SANDIA REPORT

SAND2013-2608
Unlimited Release
Printed March 2013

Developing Highly Scalable Fluid Solvers for Enabling Multiphysics Simulation

Jonathan R. Clausen

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-2608
Unlimited Release
Printed March 2013

Developing Highly Scalable Fluid Solvers for Enabling Multiphysics Simulation

Jonathan R. Clausen
Thermal Fluid Processes
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0828
jclause@sandia.gov

Abstract

We performed an investigation into explicit algorithms for the simulation of incompressible flows using methods with a finite, but small amount of compressibility added. Such methods include the artificial compressibility method and the lattice-Boltzmann method. The impetus for investigating such techniques stems from the increasing use of parallel computation at all levels (processors, clusters, and graphics processing units). Explicit algorithms have the potential to leverage these resources.

In our investigation, a new form of artificial compressibility was derived. This method, referred to as the Entropically Damped Artificial Compressibility (EDAC) method, demonstrated superior results to traditional artificial compressibility methods by damping the numerical acoustic waves associated with these methods. Performance nearing that of the lattice-Boltzmann technique was observed, without the requirement of recasting the problem in terms of particle distribution functions; continuum variables may be used.

Several example problems were investigated using a finite-difference and finite-element discretizations of the EDAC equations. Example problems included lid-driven cavity flow, a convecting Taylor-Green vortex, a doubly periodic shear layer, freely decaying turbulence, and flow over a square cylinder. Additionally, a scalability study was performed using in excess of one million processing cores. Explicit methods were found to have desirable scaling properties; however, some robustness and general applicability issues remained.

This page intentionally left blank.

Contents

Nomenclature **7**

1 Introduction **11**

2 Thermodynamic Limit of Incompressibility **14**
2.1 Isentropic Limit 15
2.2 Minimization of Density Fluctuations 16
2.3 Discussion 17

3 Finite Difference Implementation **18**
3.1 Numerical Method 18
3.2 Characteristic Analysis 19
3.3 Results 20

4 Finite Element Implementation **29**
4.1 Numerical Method 29
4.2 Results 30

5 Scalability Study **37**

6 Conclusions **40**

References **42**

Figures

1 Schematic of characteristic waves entering and exiting the domain at flow inlet and outlet. The wave propagation speeds are also shown. 20
2 Streamlines from the EDAC simulations at Ma = 0.1 showing the formation of three vortices: the primary, bottom left, and bottom right. At Re = 5000 an additional vortex forms along the top left of the domain, as is expected. Warmer colors indicate higher stream function values. Figure from Clausen (2013) [13]. 21
3 LDCF results for Re = 5000 showing excellent agreement for the (a) u and (b) v velocity components Figure from Clausen (2013) [13]. 22
4 Traveling wave solution showing the pressure (a) and velocity divergence (b) at location $(1/2, 1/2)$ at Re = 1. KRLNS method error creates qualitatively incorrect behavior unless correction term is included. Figure from Clausen (2013) [13]. 23
5 Traveling wave solution showing the pressure (a) and velocity divergence (b) at location $(1/2, 1/2)$ at Re = 1. Pressure and velocity divergence errors exist for the AC and LB methods. The EDAC method rapidly damps the fluctuations showing greatly reduced errors. Figure from Clausen (2013) [13]. 24
6 Images of the $\omega_z = 0.8$ vorticity contours at times (a) $t = 0$, (b) $t = 1$, (c) $t = 2$, (d) $t = 4$, and (e) $t = 8$. Maximum enstrophy occurs near (d). 27

7 Freely decaying turbulence showing the (a) energy and (b) enstrophy decay. 28
8 Zoom near object showing mesh used for simulating flow over square cylinder. . . . 31
9 Pressure field created by flow over a blunt object demonstrating (a) pressure locking phenomena along the leading edge where solution gradients are under-resolved, and (b) amelioration of pressure oscillations with suitable pressure stabilization. . . 32
10 Flow over square cylinder showing a von Kármán vortex street. Images show (a) pressure (b) u-component of velocity, (b) v-component of velocity, and (c) vorticity. 33
11 Comparison of u and v velocity components with the FD–EDAC method and existing numerical results. 34
12 Vorticity magnitude for (a) FE–EDAC and (b) SUPG simulations of the doubly periodic shear layer. 35
13 The decay of (a) energy and (b) enstrophy for a doubly periodic shear layer simulation using the FE–EDAC method. 36
14 Strong scaling studies for the $1024 \times 1024 \times 1024$ domain size showing the (a) scalability and (b) parallel efficiency. 38
15 Strong scaling studies for the $2048 \times 2048 \times 2048$ domain size showing the (a) scalability and (b) parallel efficiency. 39
16 Weak scaling studies performed at a fixed computational load of 32^3 computational nodes per core. 39

Nomenclature

AC artificial compressibility

CNS compressible Navier–Stokes

EDAC entropically damped artificial compressibility

GFLOPS billion floating point operations per second

GPU graphics processing unit

INS incompressible Navier–Stokes

KRLNS kinetically reduced local Navier–Stokes

LB lattice Boltzmann

Ma Mach number

MMS method of manufactured solutions

NS Navier–Stokes

Pr Prandtl number

Re Reynolds number

St Strouhal number

SUPG streamline-upwind/Petrov Galerkin

c_p specific heat at constant pressure

c_s speed of sound

c_v specific heat at constant volume

E total kinetic energy

e_k kinetic energy

f frequency

$\boldsymbol{F}(\boldsymbol{U})$ x-direction advection in vector form

g covariant element metric tensor

$\mathcal{G}$ grand potential

$\boldsymbol{G}(\boldsymbol{U})$ y-direction advection in vector form

$\mathbf{J}_F$ flux Jacobians for $\boldsymbol{F}$

$\mathbf{J}_G$ flux Jacobians for $\boldsymbol{G}$

k thermal conductivity

L length scale

n wall-normal direction

P pressure

P_0 reference pressure

q pressure test function

$\mathbf{S}$ rate of strain tensor

s specific entropy

$\boldsymbol{S}(\boldsymbol{U})$ source terms in vector form

t time

t_a acoustic time scale

t_c convective time scale

t_d diffusive time scale

T temperature

T_0 reference temperature

$\boldsymbol{U}$ solution unknowns in vector form

$\boldsymbol{u}$ velocity

u_0 reference velocity

u_∞ free-stream velocity

V simulation domain volume

V_i cell volume

$\boldsymbol{w}$ velocity test function

α isothermal compressibility coefficient

β thermal expansion coefficient

γ ratio of specific heats

Δx x-direction grid spacing

Δy y-direction grid spacing

Δt time step

ϵ kinetic energy dissipation

η Kolmogorov length scale

$\mathbf{\Lambda}$ characteristic eigenvectors

λ_i characteristic eigenvalues

μ dynamic viscosity

ν kinematic viscosity

ρ density

ρ_0 reference density

$\boldsymbol{\sigma}$ viscous stress tensor

τ Kolmogorov time scale

τ_{LB} lattice-Boltzmann relaxation time

τ_P tau parameter for pressure stabilization

τ_{SUPG} tau parameter for SUPG stabilization

Φ viscous dissipation

Ω enstrophy

$(,)$ scalar product in L^2

This page intentionally left blank.

1 Introduction

The governing equations of continuum fluid mechanics are the Navier–Stokes (NS) equations. These equations can be broadly split into two classes: compressible and incompressible. For the compressible case, no assumptions are made regarding the Mach number, and compressibility effects cannot be ignored. The governing equations include mass conservation, momentum conservation, energy conservation, and a thermodynamic equation of state, collectively referred to as the compressible Navier–Stokes (CNS). Compressibility effects are governed by the Mach number, $\mathrm{Ma} \equiv u_0/c_s$, where u_0 is the flow velocity and c_s is the speed of sound. Time scales of interest include the acoustic time scale, $t_a \sim L/c_s$, where L is a length scale of interest, the diffusive time scale, $t_d \sim L^2/\nu$, where ν is the fluid kinematic viscosity, and the convective timescale, $t_c \sim L/u_0$.

A common simplification of the more general compressible Navier–Stokes (CNS) equations involves taking the low Mach number limit ($\mathrm{Ma} \ll 1$) to derive the incompressible Navier–Stokes (INS) equations. In this case, the acoustic timescale becomes singularly perturbed, i.e., acoustic pressure disturbances are transmitted instantaneously. The thermodynamic state becomes fixed, thus the mass and momentum equations become decoupled from the energy equation, simplifying the system considerably compared with the CNS equations. The strict decoupling of the energy equation found in the INS can be weakened somewhat via an acoustic filtering procedure that separates the pressure into mechanical and thermodynamic parts, thus yielding a temperature-only equation of state for the thermodynamic pressure [33]. These equations allow for a temperature-dependent density and are typically used in cases where flow-driven compressibility effects are negligible, yet temperature gradients create large density gradients that cannot be adequately described via a Boussinesq approximation.

Although the INS represents a conceptual simplification of the CNS equations, solving the INS using computational techniques still remains challenging. The instantaneous propagation of pressure fluctuations (acoustic waves) results in an elliptic-type system that requires an implicit formulation and nonlocal communication. One method of solution is via the pressure-Poisson solve common to so many INS solvers. When attempting to scale computational techniques on many processing cores, this nonlocality results in high interprocessor communication, and consequently, nontrivial parallel scalability. One successful class of solution strategies is the fractional step approach. These methods have been developed based on a Hodge decomposition of the velocity field, pioneered with the original projection methods of [11, 12]. These methods rely on some variation of operator splitting [34, 19] to decompose the problem into several steps, which can often be generalized as the following: a velocity update to an intermediate value, application of an incompressibility constraint, and a velocity correction step to maintain a solenoidal velocity field. Original formulations of the projection schemes satisfied the incompressibility constraint discretely; however, more recent formulations have favored approximate forms that only satisfy the discrete divergence to the numerical accuracy of the method [2, 28, 7]. This point is important, since although the equations associated with projection methods may be strictly incompressible, the discrete solution will contain numerical errors resulting in some degree of nonzero velocity divergence. The form of the incompressibility constraint is typically formed using the pressure-Poisson form of the continuity equation, which is elliptic and thus requires an iterative approach to

solving. A more detailed review on projection-based methods can be found in [7].

The complexity associated with the scalable solution to the INS is compounded by current trends in high-performance computing. Next generation so-called "exascale" supercomputers will have processing units (cores) in excess of one-million. For example, the current DOE system Sequoia[1] contains 1.6 million cores. This trend is also exacerbated by use of graphics processing units (GPUs) for general purpose computation. GPUs achieve their formidable computational throughput through massively parallel execution, even within a single GPU. For comparison an Intel Xeon X5680 (Westmere) processor has a peak performance of 160 GFLOPS, while an NVIDIA Tesla 2050 (Fermi) GPU has a peak performance of 1180 GFLOPS [38]. Algorithms which are capable of scaling on these resources are desirable now, and will become a requirement with the continuing evolution of computing towards massively parallel architectures.

In the context of incompressible flows, an explicit and local procedure has the potential to use these increasingly parallel computational resources. Since elliptic systems cannot be solved explicitly, one must change the mathematical nature of the incompressibility constraint to remove the requirement of a pressure–Poisson solve. This can be accomplished by considering a pseudocompressible fluid, i.e., a fluid with a small but finite amount of compressibility. One method, artificial compressibility (AC), accomplishes this by prescribing an artificial equation of state relating changes in pressure to small changes in the density. This strategy was taken early in the simulation of fluid flows [10]. It is shown that this artificial equation of state is equivalent to the assumption of the isentropic limit (see section 2), and compressibility effects are $O(\mathrm{Ma}^2)$. In this method, the Mach number becomes an artificial parameter, which may be larger than the physical Mach number, yet small enough to ensure negligible compressibility errors. This method was originally conceived as a means to iterate towards a steady solution, and many authors have noted problems with performing time-accurate solutions unless a dual-time-stepping iterative procedure is used [27, 40, 23, 30], although some evidence exists that with a sufficiently small Mach number, time-accurate solutions are feasible [18, 35]. The lattice-Boltzmann (LB) method has also seen success recently [8, 1]. This method, although based on a discrete version of the Boltzmann equation and kinetic in formulation, ultimately approximates the behavior of a pseudocompressible fluid. Because of this pseudocompressibility, the method is local and amenable to parallel computation.

As originally postulated by Chorin [10], the standard AC method relies on an artificial equation of state. One can show that this artificial equation of state is equivalent to providing an additional thermodynamic constraint to the original compressible Navier–Stokes (CNS) system. This additional constraint is necessary to eliminate the energy or entropy equation from the CNS reducing the number of equations and unknowns by one. The traditional AC method of Chorin is isentropic. While the INS are isentropic in the limit $\mathrm{Ma} \to 0$, compressibility effects remain to $O(\mathrm{Ma}^2)$. Therefore, the artificial method suggested by Chorin is just one potential (and artificial) closure to approximate the INS. While acoustic timescales are generally assumed to be isentropic, at longer viscous and convective time scales, dissipation is important and the isentropic assumption may not be appropriate.

[1] https://asc.llnl.gov/computing_resources/sequoia/

In this report an alternate constraint equation is proposed based on minimizing density fluctuations. By assuming a constant thermodynamic density separate from the low-Mach limiting process, a thermodynamic constraint is derived relating changes in temperature to changes in pressure. As discussed in section 2, this constraint results in markedly reduced pressure fluctuations by introducing a smoothing term into the equation governing the evolution of the pressure. Looking at the entropy balance, one can see that isentropic behavior is not maintained. In effect, the acoustic waves are rapidly damped through entropy generation. This contrasts with the isentropic case, where acoustic wave damping occurs strictly through the coupling of pressure with the momentum equation. The resulting system of equations subject to the above constraint is referred to as the entropically damped artificially compressible (EDAC) Navier–Stokes. Physically, at time scales much larger than t_a, acoustic waves have dissipated viscously, which is not a process that conserves entropy. It is shown in section 2 that the EDAC equations converge to the INS as $\mathrm{Ma} \to 0$.

This research is in part motivated by a series of papers that also proposed an alternate thermodynamic basis for the limit of incompressibility [3, 20, 6], in which the authors noted that an additional smoothing term is beneficial. This method is referred to as the Kinetically Reduced Local Navier–Stokes (KRLNS) method. This research extends the existing KRLNS research in the following manner: The governing equations are not recast using the thermodynamic grand potential since this substitution complicates the derivation. Instead, the derivation proceeds directly from the CNS in terms of the thermodynamic variables of interest. Furthermore, the reduced equations in [6] are missing an important term, which is identified in section 2 and shown to be critical for resolving moderate to high Reynolds number flows. Furthermore, we stress that the smoothing or dissipative term is a direct consequence of an artificially chosen thermodynamic relationship. This relationship is used in the derivation of [3] as an approximation, yet is stated unambiguously here as an artificial constraint, and only one possible constraint that approximates the INS to $O(\mathrm{Ma}^2)$. Thus, the equations derived in both the EDAC and KRLNS methods do not represent the true thermodynamic limit of incompressibility, just one potential limit subject to the chosen constraint.

This paper proceeds by closely examining the CNS equations as $\mathrm{Ma} \to 0$ in section 2, and the effect of the various thermodynamic constraints used to derive an artificial equation of state. Section 3 describes a simple finite-difference discretization of the method and then explores the performance and accuracy of the EDAC method on several model problems. A characteristic analysis of the hyperbolic system associated with the EDAC equations is also presented. Much of the theory development and finite-difference results are available in Clausen (2013) [13]. The theory development is recapitulated and expounded upon where appropriate; however, only pertinent sections of the previously published numerical method and results are repeated. Next, section 4.2 describes a more general finite-element version of the EDAC algorithm and investigates its performance on several model problems. More work is needed to make the finite-element version acceptably robust. Lastly, section 5 details a scalability study using the finite-difference version of the algorithm. This study is performed on the Sequoia supercomputer, with scalable results demonstrated on over one-million cores. A discussion of the relative merits for explicit fluid dynamics solvers is found in section 6.

2 Thermodynamic Limit of Incompressibility

To study the various forms of artificial compressibility and the nature of the Navier–Stokes equations in the limit of incompressibility, we start with the dimensional form of the Navier–Stokes [32], which after using Fourier's heat flux can be written as

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\boldsymbol{u}, \tag{1}$$

$$\rho\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla P + \nabla\cdot\boldsymbol{\sigma}, \tag{2}$$

$$\rho\frac{\mathrm{D}s}{\mathrm{D}t} = \frac{1}{T}\nabla\cdot(k\nabla T) + \frac{1}{T}\boldsymbol{\sigma}:\mathbf{S}, \tag{3}$$

where $\boldsymbol{u}$ is the fluid velocity, ρ is the fluid density, P is the pressure, t is the time, $\boldsymbol{\sigma}$ is the viscous stress tensor, $\mathbf{S}$ is the rate of strain tensor, s is the specific entropy, k is the thermal conductivity, and T is the temperature. Here the entropy balance has been used in lieu of the energy conservation equation, and the convective derivative is defined as

$$\frac{\mathrm{D}(\cdot)}{\mathrm{D}t} \equiv \frac{\partial(\cdot)}{\partial t} + \boldsymbol{u}\cdot\nabla(\cdot).$$

In order to study the behavior of the CNS in the limit of incompressibility, the CNS equations are made dimensionless. Accordingly, the variables of interest are scaled according to

$$\rho^* = \rho/\rho_0 \qquad \boldsymbol{u}^* = \boldsymbol{u}/u_0$$

$$T^* = \frac{T - T_0}{\mathrm{Pr}\, u_0^2/c_p} \qquad s^* = \frac{s}{c_p}$$

$$P^* = \frac{P - P_0}{\rho_0 u_0^2} \qquad t^* = \frac{t u_0}{L}$$

where starred quantities are dimensionless, and ρ_0, u_0, L, P_0, and T_0 are chosen reference quantities. Several dimensionless parameters are of interest: The Prandtl number is defined as $\mathrm{Pr} \equiv \mu c_p/k$, where μ is the fluid shear viscosity, and c_p is the specific heat at constant pressure. The Reynolds number is defined as $\mathrm{Re} \equiv \rho_0 u_0 L/\mu$. The ratio of specific heats is defined as $\gamma \equiv c_p/c_v$, where c_v is the specific heat at constant volume. Note that we have assumed several fluid material properties to remain constant: viscosity, specific heats, and thermal conductivity. Dropping the starred notation hereafter, and assuming all quantities are dimensionless unless otherwise noted, the dimensionless form of the Navier–Stokes can be written as

$$\frac{1}{\rho}\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\nabla\cdot\boldsymbol{u}, \tag{4}$$

$$\rho\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla P + \frac{1}{\mathrm{Re}}\nabla\cdot\boldsymbol{\sigma}, \tag{5}$$

$$\left(\frac{\mathrm{Pr}\gamma\mathrm{Ma}^2}{A}T + 1\right)\rho\frac{\mathrm{D}s}{\mathrm{D}t} = \frac{\mathrm{Ma}^2\gamma}{A}\frac{1}{\mathrm{Re}}\left(\nabla^2 T + \Phi\right), \tag{6}$$

where $\Phi \equiv \boldsymbol{\sigma} : \mathbf{S}$ is the viscous dissipation, $A \equiv \alpha\rho_0 c_p T_0$ is an additional dimensionless parameter, and α is the isothermal compressibility coefficient. The dimensionless forms of the continuity and momentum equations above should be familiar; however, the entropy equation requires some algebra. After applying the previously mentioned scalings and noting that (cf. eq 10.3.7 in [32])

$$\frac{u_0^2}{c_p T_0} = \frac{\mathrm{Ma}^2\gamma}{A}, \tag{7}$$

we can simplify (3) into the form shown in (6). Note that only two thermodynamic variables are independent, e.g., one can express $\rho = \rho(P, s)$ and $T = T(P, s)$ to ensure the system described by (4), (5), and (6) is complete.

Further simplification is possible by expressing the density in terms of its thermodynamic relationship,

$$\mathrm{d}\rho = \mathrm{Ma}^2 \mathrm{d}P - B\left(\frac{\mathrm{Pr}\gamma\mathrm{Ma}^2}{A}T + 1\right)\rho \mathrm{d}s, \tag{8}$$

where $B \equiv \beta T_0$, and β is the thermal expansion coefficient. If we combine the time derivative of (8), the continuity (4), and the entropy (6) equations, the result is

$$-\rho\nabla\cdot\boldsymbol{u} = \mathrm{Ma}^2\frac{\mathrm{D}P}{\mathrm{D}t} - \frac{\mathrm{Ma}^2\gamma}{\mathrm{Re}}\frac{B}{A}\left(\nabla^2 T + \Phi\right). \tag{9}$$

The equations given by (9) and the momentum equation (5) do not represent a complete system since the number of unknowns (two thermodynamic and three momentum) is greater than the number of equations (four). An additional equation, in the form of an artificial thermodynamic constraint, is necessary to complete the system. Equation 9 can also be derived by combining the continuity equation, the thermal energy equation, and the thermodynamic relationship for the density with pressure and temperature as the independent thermodynamic variables; however, the derivation from the entropy form is more convenient.

2.1 Isentropic Limit

If we assume isentropic behavior, $\mathrm{d}s = 0$, thus, (9) becomes

$$\frac{\mathrm{D}P}{\mathrm{D}t} = -\frac{1}{\mathrm{Ma}^2}\rho\nabla\cdot\boldsymbol{u}. \tag{10}$$

To obtain the form used by classical AC methods, which are commonly motivated by assuming the artificial equation of state $\rho = \mathrm{Ma}^2 P$, requires making two approximations. First, the convective derivative can be approximated as $\mathrm{D}P/\mathrm{D}t \approx \partial P/\partial t$ since the acoustic waves travel much faster than the convective velocity. Second, the density perturbations are given as $\rho \approx 1 + O(\mathrm{Ma}^2)$, thus the higher-order terms involving the density can be neglected. The resulting update for the pressure is

$$\frac{\partial P}{\partial t} = -\frac{1}{\mathrm{Ma}^2}\nabla\cdot\boldsymbol{u}, \tag{11}$$

which is equivalent to the classical AC methods [10]. While entropy balance (6) shows isentropic behavior to $O(\mathrm{Ma}^2)$, (9) clearly shows that both the pressure and entropic contributions contain terms of the same order. This behavior suggests that an isentropic assumption, while certainly valid to $O(Ma^2)$, may not be the only acceptable choice for a pressure update procedure.

2.2 Minimization of Density Fluctuations

As an alternative constraint, consider imposing no thermodynamic fluctuations in the density, i.e., $\mathrm{d}\rho = 0$. It is essential to realize that this is an additional constraint imposed *separately* from the low-Mach limiting process. To understand the consequences of imposing $\mathrm{d}\rho$, consider the thermodynamic function for the density as function of the pressure and temperature,

$$\frac{\mathrm{d}\rho}{\rho} = \gamma \mathrm{Ma}^2 \left(\mathrm{d}P - \frac{\mathrm{Pr}B}{A} \mathrm{d}T \right). \tag{12}$$

If $\mathrm{d}\rho = 0$, the term in parentheses is also zero; consequently, a thermodynamic relationship of the form

$$\mathrm{d}P = \frac{\mathrm{Pr}B}{A} \mathrm{d}T, \tag{13}$$

exists, which can be used to eliminate the temperature diffusion term in (9). We note that this relationship between pressure and temperature fluctuations was used by [3]; however, their motivation hinged on the right-hand side of (12) being close to zero from the low-Mach limiting procedure. It is important to realize that while the density naturally becomes constant (in a Lagrangian sense) in the low-Mach-number limit for a given material region, (13) represents an additional constraint that is equivalent to a constant thermodynamic density. Without this additional constraint, (12) states that $O(\mathrm{Ma}^2)$ changes in density cause $O(1)$ changes in the dimensionless pressure and temperature and does not allow the simplification stated in (13). This constraint is entirely artificial, but so is the isentropic constraint, and as is shown in section 3.3, this constraint yields several desirable properties.

Simplifying (9) using (13) gives

$$\frac{\mathrm{D}P}{\mathrm{D}t} = -\frac{1}{\mathrm{Ma}^2} \rho \nabla \cdot \boldsymbol{u} + \frac{\gamma}{\mathrm{PrRe}} \nabla^2 P + \frac{B}{A} \frac{\gamma}{\mathrm{Re}} \Phi. \tag{14}$$

Several simplifying assumptions are made for the remainder of the investigation: First, the density is eliminated just as in the isentropic case. Second, we assume $\mathrm{Pr} = \gamma$, and lastly, the effects of viscous dissipation are neglected. These simplifications are possible through the appropriate selection of the dimensionless parameters, although certainly not required. The resulting simplified version is shown as

$$\frac{\mathrm{D}P}{\mathrm{D}t} = -\frac{1}{\mathrm{Ma}^2} \nabla \cdot \boldsymbol{u} + \frac{1}{\mathrm{Re}} \nabla^2 P, \tag{15}$$

which, when combined with the momentum equation (5) forms a complete system for the solution of pseudo-compressible flow that is referred to as the EDAC method. Physically, pressure (acoustic) wave propagation in the EDAC method is damped via an entropy generating mechanism created via an artificial thermodynamic constraint that relates temperature and pressure fluctuations.

2.3 Discussion

It is instructive to consider the behavior of both the AC and EDAC methods in the incompressible limit, $\mathrm{Ma} \to 0$. In this case, both methods converge to the INS with $O(\mathrm{Ma}^2)$ errors, and both methods are isentropic (as are the INS). Also, as the viscous terms become less important, i.e., $\mathrm{Re} \to \infty$, both methods demonstrate isentropic behavior. The primary differences occur in the viscously dominated regime, particularly as $\mathrm{Re} \to 0$. In this case the EDAC method converges to the pressure–Laplacian equation that is valid for Stokes flow,

$$\nabla^2 P = 0, \tag{16}$$

which also enforces incompressibility. Conversely, the AC method is unaffected by the choice of Reynolds number. The ability of the EDAC to demonstrate proper behavior at both the viscous and inertial limits is the key to developing a time-accurate AC method.

The motivation for this research is the existing KRLNS; therefore, it is worth comparing the newly developed EDAC method with the existing KRLNS [3, 20, 6], method. The KRLNS method is based on a substitution,

$$\mathcal{G} = P - e_k, \tag{17}$$

where $\mathcal{G}$ is stated to be the negative of the grand potential, and $e_k \equiv \rho \boldsymbol{u} \cdot \boldsymbol{u}/2$ is the kinetic energy. After simplifying the CNS using the above substitution, an update procedure is obtained for $\mathcal{G}$ as [6]

$$\frac{\partial \mathcal{G}}{\partial t} = -\frac{1}{\mathrm{Ma}^2} \nabla \cdot \boldsymbol{u} + \frac{1}{\mathrm{Re}} \nabla^2 \mathcal{G}. \tag{18}$$

Taking the EDAC equation (15) and subtracting (18), reveals a difference, shown as

$$\mathrm{EDAC} - \mathrm{KRLNS} = \nabla \cdot (\boldsymbol{u} e_k) + \frac{1}{\mathrm{Re}} \nabla \boldsymbol{u} : \nabla \boldsymbol{u}. \tag{19}$$

The first term on the right-hand side is associated with the convection of kinetic energy and is likely negligible based on the difference in acoustic and convective speeds. The second term is shown to be important in obtaining a qualitatively correct pressure evolution in section 3.3. We note that these terms are present in earlier formulations of the KRLNS [3], but they are missing in the reduced set of equations [6].

In all derivations of the KRLNS [3, 20, 6], a key simplification relates fluctuations in temperature to fluctuations in the grand potential, which is based on the argument that at long time scales the density is approximately constant. This assumption requires the thermodynamic constraint shown in (13), which is not explicitly stated in the derivation of the KRLNS equations. In [3], the authors make an argument that the precursor equations to the KRLNS represent the minimal thermodynamic system to represent incompressible flows. We suggest that a more appropriate statement would be that the KRLNS precursor (and the EDAC equations) represent only one potential limit of incompressibility subject to the previously mentioned thermodynamic constraint.

3 Finite Difference Implementation

3.1 Numerical Method

The finite-difference implementation of the EDAC method (FD–EDAC) is detailed fully in Clausen (2013) [13] and a brief overview is provided here. The discretization method is a simple MacCormack predictor–corrector scheme [26]. The MacCormack scheme has a nominal accuracy of $O(\Delta x^2, \Delta t^2)$. Restating the governing equations in conservative form for a two-dimensional system gives

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{G}(\boldsymbol{U})}{\partial y} = \boldsymbol{S}(\boldsymbol{U}), \tag{20}$$

where $\boldsymbol{U} \equiv (P, \boldsymbol{u})$ is a vector of the solution variables. The functions $\boldsymbol{F}$ and $\boldsymbol{G}$ capture advection terms, and $\boldsymbol{S}$ contains the source terms, which are discretized using central differences. This form is readily amenable to the application of the MacCormack method. We alternate between a forward–back integration and a back–forward step to reduce errors [42]. The stability limit of the MacCormack scheme is given in dimensionless form for two dimensions as [42]

$$\Delta t_{CFL} \leq \sigma_{sf} \left[\frac{1}{\Delta x} + \frac{1}{\Delta y} + \frac{1}{\mathrm{Ma}} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \right]^{-1}, \tag{21}$$

where σ_{sf} is a safety factor and is chosen to be $1/2$ in this work. An additional stability limit exists for the diffusion term such that

$$\Delta t_{diff} \leq \frac{\sigma_{sf}\mathrm{Re}}{2} \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right]^{-1}. \tag{22}$$

Stability is ensured by selecting a time step such that $\Delta t = \min(\Delta t_{CFL}, \Delta t_{diff})$.

Code verification is performed by the method of manufactured solutions (MMS) [39]. The solution form is chosen as a traveling wave problem [29], which has an exact solution for the INS; however, the AC, KRLNS and EDAC methods only approximate the INS equations with an error of $O(\mathrm{Ma}^2)$. Thus, the manufactured solution accounts for this $O(\mathrm{Ma}^2)$ error, and the errors reported in [13] are restricted to discretization errors. A grid-refinement study is conducted and the method demonstrated the expected order of convergence (quadratic).

For comparison purposes, a single-relaxation-time LB method is also used. Although a kinetic method based on a discretization of the Boltzmann equation, the LB method approximates the INS equations in the long time limit using a form of pseudocompressibility. The LB method has been long-known to behave similarly to an AC method [18] with an equation of state $P = c_s^2 \rho$. The LB method has been quite successful in simulating a variety of flows including turbulent, suspension, and multiphase flows. The locality of information propagation, i.e., pseudo-compressibility, of the method has been key to its scalability, with simulations scaling to $O(100,000)$ processors [14]. The reader is referred to references [8, 1] for more information on the LB method. It is important to realize that the LB method is explicit, and as such has time step restrictions just as in AC-based methods. Time step sizes are limited by the need to maintain $\mathrm{Ma} \ll 1$, as well as keeping the relaxation time from being overdamped, which results in a rapid loss of accuracy [15].

3.2 Characteristic Analysis

The nature of the EDAC equations are now hyperbolic instead of elliptic and a characteristic analysis can be performed. The characteristic analysis provides a means to describe wave propagation in terms of characterstic variables and to specify characteristic boundary conditions. These boundary conditions rely on recasting a one-dimensional form of the EDAC equations into characteristic variables, then handling open boundary conditions such that incoming characteristic waves are consistent with the physics. Also possible are nonreflecting forms, in which outgoing characteristics waves are allowed to escape the domain such that they are not reflecting. Doing so decreases the time necessary for artificial acoustic waves to damp out of the domain, particularly at open-flow boundaries.

Following standard hyperbolic theory [25], the EDAC equations represent the propagation of waves throughout the domain. The two-dimensional EDAC equations can be written in matrix form as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \mathbf{J}_F \frac{\partial \boldsymbol{U}}{\partial x} + \mathbf{J}_G \frac{\partial \boldsymbol{U}}{\partial y} = \text{Source Terms}, \tag{23}$$

where

$$\boldsymbol{U} = \begin{pmatrix} u \\ v \\ P/\rho \end{pmatrix} \tag{24}$$

is the solution vector and the flux Jacobian matrices are defined as

$$\mathbf{J}_F = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \begin{bmatrix} u & 0 & 1 \\ 0 & u & 0 \\ c^2 & 0 & u \end{bmatrix} \qquad \mathbf{J}_G = \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} = \begin{bmatrix} v & 0 & 0 \\ 0 & v & 1 \\ 0 & c_s^2 & v \end{bmatrix}. \tag{25}$$

For $\mathbf{J}_F$ this gives the eigenvalues of $\{u - c_s, u, u + c\}$, which correspond to three characteristic waves: two associated with the propagation of acoustic waves in either direction, and one associated with bulk convective motion. The associated left eigenvectors are written as

$$\boldsymbol{\Lambda}_x = \begin{bmatrix} -c & 0 & 1 \\ 0 & 1 & 0 \\ c_s & 0 & 1 \end{bmatrix}. \tag{26}$$

It is possible to write the governing equations in characteristic form by defining $\mathcal{L}_i \equiv \lambda_i \boldsymbol{\Lambda}_i \boldsymbol{U}_x$, substituting into (23), and simplifying[43, 36]. The locally one-dimensional system is written as

$$\frac{\partial u}{\partial t} + \frac{1}{2c}\left[\mathcal{L}_3 - \mathcal{L}_1\right] = 0 \tag{27}$$

$$\frac{\partial v}{\partial t} + \mathcal{L}_2 = 0 \tag{28}$$

$$\frac{\partial P}{\partial t} + \frac{\rho}{2}\left[\mathcal{L}_1 + \mathcal{L}_3\right] = 0. \tag{29}$$

From these relationships it is possible to enforce boundary conditions in a consistent manner. For an inlet boundary condition, shown in figure 1(a), in which the velocity at the inlet is specified,

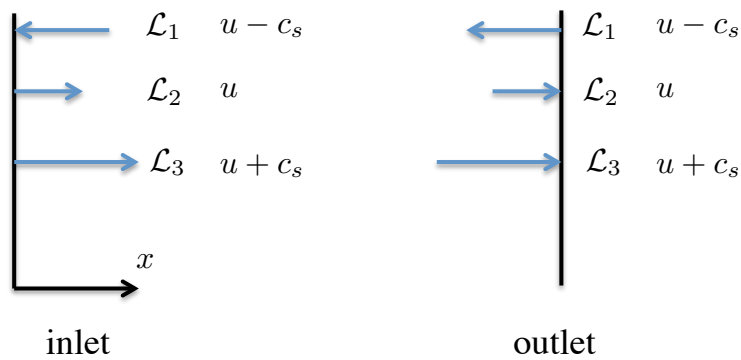

Figure 1. Schematic of characteristic waves entering and exiting the domain at flow inlet and outlet. The wave propagation speeds are also shown.

two characteristic waves propagate into the domain, while one exits. It is easy to verify from (27–29) that $\mathcal{L}_1 = \mathcal{L}_3$ and $\mathcal{L}_2 = 0$, assuming u and v are not functions of time. These relationships provide a way to specify the pressure boundary condition in order to maintain consistency, where the time derivative in the pressure is related to outgoing characteristic $\mathcal{L}_1$. Since $\mathcal{L}_1$ is outgoing, i.e., no information associated with this characteristic is entering the domain, and all derivatives can be approximated numerically via extrapolation. For an outflow, only one characteristic is entering the domain ($\mathcal{L}_1$), as shown in figure 1(b). In a typical instance, this characteristic is associated with specifying a back pressure. In this case, $\mathcal{L}_2$ and $\mathcal{L}_3$ are calculated via extrapolation, and $\mathcal{L}_1$ is set via the pressure boundary condition, usually through a penalty mechanism[36].

The characteristic boundary conditions can be generalized to a three-dimensional inlet normal; however, the relationships are only approximations assuming locally one-dimensional inviscid behavior. Additional boundary conditions may be necessary for the higher order derivatives (Neumann conditions, see [36]). For nonreflecting boundary conditions, the characteristic wave strength entering the domain is set to zero, i.e., $\mathcal{L}_{\text{entering}} = 0$.

3.3 Results

Using the FD–EDAC method detailed above, this section details the ability of the EDAC equations to model the INS equations in both steady and transient flows. Many of the results in this section are available in detail in Clausen (2013) [13], and only summaries of the results are given. These results include those for lid-driven cavity flow, a convecting vortex, and a doubly periodic shear layer. In some cases, comparisons are made with the KRLNS method, the traditional AC method, and the LB method. New results for the FD–EDAC include decaying three-dimensional turbulence. The results show that the EDAC method is capable of accurate simulation of INS flows even at relatively high Mach numbers. The KRLNS method is qualitatively incorrect, and the traditional AC method is subject to pressure/divergence fluctuations, which highlight the need for the

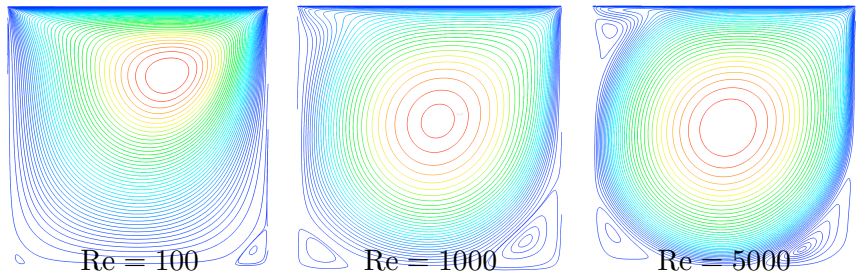

Figure 2. Streamlines from the EDAC simulations at Ma = 0.1 showing the formation of three vortices: the primary, bottom left, and bottom right. At Re = 5000 an additional vortex forms along the top left of the domain, as is expected. Warmer colors indicate higher stream function values. Figure from Clausen (2013) [13].

smoothing term present in the EDAC equations.

Lid-driven Cavity Flow

Lid-driven cavity flow is a canonical flow in the evaluation of incompressible flow solvers due to the complex flow patterns exhibited. Simulations are performed using three methods (AC, EDAC, and KRLNS) at Re=100, 1000, and 5,000. For the AC method, it is trivial to see that at steady state (11) is independent of the Mach number and identical to the INS result; however, for the EDAC and KRLNS methods an error still remains. Thus, Ma=0.1 and 0.3 simulations are performed. All simulations are performed on a 256×256 mesh. The velocity profiles are compared with high-resolution stream–vorticity formulations [17, 16]. Unlike traditional INS solvers, the pressure evolution equation requires an additional boundary condition. Following [6], for the AC, EDAC, and KRLNS simulations a Neumann condition for the pressure is used enforcing $\partial P/\partial n = 0$, where n is the wall-normal distance. The boundary condition is implemented via a second-order one-sided finite difference. Simulations are considered steady state when the L^2-norms of the unsteady terms are less than 1.0×10^{-8}. For a more thorough description of the results, including data at all Reynolds numbers as well as vortex locations, see [13]. Here, only salient features will be discussed.

First, figure 2 shows streamlines at various Reynolds numbers, and the formation of primary, secondary, and (in high-Re cases) tertiary vortices are shown. Figure 3 shows mid-plane velocity profiles for the Re = 5000 case. The trends seen at the other Reynolds numbers are also present at Re = 5000. Excellent agreement is seen with the AC and EDAC methods at all Mach numbers simulated; however, the KRLNS method shows deviations from the expected results. The error term in (19) scales $1/\mathrm{Re}$; however, larger velocity gradients are present at higher Reynolds number, which creates larger errors. We note that the apparent error in the KRLNS method is not unique to simulations we performed, but is also present in the results shown in [6].

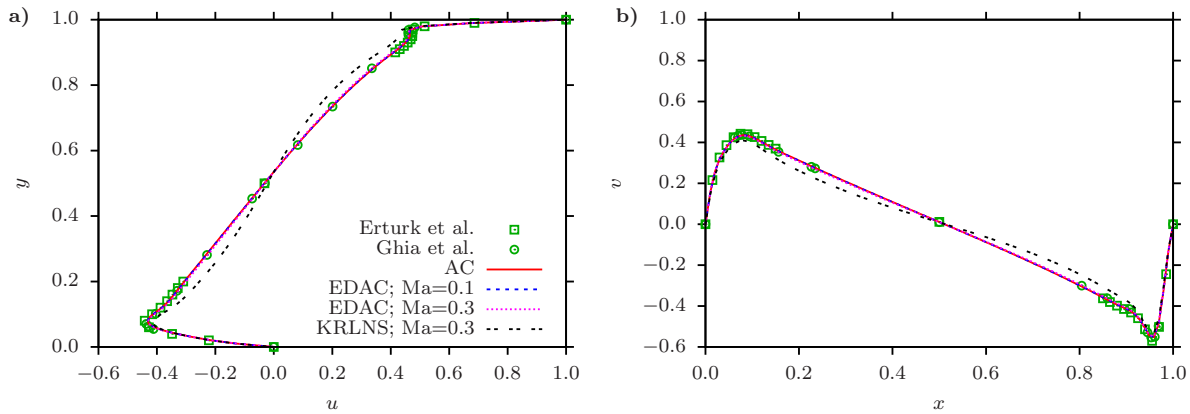

Figure 3. LDCF results for Re = 5000 showing excellent agreement for the (a) u and (b) v velocity components Figure from Clausen (2013) [13].

Traveling Wave

Another problem examined in depth in Clausen (2013) [13] is a convected vortex, which was also used as the basis for the code verification manufactured solution. The exact INS solution for this flow is given by

$$u(x,y,t) = \frac{1}{3} + \frac{2}{3}\left\{\cos\left[2\pi\left(x-\frac{t}{3}\right)\right]\sin\left[2\pi\left(y-\frac{t}{3}\right)\right]\exp\left(\frac{-8\pi^2 t}{\mathrm{Re}}\right)\right\} \tag{30}$$

$$v(x,y,t) = \frac{1}{3} - \frac{2}{3}\left\{\sin\left[2\pi\left(x-\frac{t}{3}\right)\right]\cos\left[2\pi\left(y-\frac{t}{3}\right)\right]\exp\left(\frac{-8\pi^2 t}{\mathrm{Re}}\right)\right\} \tag{31}$$

$$P(x,y,t) = -\frac{1}{9}\left\{\cos\left[4\pi\left(x-\frac{t}{3}\right)\right] + \cos\left[4\pi\left(y-\frac{t}{3}\right)\right]\right\}\exp\left(\frac{-16\pi^2 t}{\mathrm{Re}}\right) \tag{32}$$

This equation is an exact solution to the INS for a convecting, decaying Taylor–Green vortex; however, the above is not an exact solution to either the AC or EDAC methods, thus an initial perturbation is introduced that must decay, and the smoothing term that appears in (15) is crucial to recover time-accurate dynamics at moderate values of the Mach number. Traditional AC schemes, which do not contain the smoothing term, require extremely small Mach numbers (and hence time steps size) in order to keep the pressure fluctuations from polluting the solution domain. The small Mach-number requirement is especially prevalent at lower Reynolds numbers.

Unlike the LDCF problem, where the KRLNS produced a physically realizable solution, albeit with some error, the KRLNS result fails to produce a qualitatively correct solution for the traveling wave problem. To show this error, simulations are performed on a 65×65 domain at Re = 1 with periodic boundary conditions and $\Delta t = 2.96 \times 10^{-5}$. Simulations are performed using the EDAC method, the KRLNS method, and the KRLNS method with the error term identified in (19) added back to the solution. In figure 4(a), the implementation of the reduced KRLNS equations

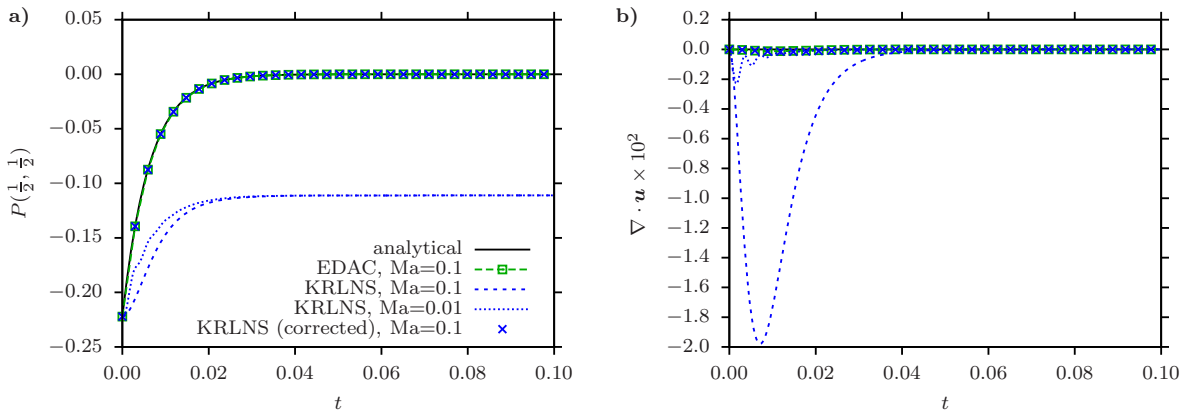

Figure 4. Traveling wave solution showing the pressure (a) and velocity divergence (b) at location $(1/2, 1/2)$ at Re = 1. KRLNS method error creates qualitatively incorrect behavior unless correction term is included. Figure from Clausen (2013) [13].

listed in [6] results in a pressure decay that is much too rapid, and an incorrect steady pressure after the complete decay of the vortex. This pressure error does not decay as the Mach number is refined, although the divergence error shown in figure 4(b) does show some improvement. The pressure results obtained using the corrected form, i.e., with the error term in (19) added back into the solution, agree well with both analytic and EDAC results. The divergence is also closer to the expected incompressible value of zero. The errors in the pressure and velocity divergence are much more pronounced than the velocity component errors, which are not shown for these cases. Since the KRLNS method yields qualitatively incorrect behavior without the correction terms, the focus throughout the remaining portion of this manuscript will be on characterizing the EDAC method with respect to AC and LB methods.

The previous experiment is repeated using the AC, LB, and EDAC methods, and the results are shown in figure 5. Since no LB results were found in the literature for this test case, we used a simple single-relaxation-time, two-dimensional, nine-direction LB model for comparisons, with the relaxation time denoted τ_{LB}. Further details involving the theory and implementation of the LB method are outside the scope of this paper and can be found in [1]. The same number of grid points and time step size are used as the above problem, with the exception of the LB method, where $\Delta t = 8.88\times10^{-5}$ and $\tau_{LB} = 1.62$. Oscillations are seen in both the traditional AC method and the LB method, although the pressure oscillations are much less with the LB method. At larger Mach numbers (Ma = 0.1), the AC method becomes meaningless, without even a qualitative resemblance to the physical solution. Note that higher Mach-number LB solutions are not shown because increasing the Mach number at a fixed Reynolds number requires increasing τ_{LB}, which causes a degradation in the solution accuracy particularly for the single-relaxation time model used here; thus, LB simulations are limited to Ma = 0.01. Essentially no pressure oscillations are seen for the EDAC method, and the divergence errors are much smaller than other methods.

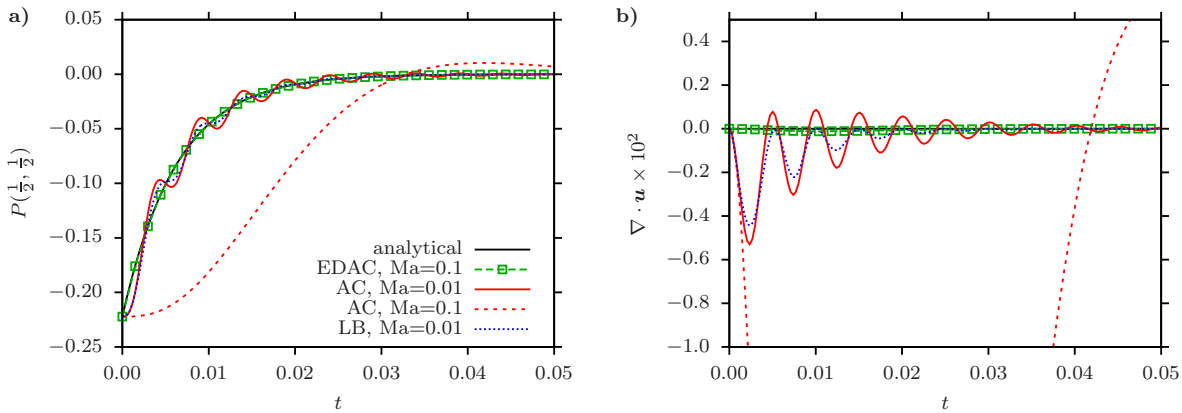

Figure 5. Traveling wave solution showing the pressure (a) and velocity divergence (b) at location $(1/2, 1/2)$ at Re = 1. Pressure and velocity divergence errors exist for the AC and LB methods. The EDAC method rapidly damps the fluctuations showing greatly reduced errors. Figure from Clausen (2013) [13].

The impressive increase in accuracy of the EDAC method is somewhat tempered by the explicit nature of the algorithm. At Re = 1 the simulation time step is diffusion limited for cases where the Mach number increases much beyond 0.01; however, mixed implicit–explicit algorithms are a possibility. Perhaps more importantly, the poor behavior of the AC method at Ma = 0.1, which should still approximate incompressible flow, reveals that the underlying physics associated with the AC method are not generally applicable for transient simulations at all Reynolds numbers. These restrictions do not exist with the EDAC method. Another complicating factor that deserves mention is the possibility of pressure-mode coupling created by using collocated (non-staggered) pressure and velocity variables. Previous research using the AC method on staggered grids found similar pressure fluctuations to the LB method [18], although some investigators have used AC on non-staggered meshes [35]. The ability to use the EDAC method on meshes with collocated pressure and velocity variables reduces code complexity. At higher Reynolds numbers, pressure smoothing term becomes less dominant; however, retaining the small amount of smoothing is necessary to maintain oscillation free solutions. For details, see [13].

Doubly Periodic Shear Layer

In Clausen (2013) [13], a doubly periodic shear layer is studied in depth. This problem uses a periodic domain with velocity initial condition that creates two horizontal shear layers of a prescribed width. A vertical velocity perturbation is also prescribed. This system was first investigated by [4] and was later studied in depth by [29]. For resolved cases, the shear layers roll-up into a pair of symmetric vortices; however, for under-resolved mesh resolutions, spurious vortices appear.

The global energy and enstrophy decays are compared between the FD–EDAC and several

other known numerical results (cf. figures 10 and 11 from [13]). For the coarsely meshed cases, spurious vortex creation is seen for all methods, including AC and FD–EDAC. The simulations are performed at Re = 10,000, thus the amount of dissipation introduced by the pressure-smoothing term in the EDAC method is minimal. At finer mesh resolutions, the EDAC and AC methods agree well with the reference pseudo-spectral method. In general, enstrophy decay is insensitive to Mach number; energy decay shows more oscillations and is unphysically high at large Mach numbers, e.g. Ma = 0.2. This model problem will be revisited in the context of a finite-element solver for the EDAC equations in section 4.2.

Freely Decaying Turbulence

The last model problem investigated with the FD–EDAC method is that of freely decaying turbulence. As turbulence decays, energy from the flow is channeled into ever smaller vortical structures, until the flow is viscously dissipated. While the macro-scale Reynolds number is large, and thus viscous dissipation is minimal, at the length scales of the smallest structures, viscous dissipation dominates. In general, higher Reynolds number flows have smaller length and time scales of the smallest features of turbulence. These scales are known as the Kolmogorov scales [37],

$$\eta \equiv \left(\frac{\nu^3}{\epsilon}\right)^{1/4}, \tag{33}$$

$$\tau \equiv \left(\frac{\nu}{\epsilon}\right)^{1/2}, \tag{34}$$

where ν is the kinematic viscosity and ϵ is the kinetic energy dissipation. The Kolmogorov scales describe the smallest scales that must be resolved for fully resolved turbulence. The ratio of Kolmogorov to macroscopic scales goes as $\eta/L \sim \mathrm{Re}^{-3/4}$ and $\tau/t \sim \mathrm{Re}^{-1/2}$. Thus, a simulation at Re = 2000 in a unit domain has a Kolmogorov length scale around $1/300$.

In this case, an initial velocity profile from [22], shown as

$$u(x, y, z) = \sin x\,(\cos 3y \cos z - \cos y \cos 3z)\,, \tag{35}$$

$$v(x, y, z) = \sin y\,(\cos 3z \cos x - \cos z \cos 3x)\,, \tag{36}$$

$$w(x, y, z) = \sin z\,(\cos 3x \cos y - \cos x \cos 3y)\,, \tag{37}$$

is prescribed. This initial profile has been studied extensively in the literature using spectral [22, 5] and LB methods [21, 9]. The initial conditions are periodic, but they also contain many forms of symmetry [22]. This symmetry is useful for spectral methods, and results in a considerable reduction of computation. Non-spectral methods, including the FE and FD EDAC methods, gain no advantage to the symmetries present. The enstrophy and entropy decay are calculated via

$$\Omega = \sum_{i=1}^{N} \frac{\omega_i^2 V_i}{2V}, \tag{38}$$

$$E = \sum_{i=1}^{N} \frac{(\boldsymbol{u} \cdot \boldsymbol{u}) V_i}{2V}, \tag{39}$$

respectively, where the summation is over the number of nodes N, V_i is the cell volume, and V is the overall domain volume. The vorticity is calculated using a 2nd-order central difference.

An EDAC simulation is performed on a $(2\pi)^3$ domain at $\mathrm{Re} = 1/\nu = 2000$ on a 512^3 mesh, thus the simulation is fully resolved. The Mach number is 0.02, and the time step is 5.0×10^{-6}, thus the temporal scales are fully resolved. Figure 6 shows the isosurfaces of vorticity for several time planes. Vortex stretching and breakup is clearly evident as time advances. Figure 7 shows the energy and entropy decay as a function of time. The results from [22] are also shown for comparisons. The results agree fairly well, although the EDAC method predicts slightly lower enstrophy values past the initial increase in entropy. At long times, enstrophy values decay a slightly slower rate than the results in [22]. More investigation is needed to determine if this effect is significant or related to the pressure-smoothing term.

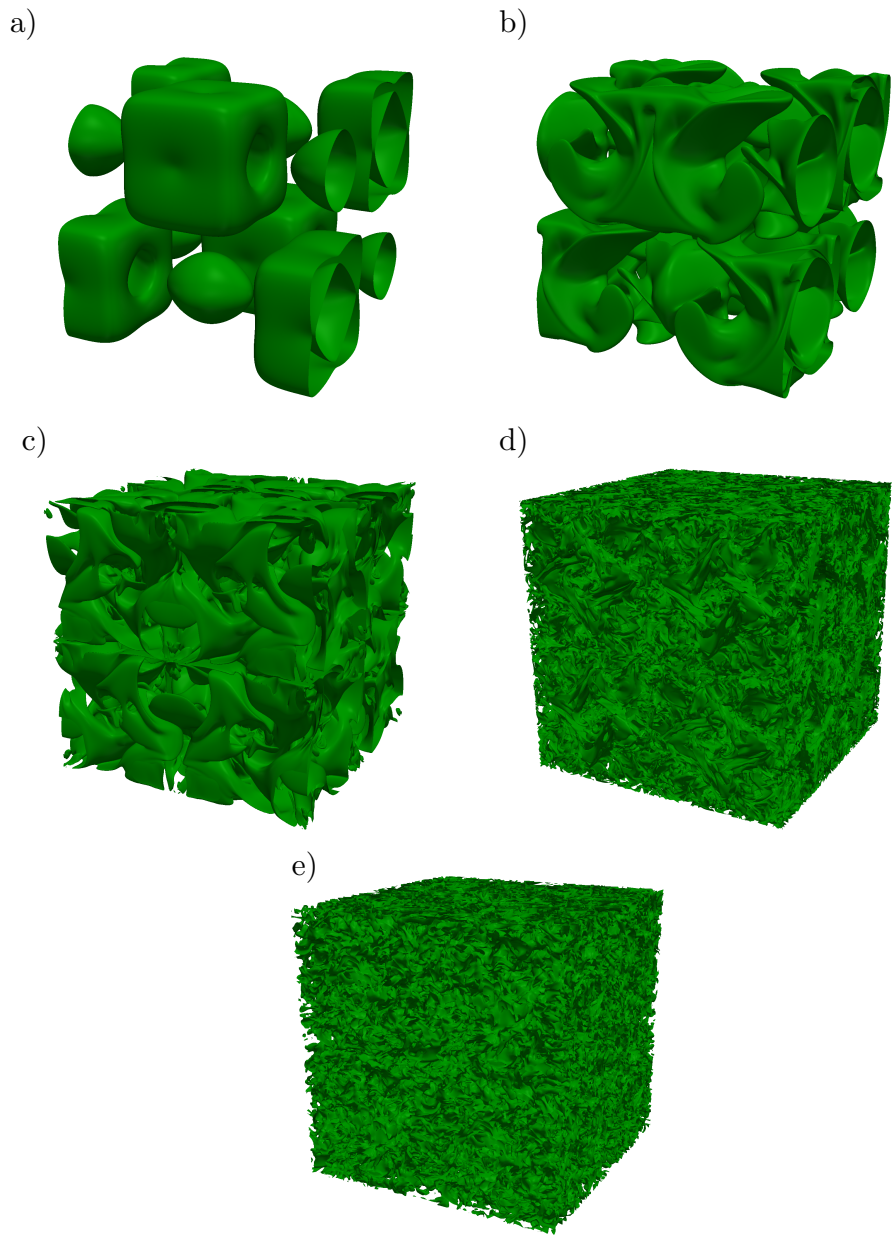

Figure 6. Images of the $\omega_z = 0.8$ vorticity contours at times (a) $t = 0$, (b) $t = 1$, (c) $t = 2$, (d) $t = 4$, and (e) $t = 8$. Maximum enstrophy occurs near (d).

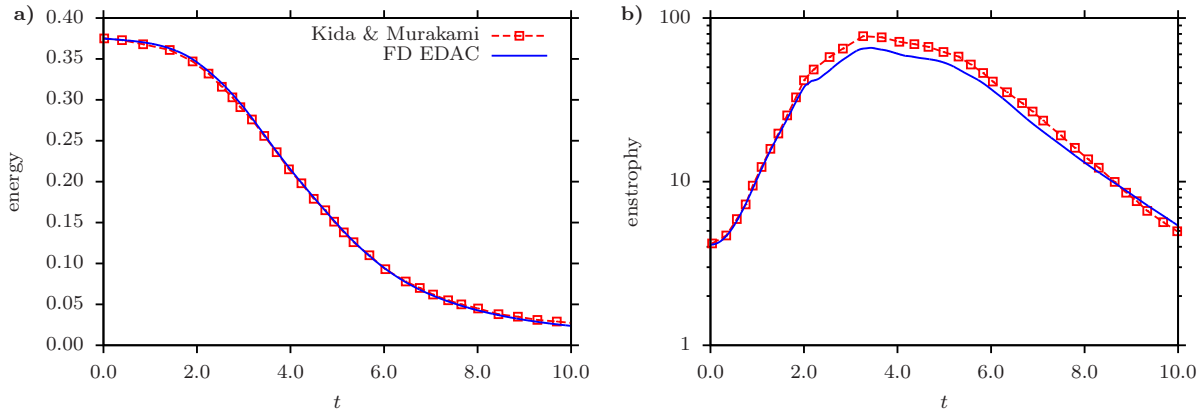

Figure 7. Freely decaying turbulence showing the (a) energy and (b) enstrophy decay.

4 Finite Element Implementation

While the finite difference implementation serves as a useful proof-of-concept for the EDAC method, a more general fluid simulation capability is desired. Accordingly, a finite element (FE) implementation of the EDAC method is implemented in the Sierra-TF codebase. Such an implementation provides an unstructured grid capability, and also provides access to a wealth of physical models that are already implemented. A successful version of the FE–EDAC method is demonstrated and some preliminary results are reported in the following sections; however, two unresolved obstacles were discovered. First, the FE–EDAC method demonstrates some robustness issues, and time steps restrictions appear to be more severe than the finite-difference version. It is suspected that this lack of robustness is related to interaction with pressure stabilization that is required for FE fluid dynamics (both explicit and traditional implicit solvers). Second, explicit algorithms require very efficient assembly routines since the time step restrictions inherent to these methods require many more steps and nearly all of the time is spent in assembly since there is no associated linear system solution.

4.1 Numerical Method

The FE–EDAC method uses Q1–Q1 FE basis functions for approximating the velocity and pressure solutions. As is typical for incompressible FE solvers, convection-dominated flows require additional stabilization, and a variant of SUPG (streamline-upwind/Petrov Galerkin) is used. For incompressible flows, a purely Galerkin formulation using equal-order basis functions results in a scheme that does not satisfy the Babuska–Brezzi inf-sup condition, and some form of pressure stabilization is necessary. The EDAC method is not strictly incompressible, and the pseudocompressibility is sufficient to result in a stable scheme for many situations; however, it is possible to create pressure locking in cases where the grid resolution of spatial gradients is insufficient; consequently, additional pressure stabilization is needed. The momentum (5) and pressure evolution (15) are written in dimensional form as

$$\rho\left\{\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right\} = -\nabla P + \nabla\cdot\boldsymbol{\sigma}, \tag{40}$$

$$\frac{\partial P}{\partial t} + \boldsymbol{u}\cdot\nabla P = -\rho c_s^2\nabla\cdot\boldsymbol{u} + \frac{\mu}{\rho}\nabla^2 P, \tag{41}$$

where $\boldsymbol{\sigma} = -\nabla P\boldsymbol{\delta} + 2\mu\mathbf{E}(\boldsymbol{u})$ and $\mathbf{E}(\boldsymbol{u})$ is the rate of strain tensor. Assuming the choice of suitable velocity and pressure spaces $\boldsymbol{w}$ and q, respectively, the weak form of the solution can be written as

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{w}\right) - (P, \nabla\cdot\boldsymbol{w}) + (2\mu\mathbf{E}(\boldsymbol{u}) : \mathbf{E}(\boldsymbol{w})) + \mathrm{STAB} = 0, \tag{42}$$

$$\left(\frac{\partial P}{\partial t} + \rho c_s^2\nabla\cdot\boldsymbol{u}, q\right) + \left(\frac{\mu}{\rho}\nabla P, \nabla q\right) + \mathrm{STAB} = 0. \tag{43}$$

The notation $(\phi, \psi) = \int \phi\psi \,\mathrm{d}\Omega$ denotes the scalar product in L^2 over the domain Ω, and terms STAB denote additional stabilization terms which will be explored during the study of several

model problems below. All time integration is performed using a first-order accurate backward Euler. Typically first order time integration methods are the most diffusive, and thus, the most stable. Higher-order approaches are left to future investigations.

Stabilization

SUPG stabilization is commonly used to stabilize convection dominated flows. Given a partition $\mathcal{T}_h$ of domain Ω into K elements, SUPG stabilization can be written as

$$\mathrm{STAB}_{SUPG} = \sum_{K\in\mathcal{T}_h} \left(\tau_{SUPG}\,\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}, \boldsymbol{w} \right) + \left(\nabla P, \boldsymbol{w}\right), \rho\boldsymbol{u}\cdot\nabla\boldsymbol{w} \right)_K . \tag{44}$$

Detailed discussions regarding the formulation of τ_{SUPG} are found elsewhere in literature [41]. For this work, the form of [41] without unsteady terms is used, which is given as

$$\tau_{SUPG} = \frac{1}{\sqrt{\boldsymbol{u}\cdot\mathbf{g}\cdot\boldsymbol{u} + 12\nu^2\mathbf{g}:\mathbf{g}}}, \tag{45}$$

where $\mathbf{g}$ is the covariant metric tensor for K.

For explicit simulations, the unsteady term of (44) is not known *a priori* and is ignored. Furthermore, including the pressure term, while beneficial in the fully implicit INS case, provides a destabilizing influence on the solution, so this term is also ignored. A full explanation for this destabilizing effect is not known, although it is suggested that the "convection" of the pressure wave does not have to occur in the direction of bulk fluid flow, which is a phenomena that does not exist in the incompressible case.

The pressure is stabilized by augmenting the pressure smoothing term inherent to the EDAC method for coarse grids. Thus,

$$\mathrm{STAB}_P = \sum_{K\in\mathcal{T}_h} \left(\tau_P \rho c_s^2 \nabla P, \nabla q \right), \tag{46}$$

where $\tau_p = \tau_{SUPG}$.

4.2 Results

Several test problems are considered for the FE–EDAC algorithm. The LDCF and convecting vortex problems are repeated using the new methodology, and quantitative assessments are taken. A qualitative look at vortex shedding from flow around a square cylinder is also examined. Lastly, the case of the doubly periodic shear layer is examined, which highlights some issues in the FE implementation of this algorithm. All results unless otherwise stated include the enhanced pressure stabilization term.

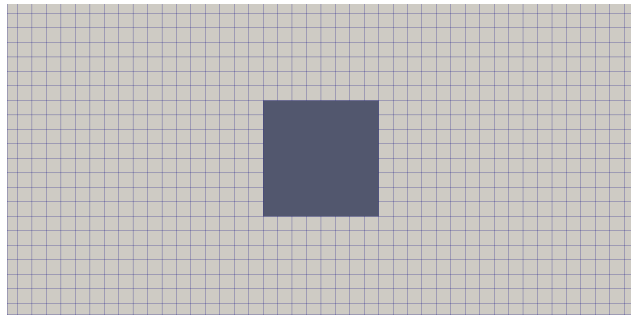

Figure 8. Zoom near object showing mesh used for simulating flow over square cylinder.

von Kármán Vortex Street

The formation of a von Kármán vortex street commonly occurs during flow past blunt objects at moderate Reynolds numbers. This type of flow is distinguished by alternating vortices that shed in the downstream wake of the object in a periodic fashion. For ease of meshing, a square obstruction was chosen, and nonreflecting boundary conditions are used for the side walls and outlet. The simulation is performed at Re = 100 and Ma = 0.1. Figure 8 shows the mesh near the object used in this study, which is quite coarse.

First, the flow is simulated with the FE–EDAC method using only SUPG stabilization, and figure 9(a) shows the pressure-locking phenomena associated with flow around a square object. Such locking is not unexpected in this case if one compares the relative magnitude of τ_P compared with that which would be used in the strictly incompressible case. For this simulation $\rho c_s^2 \tau_P \approx 12.0$, while the EDAC pressure smoothing term $\mu/\rho \sim 0.1$. Adding the τ_P based stabilization results in smooth pressure field, also shown in figure 9(b). The addition of stabilization can affect the diffusive time step limitation since the term is diffusive in nature. In this case, the time step is convection limited, thus adding stabilization does not adversely affect the time step.

Images of the velocity fields and vorticity are shown in figure 10. The frequency of oscillations are described by the Strouhal number, $\mathrm{St} \equiv fL/u_\infty$, where f is the frequency, L is the square side length, and u_∞ is the free stream velocity. The frequency is determined by plotting the velocity magnitude in the wake region and averaging 5 consecutive periods to give St = 0.11. This number compares with numerical results of 0.126 [24] and experimental results of 0.118 [31]. It is likely that the flow domain is not large enough to preclude the influence of the boundary conditions, and the sphere cylinder is under resolved. Thus, this comparison is qualitative at best.

Lid-Driven Cavity Flow

For the LDCF case, the Re = 1000 and Re = 5000 test cases are simulated using the FE–EDAC method and compared with the FD–EDAC method. A grid of 256×256 and a Mach numbers of

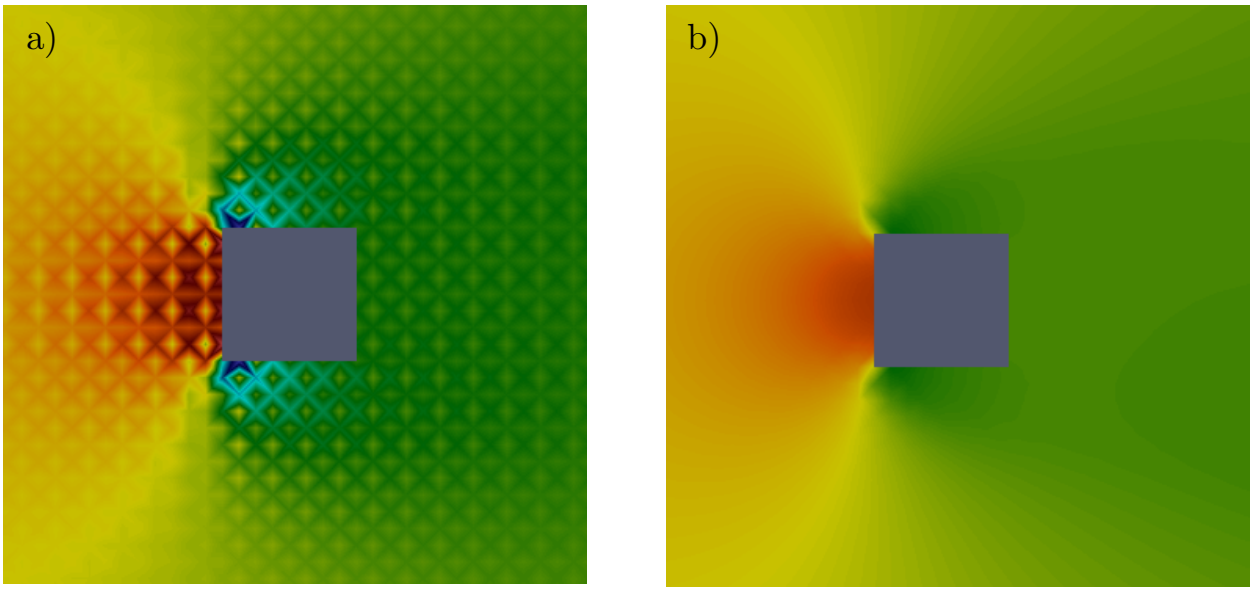

Figure 9. Pressure field created by flow over a blunt object demonstrating (a) pressure locking phenomena along the leading edge where solution gradients are under-resolved, and (b) amelioration of pressure oscillations with suitable pressure stabilization.

0.1 are used. The u and v components of velocity are plotted in figure 11. For the Re = 1000 case, good agreement is seen between previous numerical results and the FD and FE versions of EDAC; however, at the higher Reynolds number, excessive diffusion is seen in the EDAC case. Also plotted for comparison is the SUPG solution for the INS in Sierra/TF. Attempts to limit the pressure smoothing term though a premultiplied constant results in a more accurate solution, but errors still exist. This situation highlights the need for continued research into more appropriate pressure stabilization techniques to be employed with AC methods.

Doubly Periodic Shear Layer

The doubly-periodic shear layer case is also repeated with the FE–EDAC method. The simulations are performed at Re = 10000, and quantitative results include the kinetic energy and enstrophy, as defined in (39). Results are compared with the FD–EDAC results, the high-accuracy pseudospectral results of [29], and a fully implicit SUPG FE formulation. The initial conditions are given as

$$u(x,y) = \begin{cases} \tanh\left(\delta_w(y-0.25)\right), & \text{if } y \leq 0.5 \\ \tanh\left(\delta_w(0.75-y)\right), & \text{if } y > 0.5 \end{cases} \tag{47}$$

$$v(x,y) = \delta_p \sin\left(2\pi(x+0.25)\right), \tag{48}$$

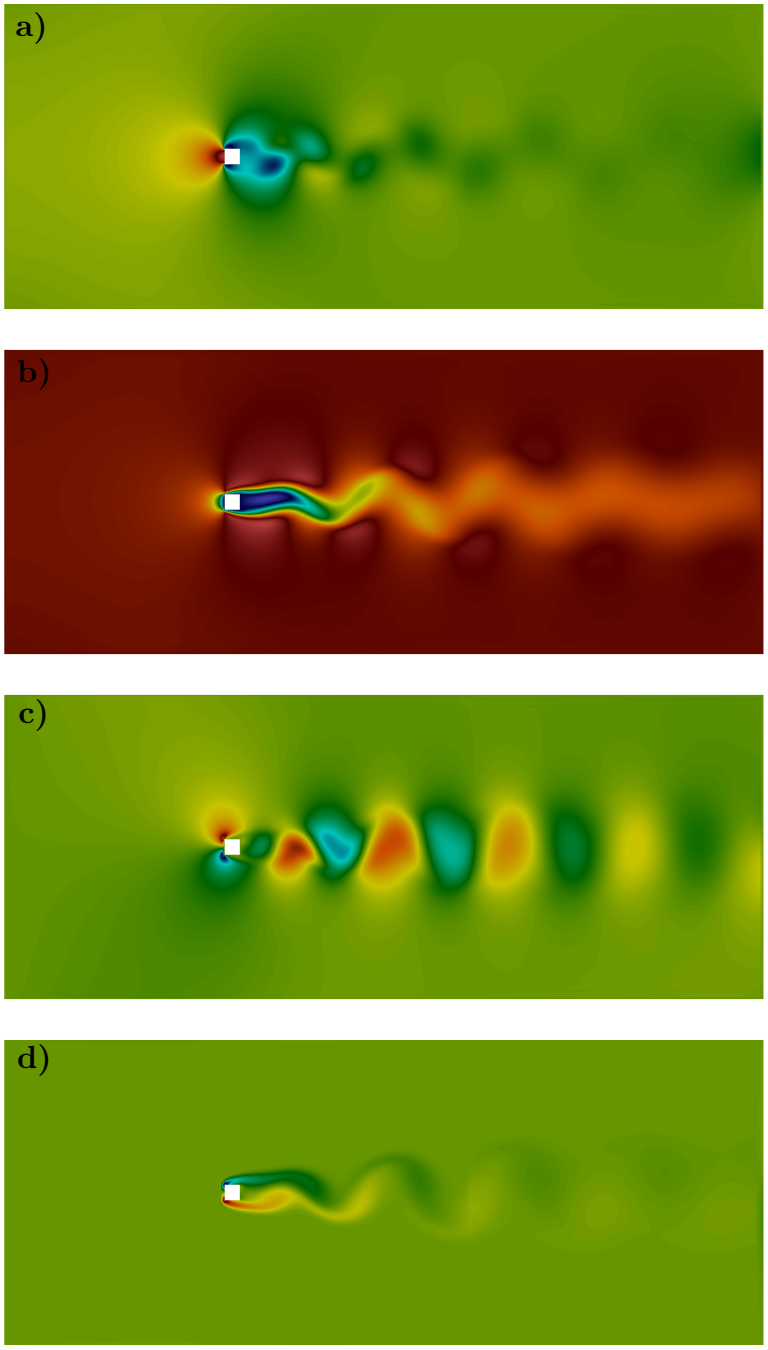

Figure 10. Flow over square cylinder showing a von Kármán vortex street. Images show (a) pressure (b) u-component of velocity, (b) v-component of velocity, and (c) vorticity.

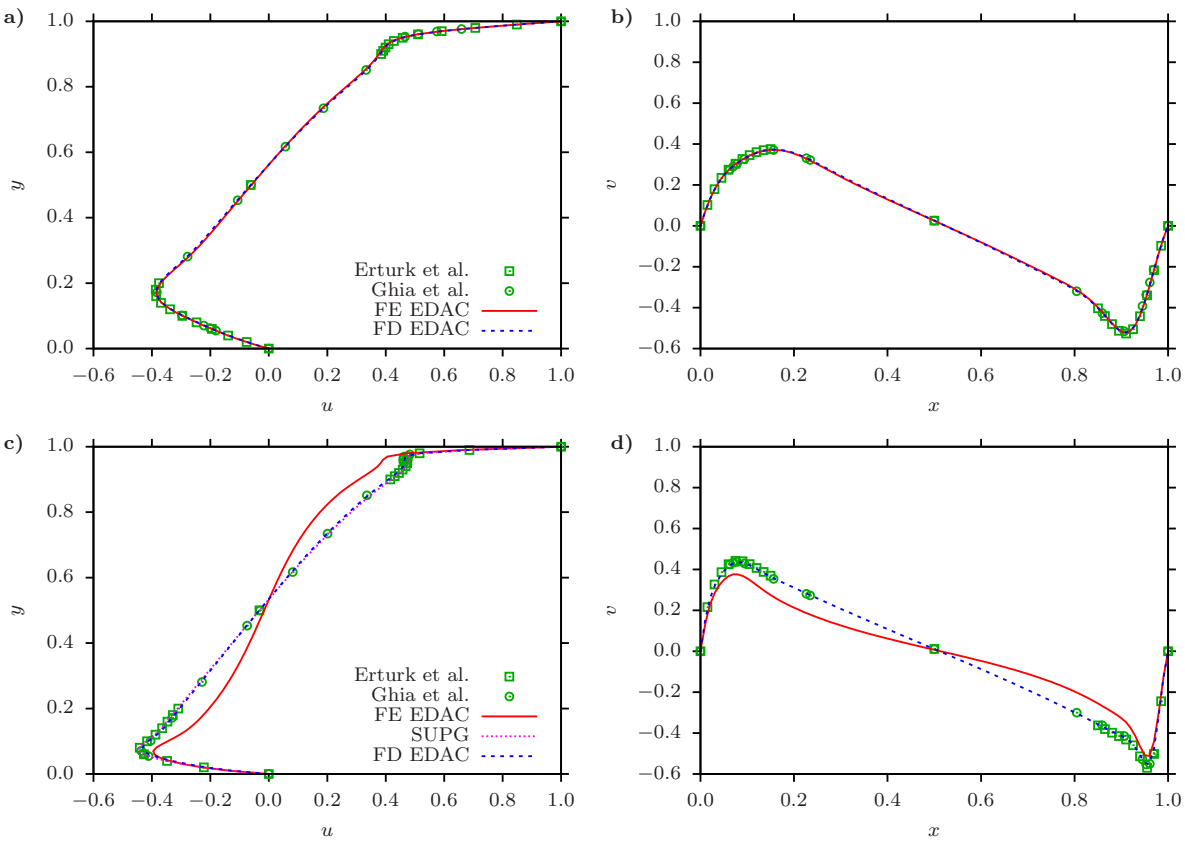

Figure 11. Comparison of u and v velocity components with the FD–EDAC method and existing numerical results.

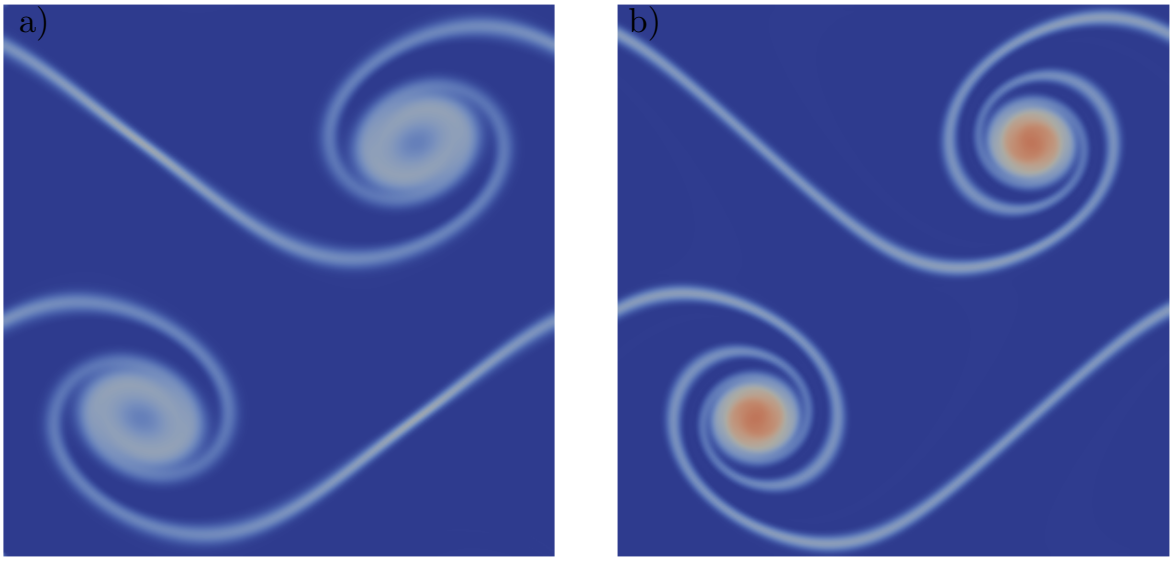

Figure 12. Vorticity magnitude for (a) FE–EDAC and (b) SUPG simulations of the doubly periodic shear layer.

where δ_w is a parameter governing the width of the shear layer and δ_p is the strength of the initial perturbation. All cases considered here use $\delta_w = 80$ and $\delta_p = 0.05$, which corresponds to the thin shear layer cases in [29]. Figure 12 shows vorticity plots of the FD–EDAC, FE–EDAC, and SUPG methods. One can clearly see that the vorticity in the FE–EDAC version is less than the other versions. Quantitatively, this effect can be seen in plots of the enstrophy and energy, shown in figure 13.

Discussion

Clearly the stabilization form chosen, while effective at mitigating pressure-locking phenomena, is too dissipative for general use. Excess dissipation is most evident at higher Reynolds numbers, and makes the FE–EDAC method unsuitable in its current form for these flows. Low-Reynolds-number flows appear to be much more accurate, although more rigorous tests are needed. Alternative forms of stabilization based on a residual formulation are awkward to use in explicit methods, since the concept of a full solution residual does not exist. Efforts to include more terms in the pressure stabilization, such as an advection term or lagged mass, resulted in numerically unstable methods. Including these terms could provide balancing terms for the pressure gradient, which would decrease the overall amount of stabilization provided. Further investigation is needed in order to create a robust FE solver of the EDAC equations.

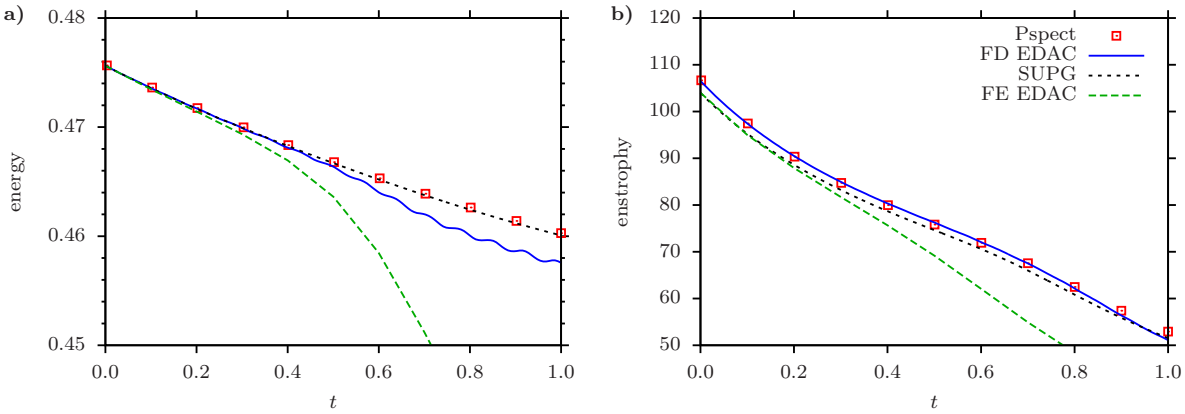

Figure 13. The decay of (a) energy and (b) enstrophy for a doubly periodic shear layer simulation using the FE–EDAC method.

5 Scalability Study

A detailed scaling study is undertaken on the Sequoia computer at Lawrence Livermore National Lab (LLNL)[2]. As mentioned in the introduction, Sequoia is a massively parallel BlueGene/Q architecture that contains approximately 1.6 million processing cores. These cores are contained in 16 core processing units that comprise a single compute node. Theoretical performance is rated at 20 PetaFlops (quadrillion Floating-Point Operations per Second), with 3 petabyte/s interconnect bandwidth, and a 60 terabyte/s bisection bandwidth.

The scaling study is undertaken using the finite difference version of the code (FD–EDAC), rather than the finite element version (FE–EDAC). This choice is driven by two main factors: First, the FE–EDAC version has remaining robustness issues. Second, the FE–EDAC code is implemented in the Sierra-TF codebase, which depends on a complicated software stack, largely due to unstructured techniques. This would make building on Sequoia, optimizing the parallel communication routines, and isolating any performance bottlenecks more time consuming, although certainly feasible.

The FD–EDAC method is a structured Cartesian mesh, thus communications can use the native MPI Cartesian communicator, with load balanced computation per core. A simple ghosting procedure is used, with communication occurring after each predictor and corrector step. At this point, file output is not adequately parallel, and will not scale over a moderate number of cores due to file system limitations. The problem simulated for these benchmarks is the Kida turbulence decay of section 3.3, at Re = 10000.

In scaling studies, two forms of scalability are commonly used: strong and weak scaling. In strong scaling, the problem size remains fixed and the number of cores is increased to see how much faster the simulation runs. This decrease in simulation time is quantified using the speedup, defined as

$$\text{speedup} = \frac{t_{\text{baseline cores}}}{t_{n\ \text{cores}}}, \tag{49}$$

where t is the physical or wall time, and the n subscript refers to the number of cores. For a perfectly parallel algorithm, doubling the number of cores halves the computation time, which yields a speedup of two. Another useful metric is the parallel efficiency

$$\text{efficiency} = \frac{\text{speedup}}{\text{no. cores}}, \tag{50}$$

which is ideally 100%.

For the strong scaling case, two different simulation domains are used. The first is $1024 \times 1024 \times 1024$ with 4.3 million unknowns, and the second is $2048 \times 2048 \times 2048$ with 34 million unknowns. The scalability (speedup) and parallel efficiency of the $1024\times1024\times1024$ case is shown in figure 14. The smaller case ranges between 4096 and 1048576 cores, with each core responsible for from 262144 to 1024 finite-difference grid points. Positive scalability is seen all the way to one million cores, i.e., at no point did increasing the number of cores result in a slower simulation.

[2] https://asc.llnl.gov/computing_resources/sequoia/

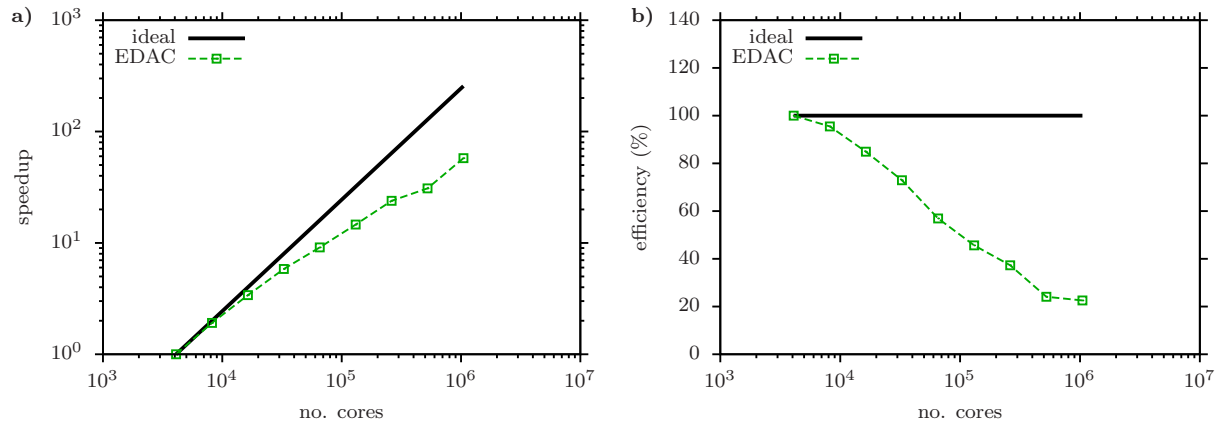

Figure 14. Strong scaling studies for the $1024 \times 1024 \times 1024$ domain size showing the (a) scalability and (b) parallel efficiency.

Adding more cores would still speed the calculation; however, these additional cores would not be efficiently used, as the parallel efficiency does degrade quite noticeably in excess of 65536 cores. At more extreme core counts, the amount of computation to communication decreases. For example, the 262 thousand core case corresponds to a 16^3 domain (4096 nodes) for each core. Assuming one level of ghosting, this corresponds to communicating $17^2 * 6 = 1734$ nodes. The $2048 \times 2048 \times 2048$ case is shown in figure 15. For the larger overall problem, the relative speedup and efficiencies are a bit closer to ideal. The largest simulation is again over one million cores, and corresponds to 8192 finite different grid points per core.

The second metric of scalability is weak scaling, in which the relative computational work per core is held fixed, while the simulation domain is increased. For a perfectly scalable algorithm, the addition of more work is offset by the addition of more processing cores, and the total simulation time should remain fixed. The weak scalability is shown in figure 16 from 512 to 262144 cores. The corresponding domain sizes range from 256^3 to 2048^3, and the simulations are run for a fixed number of time steps. Good weak scalability is seen, although there seems to be some variability in the results as evidenced by the speedup for the 4096 core case.

One caveat to the weak scaling must be mentioned. Figure 16 shows the time to a fixed number of time steps. In order to maintain a constant time step size, a fixed discretization size is needed to maintain the same time step size due to stability constraints. This situation would correspond to increasing the size of the simulation domain, while maintaining the same grid spacing. A perhaps more useful form of weak scaling is one where the grid resolution is increased, while keeping all flow parameters fixed. In this case, due to the CFL time step restriction associated with the propagation of numerical sounds waves, $\Delta t \sim \Delta x$. Thus, increasing the mesh resolution from 256^3 to 2048^3 while maintaining the same Reynolds number, a simulation to the same physical time would take 8 times as long, and a weak scaling linear in the number of cores would be observed.

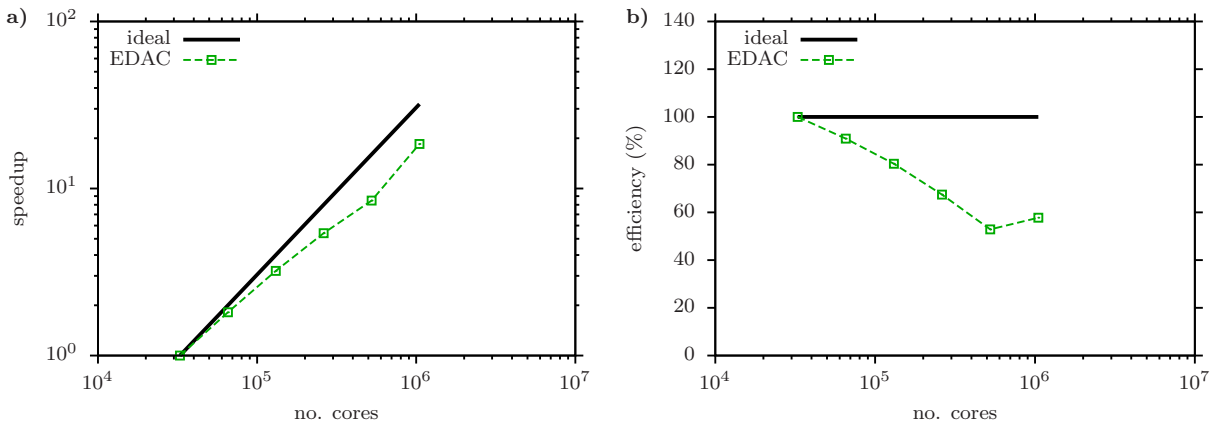

Figure 15. Strong scaling studies for the $2048 \times 2048 \times 2048$ domain size showing the (a) scalability and (b) parallel efficiency.

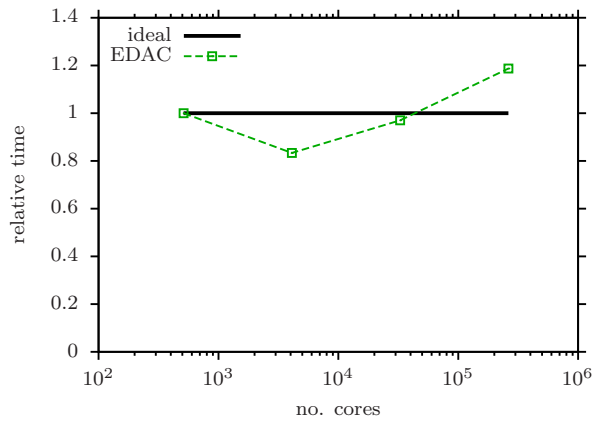

Figure 16. Weak scaling studies performed at a fixed computational load of 32^3 computational nodes per core.

6 Conclusions

We have developed an extension to the standard AC techniques called the EDAC method. This method uses a new thermodynamic constraint in order to form a modified pressure evolution equation to use in place of the continuity equation. The original AC formulation [10] enforces isentropic behavior, which is undesirable since it minimizes dissipation of acoustic pressure waves. These pressure waves are entirely numerical and undesirable. The EDAC method, in contrast, is not isentropic, which serves to damp out the pressure fluctuations created through the introduction of a small, but finite, amount of compressibility. The development of the EDAC method follows the recently published KRLNS method [3, 20, 6]; however, the development of the EDAC method does not rely on a change of variables to the grand potential. Instead, the EDAC method progresses directly from the CNS equations. By constraining the density to be constant (in a thermodynamic sense), a relationship between temperature and pressure fluctuations is derived. The result is a smoothing term in the pressure evolution equation. Additionally, by comparing the EDAC equations with the KRLNS equations, a term missing from the KRLNS equations is found and is shown to be vitally important to the simulation of transient flows.

The AC, EDAC, and KRLNS methods are discretized using a simple MacCormack finite-difference scheme to investigate their behaviors. A characteristics analysis is performed to allow the application of more advanced boundary conditions. Example simulations include lid-driven cavity flow, a convecting Taylor–Green vortex, shear-layer roll-up, and freely decaying turbulence. The KRLNS method is shown to produce erroneous results due to the aforementioned missing term. The EDAC method results in a noticeable smoothing of the pressure fluctuations present in the traditional AC method. The errors in the traditional AC method are particularly evident in lower-Reynolds-number flows. Both the AC and EDAC methods converge to the INS in the limit $\mathrm{Ma} \to 0$; however, differences exist in terms of $O(\mathrm{Ma}^2)$. As $\mathrm{Re} \to \infty$, fluid behavior naturally becomes isentropic, and the AC and EDAC methods converge. In contrast to the traditional AC method, the EDAC method is consistent in the limit $\mathrm{Re} \to 0$, since the pressure evolution becomes the pressure-Poisson equation for a Stokes fluid.

Next, the EDAC method is discretized using the finite element method. Results using this discretization method proved less fruitful than the finite-difference version, but several important observations can be made: First, even though the EDAC equations have some minor amount of compressibility, some form of pressure stabilization is needed in order to prevent pressure-locking phenomena. This phenomena is most noticeable in cases where high pressure gradients exist, particularly on under-resolved meshes. The form of pressure stabilization chosen is effective at mitigating pressure-locking phenomena, but is overly diffusive. Secondly, convective stabilization is needed for the momentum equation; however, using all terms required in a residual-based formulation is not possible, and frequently an unstable calculation was encountered. Further research is needed to improve the robustness of a FE version of the EDAC method.

Renewed interest in the general class of pseudo-compressible methods, i.e., methods in which the flow is allowed to have a small but finite level of compressibility, is motivated by the ability of these methods to operate in a fully explicit manner when simulating (nearly)incompressible flows, avoiding the necessarily implicit pressure-Poisson solve. Since explicit methods require only local

communication, they are capable of scaling on massively parallel architectures including leading-class supercomputer clusters and GPU based computations. This massively parallel capability has been one of the reasons that the LB method, also a pseudo-compressible method, has become popular for certain types of simulation. Unlike LB methods, however, the EDAC methods are formed based on continuum conservation laws which can alleviate difficulties in applying more complicated constitutive models as well as alternative numerical discretization schemes (although as seen with the FE discretization, this application may still not be trivial).

To demonstrate the massive scalability of EDAC methods, a scalability study is undertaken on the Sequoia supercomputer. Scalability is shown to over one-million cores. Owing to the ability of the EDAC method to scale to such large core counts, the Scientific Applications and User Support group at Sandia National Labs plans to use the code in the future to evaluate performance and scalability of hardware and software stacks on various clusters around DOE including Sequoia and Cielo. There is also some interest in using the EDAC algorithm to evaluate the performance of Kokkos, the next generation array container, on hybrid GPU-based clusters.

Despite their attractiveness, fully explicit methods have significant limitations, namely the time step limitations due to diffusion, acoustic wave propagation, and convection. As such, these methods excel in transient simulations where time accuracy requires time steps on the order of the stability limits. Examples include turbulent DNS and suspension and multiphase flows. Whether explicit methods can become competitive with traditional implicit-based methods remains an open research question. Explicit algorithms require further research in order to make the algorithms more robust and in the case of fluid flows, more accurate approximations to the INS equations. Furthermore, if computational resources continue to trend towards more parallel architectures, the attractiveness of explicit methods will increase. With AC methods based on continuum variables, like the EDAC method, more advanced techniques could also be employed, including mixed implicit/explicit algorithms, local time stepping routines, and multigrid methods.

References

[1] C. K. Aidun and J. R. Clausen. The lattice-Boltzmann method for complex flows. *Annu. Rev. Fluid Mech.*, 42(1):439–72, 2010.

[2] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, 17(2):358–69, 1996.

[3] S. Ansumali, I. V. Karlin, and H. C. Ottinger. Thermodynamic theory of incompressible hydrodynamics. *Phys. Rev. Lett.*, 94(8):080602, 2005.

[4] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 85(2):257–83, 1989.

[5] O. N. Boratav and R. B. Pelz. Direct numerical simulation of transition to turbulence from a high-symmetry initial condition. *Phys. Fluids*, 6(8):2757–84, 1994.

[6] S. Borok, S. Ansumali, and I. V. Karlin. Kinetically reduced local Navier-Stokes equations for simulation of incompressible viscous flows. *Phys. Rev. E*, 76(6):066704, 2007.

[7] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 168(2):464–99, 2001.

[8] S. Chen and G. D. Doolen. Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):329–64, 1998.

[9] S. S. Chikatamarla, C. E. Frouzakis, I. V. Karlin, A. G. Tomboulides, and K. B. Boulouchos. Lattice boltzmann method for direct numerical simulation of turbulent flows. *J. Fluid Mech.*, 656(1):298–308, 2010.

[10] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2(1):12–26, 1967.

[11] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp*, 22(104):745–62, 1968.

[12] A. J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comp.*, 23(106):341–53, 1969.

[13] J. R. Clausen. Entropically damped form of artificial compressibility for explicit simulation of incompressible flow. *Phys. Rev. E*, 87(1):013309, 2013.

[14] J. R. Clausen, D. A. Reasor, and C. K. Aidun. Parallel performance of a lattice-Boltzmann/finite element cellular blood flow solver on the IBM Blue Gene/P architecture. *Comput. Phys. Commun.*, 181(6):1013–20, 2010.

[15] E. J. Ding and C. K. Aidun. Extension of the lattice-Boltzmann method for direct simulation of suspended particles near contact. *J. Stat. Phys.*, 112(3):685–708, 2003.

[16] E. Erturk, T. C. Corke, and C. Gökçöl. Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers. *Int. J. Numer. Meth. Fluids*, 48(7):747–774, 2005.

[17] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48(3):387–411, 1982.

[18] X. He, G.D. Doolen, and T. Clark. Comparison of the lattice Boltzmann method and the artificial compressibility method for Navier-Stokes equations. *J. Comput. Phys.*, 179(2):439–51, 2002.

[19] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.*, 62(1):40–65, 1986.

[20] I. V. Karlin, A. G. Tomboulides, C. E. Frouzakis, and S. Ansumali. Kinetically reduced local Navier-Stokes equations: An alternative approach to hydrodynamics. *Phys. Rev. E*, 74(3):035702, 2006.

[21] B. Keating, G. Vahala, J. Yepez, M. Soe, and L. Vahala. Entropic lattice Boltzmann representations required to recover Navier–Stokes flows. *Phys. Rev. E*, 75(3):036712, Mar 2007.

[22] S. Kida and Y. Murakami. Kolmogorov similarity in freely decaying turbulence. *Phys. Fluids*, 30:2030–9, 1987.

[23] C. Kiris and D. Kwak. Aspects of unsteady incompressible flow simulations. *Comput. Fluids*, 31(4-7):627–38, 2002.

[24] K. M. Klekar and S. V. Patankar. Numerical prediction of vortex shedding behind square cylinders. *Int. J. Numer. Meth. Fluids*, 14:327–41, 1992.

[25] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[26] R. W. MacCormack. The effect of viscosity in hypervelocity impact cratering. Technical Report 69-354, AIAA, 1969.

[27] C. Merkle and M. Athavale. Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. In *Computational Fluid Dynamics Conference, 8th, Honolulu, HI*, pages 397–407, 1987.

[28] M. L. Minion. A projection method for locally refined grids. *J. Comput. Phys.*, 127(1):158 –78, 1996.

[29] M. L. Minion and D. L. Brown. Performance of under-resolved two-dimensional incompressible flow simulations II. *J. Comput. Phys.*, 138(2):734–65, 1997.

[30] R. R. Nourgaliev, T. N. Dinh, and T. G. Theofanous. A pseudocompressibility method for the numerical simulation of incompressible multifluid flows. *Int. J. Multiphas. Flow*, 30(7-8):901–37, 2004.

[31] A. Okajima. Strouhal numbers of rectangular cylinders. *J. Fluid Mech*, 123:379–398, 1982.

[32] R. L. Panton. *Incompressible Flow*. Wiley-IEEE, 1996.

[33] S. Paolucci. Filtering of sound from the Navier–Stokes equations. Technical Report SAND82-8257, Sandia National Laboratories, 1982.

[34] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Tran.*, 15(10):1787–1806, 1972.

[35] A. Perrin and H. H. Hu. An explicit finite-difference scheme for simulation of moving particles. *J. Comput. Phys.*, 212(1):166–87, 2006.

[36] T . J. Poinsot and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.*, 101(1):104–29, 1992.

[37] Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[38] R. Rao R. Bordawekar, U. Bondhugula. Can CPUs Match GPUs on Performance with Productivity?: Experiences with Optimizing a FLOP-intensive Application on CPUs and GPU. Technical Report RC25033 (W1008-020), IBM T.J. Watson Research Center, 2010.

[39] P. J. Roache. Code verification by the method of manufactured solutions. *J. Fluids Eng.*, 124:4–10, 2002.

[40] S. E. Rogers, D. Kwak, and C. Kiris. Steady and unsteady solutions of the incompresible Navier–Stokes equations. *AIAA J*, 29(4):603–10, 1991.

[41] F. Shakib and T. J. R. Hughes. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Comput. Method. Appl. M.*, 87(1):35–58, 1991.

[42] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher. *Computational fluid mechanics and heat transfer*. Taylor & Francis, Washington, DC, 1997.

[43] K. W. Thompson. Time dependent boundary conditions for hyperbolic systems. *J. Comput. Phys.*, 68(1):1–24, 1987.

DISTRIBUTION:

1	MS 0828	Ryan Bond, 1541
1	MS 0828	Steven Kempka, 1510
1	MS 0380	Kyran Mish, 1542
1	MS 0836	David Noble, 1514
1	MS 0828	Martin Pilch, 1514
1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

This page intentionally left blank.

Sandia National Laboratories