface (MPI). Notably, the NAR's single collocated mesh frees the parallelization from significant complications associated with de-coupled pressure–velocity "nested" grids, typically utilized in Poisson equation-based methods, using multigrid linear algebra solvers. Furthermore, since the NAR algorithm is local, the data exchange is required only at the end of each stretched-time iteration. Therefore, the NAR algorithm more effectively utilizes the communication load since it requires less number of iterations than the number of the iteration sweeps in the Poisson equation solver. Scalability of the NAR algorithm is found to be in the range 0.9–0.95, which is quite close to the 'ultimate scalability' limit of 1.

It is of interest to compare the NAR method to the Lattice Boltzmann Equation (LBE) method. In a previous study (Nourgaliev et al., 2003a), we related advantageous features of the LBE method over Poisson-equation-based solvers to the LBE's pseudocompressible formulation. The development of NAR and results presented in this paper confirm that the NAR method exhibits, and even exceeds, similar algorithmic advantages once assigned to the LBE method (parallelizability, simplicity). Specifically, parallelization with the NAR is more effective than with the LBE: in 2D, the NAR's processor-network requires four neighbor-processors versus eight neighbor-processors in the LBE $\mathsf{D_2Q_9}$ scheme. In comparison to the LBE, the NAR also requires less variables to be stored ($[P,u,v]_{2\mathrm{D}}$ versus $[P,u,v,f_a,f_a^{(\mathrm{eq})}]_{\mathsf{D_2Q_9}},\ a=0,\ldots,8$), and less PDEs to be solved (3 versus 9). Numerically, the NAR is more accurate than the LBE: the LBE's stream-and-collide strategy has, at its best, the second-order accuracy, while the NAR employs the fourth- and fifth-order-accurate discretization schemes for viscous diffusion and hyperbolic terms, respectively. Perhaps more importantly, the NAR is more flexible and can be applied for viscous, high-Reynolds-number and inviscid flows, while the LBE method is practically limited to low-Reynolds-number flows (see discussion in Nourgaliev et al. (2003a)).

7. Conclusions

Based on the experience derived from the test cases considered in this paper, we can surmise the following attributes for the NAR:

1. The numerical scheme appears suitable for the robust and accurate simulation of incompressible single and multiphase flows, by solving directly the Navier–Stokes equations.
2. Divergence-free errors are effectively controlled with an appropriate choice of the numerical Mach number (typically $\mathsf{N}^{(\mathsf{M})} < 0.1$), and with the high-order $\mathsf{WENO_5}$ scheme the accuracy is the same or better than the Approximate Projection method, on the same grid, for simple single-phase flows.
3. For multiphase flow, conservation errors near interfaces are effectively controlled by the Ghost-Fluid treatment, even at very large (actually arbitrary) density discontinuities.
4. The NAR method overtakes in efficiency the methods based on Poisson equation (e.g. Approximate Projection) in highly resolved simulations, as the density ratio increases. In the problems (and the AP solver) tested in this paper, we found the break-even-point to be around a density ratio of 7.

Based on the above, we can conclude that NAR holds promise for the class of problems needing highly resolved computations of multifluid flows involving very large density ratios, as found in gas–liquid flows when inertia effects are important. The explicit treatment and algorithmic simplicity of NAR provide additional advantages for massively parallel simulations.

Acknowledgements

This work performed under support of NASA (grant NAG3-2119), and Lawrence Livermore National Laboratory (“ALPHA” and “MIX” projects).

References

Alcouffe, R.E., Brantd, A., Dendy Jr., J.E., Painter, J.W., 1981. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. SIAM Journal on Scientific and Statistical Computing 2, 430–454.

Aris, R., 1962. Vectors, Tensors, and the Basic Equations of Fluid Mechanics. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Bell, J.B., Colella, P., Glaz, H.M., 1989. An efficient second-order projection method for the incompressible Navier–Stokes equation. Journal of Computational Physics 85, 257–283.

Bernsdorf, J., Durst, F., Schäfer, A., 1999. Comparison of cellular automata and finite volume techniques for simulation of incompressible flows in complex geometries. International Journal for Numerical Methods in Fluids 29, 251–264.

Chandrasekhar, S., 1981. Hydrodynamic and Hydromagnetic Stability. Dover Publications Inc., New York. pp. 444–449.

Chang, Y.C., Hou, T.Y., Merriman, B., Osher, S., 1996. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. Journal of Computational Physics 124, 449–464.

Chen, S., Doolen, G.D., 1998. Lattice Boltzmann method for fluid flows. Annual Review of Fluid Mechanics 30, 329–364.

Chen, Y.G., Giga, Y., Goto, S., 1991. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. Journal of Differential Geometry 33, 749–786.

Chorin, A.J., 1967. A numerical method for solving incompressible viscous flow problems. Journal of Computational Physics 2, 12 (Recent edition, 1997: 135, 118–125).

Dafermos, C.M., 2000. Hyperbolic Conservation Laws in Continuum Physics. Springer, Berlin. 443p.

Evans, L.C., Spruck, J., 1991. Motion of level set via mean curvature I. Journal of Differential Geometry 33, 635–681.

Fedkiw, R.P., 2002. Coupling an eulerian fluid calculation to a lagrangian solid calculation with the Ghost-Fluid method. Journal of Computational Physics 175, 200–224.

Fedkiw, R., Merriman, B., Donat, R., Osher, S., 1998. The penultimate scheme for systems of conservation laws: finite difference ENO with Marquina's flux splitting. In: Hafez, M. (Ed.), Progress in Numerical Solutions of Partial Differential Equations, Arcachon, France, July 1998.

Fedkiw, R.P., Aslam, T., Merriman, B., Osher, S., 1999. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the Ghost-Fluid method). Journal of Computational Physics 152, 457–492.

Ghia, U., Ghia, K.N., Shin, C.T., 1982. High-*Re* solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. Journal of Computational Physics 48, 387–411.

Glimm, J., Grove, J.W., Li, X.L., Shyue, K.-M., Zeng, Y., Zhang, Q., 1998. Three-dimensional front tracking. SIAM Journal on Scientific Computing 19, 703–727.

He, X., Chen, S., Zhang, R., 1999. A Lattice Boltzmann scheme for incompressible multiphase flow and its application in simulation of Rayleigh–Taylor instability. Journal of Computational Physics 152, 642–663.

Hirt, C.W., Nichols, B.D., 1981. Volume of fluid (VOF) methods for the dynamics of free boundaries. Journal of Computational Physics 39, 201–225.

Hirt, C.W., Amsden, A.A., Cook, J.L., 1974. An arbitrary lagrangian–eulerian computing method for all flow speeds. Journal of Computational Physics 14, 201–225.

Jiang, G.S., Peng, D.P., 2000. Weighted ENO schemes for Hamilton–Jacobi equations. SIAM Journal on Scientific Computing 21, 2126–2143.

Jiang, G.S., Shu, C.-W., 1996. Efficient implementation of weighted ENO schemes. Journal of Computational Physics 126, 202–228.

Koshizuka, S., Oka, Y., 1996. Moving-Particle Semi-Implicit method for fragmentation of incompressible fluid. Nuclear Science and Engineering 123, 421–434.

Kwak, D., Kiris, C., 2003. Successes and challenges of incompressible flow simulation. In: 16th AIAA Computational Fluid Dynamics Conference, Orlando, FL, June 23–26, 2003.

Landau, L.D., Lifschitz, E.M., 1988. Hydrodynamics, fourth ed. In: Theoretical Physics, vol. VI. Nauka, Moscow (Chapter II.15, in Russian).

Ladd, A.J.C., Verberg, R., 2001. Lattice-Boltzmann simulations of particle–fluid suspensions. Journal of Statistical Physics 104, 1191–1251.

Lax, P.D., 1973. Hyperbolic systems of conservation laws and the mathematical theory of shock waves. SIAM, Series in Applied Mathematics, 11.

Le Veque, R.J., 1992. Numerical Methods for Conservation Laws. Birkhäuser Verlag, Basel.

Martin, J.C., Moyce, W.J., 1952. An experimental study of the collapse of liquid columns on a rigid horizontal plane. Philosophical Transactions of the Royal Society of London, Series A 244, 312–324.

Minion, M.L., Brown, D.L., 1997. Performance of under-resolved two-dimensional incompressible flow simulations II. Journal of Computational Physics 138, 734–765.

Mulder, W., Osher, S., Sethian, J.A., 1992. Computing interface motion in compressible gas dynamics. Journal of Computational Physics 100, 209–228.

Nourgaliev, R.R., Dinh, T.N., Theofanous, T.G., Joseph, D., 2003a. The Lattice Boltzmann Equation method: theoretical interpretation, numerics, and implications. International Journal of Multiphase Flow 29, 117–169.

Nourgaliev, R.R., Dinh, T.N., Sushchikh, S.Y., Yuen, W.W., Theofanous, T.G., 2003b. The Characteristics-Based Matching method for compressible flow in complex geometries. In: 41st Aerospace Sciences Meeting (Fluid Dynamics), Reno, Nevada, Jan 2003, AIAA-2003-0247.

Nourgaliev, R.R., Dinh, T.N., Theofanous, T.G. The Characteristics-Based matching (CBM) method for compressible flow with moving boundaries and interfaces. ASME Journal of Fluids Engineering, in press.

Oran, E.S., Boris, J.P., 1987. Numerical Simulation of Reactive Flow. Elsevier Science Publishing Co., Inc., New York, Amsterdam, London.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. Journal of Computational Physics 79, 12–49.

Peng, D.P., Merriman, B., Osher, S., Zhao, H., Kang, M., 1999. A PDE-based fast local Level Set method. Journal of Computational Physics 155, 410–438.

Peskin, C.S., 1977. Numerical analysis of blood flow in the heart. Journal of Computational Physics 25, 220–252.

Rider, W.J., 1994. Filtering non-solenoidal modes in numerical solutions of incompressible flows. International Journal of Numerical Methods in Fluid 28, 789–814, Also, Approximate Projections methods for incompressible flow: implementation, variants and robustness, Technical Report LA-UR-2000, Los Alamos National Laboratory, 1994.

Rogers, S.E., Kwak, D., 1990. Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. AIAA Journal 28, 253–262.

Rogers, S.E., Kwak, D., 1991. Steady and unsteady solutions of the incompressible Navier–Stokes equations. AIAA Journal 29, 603–610.

Sharp, D.H., 1984. An overview of Rayleigh–Taylor instability. Physica D 12, 3–18.

Sussman, M., Smereka, P., Osher, S., 1994. A Level Set method for computing solutions to incompressible two-phase flow. Journal of Computational Physics 119, 146–159.

Turkel, E., 1987. Preconditioned methods for solving the incompressible and low speed compressible equations. Journal of Computational Physics 72, 277–298.

Unverdi, S.O., Tryggvason, G., 1992. A front-tracking method for viscous, incompressible, multi-fluid flows. Journal of Computational Physics 100, 25–37.

van Leer, B., Lee, W.T., Roe, P., 1991. Characteristic time-stepping or local preconditioning, of the Euler equations. In: AIAA 10th CFD Conference, June 1991, AIAA Paper 91-1552.

von Newmann, J., Richtmyer, R., 1950. A method for the numerical calculation of hydro-dynamic shocks. Journal of Applied Physics 21, 232–243.