Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 189 (2003) 30–44
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An efficient physics-based preconditioner for the fully implicit solution of small-scale thermally driven atmospheric flows

Jon Reisner [a,*], Andrzej Wyszogrodzki [a], Vincent Mousseau [b], Dana Knoll [b]

[a] Los Alamos National Laboratory, M.S. D401, Los Alamos, NM 87545, USA
[b] Los Alamos National Laboratory, M.S. B216, Los Alamos, NM 87545, USA

Received 3 May 2002; received in revised form 27 November 2002; accepted 10 March 2003

Abstract

In atmospheric flow situations typical of a small-scale atmospheric thermal, a separation of time scales exists between the fast sound wave time scale and the advective time scale. Atmospheric models have been designed to take advantage of this disparity of time scales with numerical approaches such as the semi-implicit or split-explicit approach being used to efficiently step over the fast sound waves. Some of these numerical approaches are first order in time. To improve accuracy over these methods, a fully implicit and nonlinearly consistent (INC) flow solver has been developed for the Navier–Stokes equation set. In our INC method, the equation set is solved by use of the Jacobian-free Newton–Krylov (JFNK) method. An efficient preconditioner has been developed which uses the semi-implicit method to solve the governing equations. Being that this preconditioner was designed to attack the fastest waves in the system and not other features in the implicit system such as advection or turbulent diffusion, the preconditioning technique is labeled as a physics-based preconditioner. A variety of linear solvers including SSOR, Krylov methods and/or multigrid approaches are used to approximately invert the pressure matrix in the semi-implicit algorithm. A suite of simulations will be conducted utilizing different linear solvers for the simple problem of the bouyant rise of a warm bubble. The problem will also document the ability of the INC approach to achieve second order in time accuracy.

Keywords: Navier–Stokes equations; Physics-based preconditioning; Nonlinear solver

1. Introduction

In [1] a novel preconditioning approach was described which enabled the fully implicit and nonlinearly consistent (INC) shallow-water equation set with Coriolis force to be solved in an

* Corresponding author. Tel.: +1-505-665-1889; fax: +1-505-665-3415.
E-mail addresses: reisner@lanl.gov (J. Reisner), wyszog@lanl.gov (A. Wyszogrodzki), vmss@lanl.gov (V. Mousseau), nol@lanl.gov (D. Knoll).

0021-9991/03/$ - see front matter
doi:10.1016/S0021-9991(03)00198-0

accurate and scalable manner. The key component of the preconditioner was that it employed a semi-implicit solution method. The resulting matrix problem was approximately solved using a linear multigrid procedure [2–4]. Being that this preconditioner was designed to attack the fastest waves in the system and not other features in the implicit system such as advection, the preconditioning technique was labeled as a physics-based preconditioner. Note that this physics-based preconditioner required the inversion of only one equation, the height equation, and thus this approach uses considerably less memory than other preconditioning techniques that would typically invert the entire coupled shallow-water system. Likewise the chosen nonlinear solution technique, the Jacobian-free Newtow–Krylov (JFNK) approach, does not require that any of the Jacobian elements of the equation set be stored, but instead uses a matrix-free approximation [5] which results in additional memory savings.

As shown in [6], for certain atmospheric flow situations the use of the explicit method of averages (MOA) approach [7] was found to produce a solution of a given accuracy more efficiently than a semi-implicit method. The primary cause for this finding was that in some atmospheric flow situations the ratio of the speed of the horizontal flow over the speed of sound is not significantly smaller than one, making the use of explicit approaches attractive. Indeed many numerical weather prediction models use explicit time stepping procedures in the horizontal direction [8,9]. In contrast, because the separation between slow and fast scales is typically larger in the vertical direction and higher resolution is employed within the boundary-layer, a semi-implicit approach is commonly used in the vertical direction in most weather prediction models. Exceptions to the above weather prediction models exist such as the Canadian MC2 [10] which does solve a three-dimensional semi-implicit pressure matrix and employs a semi-Lagrangian scheme for advection. The numerical techniques employed in these codes as well as in the MOA approach are potentially first order in time. Hence, a method which is second order in time may be able to produce a specified level of accuracy for a given computational cost more efficiently than first-order approaches. But, before applying the INC approach to a model capable of predicting the weather including the necessary physics, the INC approach will be applied to the rise of a single thermal plume. Additionally, for the chosen problem of atmospheric thermal convection, the velocities in the vicinity of a bubble are usually small compared to the sound speed. To help illustrate the above points, comparisons of accuracy and CPU time will later be made between the explicit MOA method and the INC method.

The chosen equation set for this study will be the fully compressible Navier–Stokes equations written in conservation form with a constant eddy-diffusivity. Future papers will examine how different turbulence closure models with non-constant eddy-diffusivity possibly influence both the development of a physics-based preconditioner and the overall efficiency of the entire algorithm. Unlike the shallow-water equation set discussed in [1] the compressible equation set contains two equations, the conservation of energy and mass, which are responsible for the fast waves. Therefore, a choice exists with respect to which to use in the semi-implicit formulation found within the physics-based preconditioner [11]. Likewise a choice exists with respect to what type of linear solver should be used to approximately invert the semi-implicit pressure matrix. Since in this framework the semi-implicit algorithm is being used as a preconditioner, it need not be solved to a tight tolerance. Thus, it is not obvious whether simple approaches such as SSOR or more complex methods such as Krylov and/or multigrid solvers are needed to invert the matrix. Details concerning the semi-implicit preconditioner will be given in subsequent sections of this paper with the remainder of this paper being organized as follows: in the following section the analytical and the discretized Navier–Stokes equations will be presented; in Section 3 the INC solution procedure will be described; in Section 4 the semi-implicit preconditioner will be presented; in Section 5 both model performance and accuracy results of the INC solver will be presented; and in Section 6 some concluding remarks will be made.

2. Compressible model

In this paper results will be shown for a two-dimensional Cartesian mesh of the compressible Navier–Stokes equations written in flux form. In this formulation the equations can be written as:

$$\frac{\partial u\rho}{\partial t}+\frac{\partial uu\rho}{\partial x}+\frac{\partial vu\rho}{\partial y}=-\frac{\partial p'}{\partial x}+\frac{\partial \kappa\rho\tau^{11}}{\partial x}+\frac{\partial \kappa\rho\tau^{12}}{\partial y}, \tag{1}$$

$$\frac{\partial v\rho}{\partial t}+\frac{\partial uv\rho}{\partial x}+\frac{\partial vv\rho}{\partial y}=-\frac{\partial p'}{\partial y}-g\rho'+\frac{\partial \kappa\rho\tau^{21}}{\partial x}+\frac{\partial \kappa\rho\tau^{22}}{\partial y}, \tag{2}$$

$$\frac{\partial \theta\rho}{\partial t}+\frac{\partial u\theta\rho}{\partial x}+\frac{\partial v\theta\rho}{\partial y}=\frac{\theta\rho}{T}f_{\mathrm{t}}+\frac{\partial F_{\theta x}}{\partial x}+\frac{\partial F_{\theta y}}{\partial y}, \tag{3}$$

$$\frac{\partial \rho}{\partial t}+\frac{\partial u\rho}{\partial x}+\frac{\partial v\rho}{\partial y}=0, \tag{4}$$

$$p=C_0(\rho\theta)^{\Gamma}, \tag{5}$$

where u and v are the Cartesian velocities in the horizontal, x, and vertical directions, y, respectively; θ is the potential temperature, $\theta=T(p_0/p)^{R_{\mathrm{d}}/C_{\mathrm{p}}}$, with T the temperature of the gas, p the pressure of the gas; g is the acceleration due to gravity; $p'=p-p_{\mathrm{e}}$ is the pressure perturbation with $p_{\mathrm{e}}=p_{\mathrm{e}}(z_{\mathrm{c}})$ the environmental pressure; $\rho'=\rho-\rho_{\mathrm{e}}$ is the density perturbation where $\rho_{\mathrm{e}}=\rho_{\mathrm{e}}(z_{\mathrm{c}})$ is the environmental density; $\tau^{i'j'}=(\partial u^{i'}/\partial x^{j'})+(\partial u^{j'}/\partial x^{i'})-(2/3)\delta^{i'j'}(\partial u^{s'}/\partial x^{s'})$ is the strain-rate tensor with the indices, i', j', and s' being from 1 to 2; f_{t} is a smooth heating function used in specifying the thermal; and $F_{\theta x}=\rho\kappa(\partial\theta/\partial x)$ and $F_{\theta y}=\rho\kappa(\partial\theta/\partial y)$ are diffusional fluxes of potential temperature in the x and y directions with $\kappa=50\ \mathrm{m^2\ s^{-1}}$ being the coefficient of diffusion. The equation of state, (5), relates the total pressure, p, to the density, ρ, and the potential temperature, θ, of the gas where $C_0=R_{\mathrm{d}}^{\Gamma}/p_0^{R_{\mathrm{d}}/C_{\mathrm{v}}}$ with $p_0=10^5\ \mathrm{N\ m^{-2}}$ the base state pressure, and $\Gamma=C_{\mathrm{p}}/C_{\mathrm{v}}$. The constants, $C_{\mathrm{p}}/C_{\mathrm{v}}=1004\ \mathrm{J\ K^{-1}\ kg^{-1}}/717\ \mathrm{J\ K^{-1}\ kg^{-1}}=1.4$, and $R_{\mathrm{d}}=287\ \mathrm{J\ K^{-1}\ kg^{-1}}$ are the specific heat of air at constant pressure/volume and the gas constant of dry air, respectively.

Employing the INC approach Eqs. (1)–(5) can be written in discretized form as:

$$u\rho_{i,j}^{n+1}-u\rho_{i,j}^{n}+\omega\left[\frac{\Delta t}{\Delta x}\left(u\widehat{u\rho}_{i+1/2,j}^{n+1}-u\widehat{u\rho}_{i-1/2,j}^{n+1}\right)+\frac{\Delta t}{\Delta y}\left(v\widehat{u\rho}_{i,j+1/2}^{n+1}-v\widehat{u\rho}_{i,j-1/2}^{n+1}\right)+\frac{0.5\Delta t}{\Delta x}\left(p_{i+1,j}^{\prime n+1}-p_{i-1,j}^{\prime n+1}\right)\right.$$
$$\left.-\frac{\Delta t}{\Delta x}\left(\kappa\rho\tau_{i+1/2,j}^{11^{n+1}}-\kappa\rho\tau_{i-1/2,j}^{11^{n+1}}\right)-\frac{\Delta t}{\Delta y}\left(\kappa\rho\tau_{i,j+1/2}^{12^{n+1}}-\kappa\rho\tau_{i,j-1/2}^{12^{n+1}}\right)\right]+(1-\omega)[\cdots^{n}]=F_{u\rho_{i,j}}^{n+1/2}, \tag{6}$$

$$v\rho_{i,j}^{n+1}-v\rho_{i,j}^{n}+\omega\left[\frac{\Delta t}{\Delta x}\left(u\widehat{v\rho}_{i+1/2,j}^{n+1}-u\widehat{v\rho}_{i-1/2,j}^{n+1}\right)+\frac{\Delta t}{\Delta y}\left(v\widehat{v\rho}_{i,j+1/2}^{n+1}-v\widehat{v\rho}_{i,j-1/2}^{n+1}\right)+\frac{0.5\Delta t}{\Delta y}\left(p_{i,j+1}^{\prime n+1}-p_{i,j-1}^{\prime n+1}\right)\right.$$
$$\left.-\frac{\Delta t}{\Delta x}\left(\kappa\rho\tau_{i+1/2,j}^{21^{n+1}}-\kappa\rho\tau_{i-1/2,j}^{21^{n+1}}\right)-\frac{\Delta t}{\Delta y}\left(\kappa\rho\tau_{i,j+1/2}^{22^{n+1}}-\kappa\rho\tau_{i,j-1/2}^{22^{n+1}}\right)\right]+(1-\omega)[\cdots^{n}]=F_{v\rho_{i,j}}^{n+1/2}, \tag{7}$$

$$\rho\theta_{i,j}^{n+1} - \rho\theta_{i,j}^{n} + \omega\left[\frac{\Delta t}{\Delta x}\left(u\widehat{\rho\theta}_{i+1/2,j}^{n+1} - u\widehat{\rho\theta}_{i-1/2,j}^{n+1}\right) + \frac{\Delta t}{\Delta y}\left(v\widehat{\rho\theta}_{i,j+1/2}^{n+1} - v\widehat{\rho\theta}_{i,j-1/2}^{n+1}\right) - \Delta t\left(\frac{\theta\rho f_{\mathrm{t}}}{T}\right)_{i,j}^{n+1}\right.$$
$$\left. - \frac{\Delta t}{\Delta x}\left(F_{\theta x_{i+1/2,j}}^{n+1} - F_{\theta x_{i-1/2,j}}^{n+1}\right) - \frac{\Delta t}{\Delta y}\left(F_{\theta y_{i,j+1/2}}^{n+1} - F_{\theta y_{i,j-1/2}}^{n+1}\right)\right] + (1-\omega)[\cdots^{n}] = F_{\rho\theta_{i,j}}^{n+1/2}, \tag{8}$$

$$\rho_{i,j}^{n+1} - \rho_{i,j}^{n} + \omega\left[\frac{\Delta t}{\Delta x}(u\hat{\rho}_{i+1/2,j}^{n+1} - u\hat{\rho}_{i-1/2,j}^{n+1}) + \frac{\Delta t}{\Delta y}(v\hat{\rho}_{i,j+1/2}^{n+1} - v\hat{\rho}_{i,j-1/2}^{n+1})\right] + (1-\omega)[\cdots^{n}] = F_{\rho_{i,j}}^{n+1/2}, \tag{9}$$

$$p_{i,j}^{n+1} = C_{0i,j}(\rho\theta_{i,j}^{n+1})^{\Gamma}, \tag{10}$$

where $i = 1, nx$ and $j = 1, ny$ with nx and ny being the total number of grid points in the x and y directions, ω is used as a switch between first-order ($\omega = 1$) and second-order temporal differencing ($\omega = 0.5$) with the terms $(1-\omega)[\cdots^{n}]$ being identical to terms being multiplied by ω except these terms are at the old time level, and the hat indicates variables that have been interpolated to a cell face. The spatial discretizations are based on all model variables being defined at the cell-center and the temporal discretization utilizes the second order in time Crank–Nicolson method. In the current approach, the advective flux at a cell boundary is obtained by interpolating (QUICK, [12]) the advected quantities to the cell face with this quantity then being multiplied by the appropriate normalized cell-face velocity component,

$$u_{i\pm1/2,j} = 0.5\left(\frac{u\rho_{i\pm1,j}}{\rho_{i\pm1,j}} + \frac{u\rho_{i,j}}{\rho_{i,j}}\right)\frac{\Delta t}{\Delta x} \quad \text{or} \quad v_{i,j\pm1/2} = 0.5\left(\frac{v\rho_{i,j\pm1}}{\rho_{i,j\pm1}} + \frac{v\rho_{i,j}}{\rho_{i,j}}\right)\frac{\Delta t}{\Delta y}.$$

Note, interpolated and/or flux quantities near a boundary contain a constant in-time environmental component. At the side wall boundaries of the domain, the velocity normal to the boundary was specified; whereas, the velocities normal to the top and bottom boundaries were set to zero. Spatial differencing in the pressure-gradient terms used central differencing, except at the boundaries where either first order in space forward or backward differencing formulas were used.

3. INC solution procedure

Eqs. (6)–(10) are solved using the INC-JFNK approach. Details concerning this approach along with how the physics-based preconditioner is embedded within the INC solution procedure will be given in this section. Additional details and examples concerning the JFNK approach can be found in [1,5,6,13].

Newton's method solves a system of nonlinear equations of the form

$$F(\mathbf{x}) = 0, \tag{11}$$

where $F = F_{\hat{i}} = F_{u\rho_{i,j}},\ F_{v\rho_{i,j}},\ F_{\theta\rho_{i,j}}$, and $F_{\rho_{i,j}}$ is the nonlinear residual for $\hat{i}$ from $\hat{i} = 1, ntot * \ nvar = nmax$, with $ntot$ being the total number of gridpoints and $nvar$ being the total number of variables, and $\mathbf{x}$ is a vector representing the discrete variables $u\rho_{i,j},\ v\rho_{i,j},\ \theta\rho_{i,j}$, and $\rho_{i,j}$. Newton's method solves Eq. (11) by a sequence of steps defined by

$$\mathbf{J}\delta\mathbf{x}^{k} = -F(\mathbf{x}^{k}), \tag{12}$$

where

$$\mathbf{J}_{\hat{i},\hat{j}} = \frac{\partial F_{\hat{i}}}{\partial \mathbf{x}^k_{\hat{j}}} \tag{13}$$

and

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}^k, \tag{14}$$

with the index $\hat{j} = 1, nmax$.

The iteration over k is continued until

$$\frac{\|F(\mathbf{x}^k)\|}{\|F(\mathbf{x}^o)\|} < \epsilon_{\mathrm{nl}}, \tag{15}$$

where ϵ_{nl} is the nonlinear convergence criteria ($\epsilon_{\mathrm{nl}} = 1 \times 10^{-6}$ unless otherwise noted) and for example, the numerator of Eq. (15), being the l_2 norm defined as $\|F(\mathbf{x}^k)\| = [\sum_{\hat{i}}^{nmax} F_{\hat{i}}^2]^{1/2}$. Eq. (12) is solved by using the iterative Krylov solver GMRES [14] with a right preconditioning option for m Krylov iterations. In this framework Eq. (12) can be expressed as

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{P}\delta\mathbf{x}^k_m = -F(\mathbf{x}^k), \tag{16}$$

where $\mathbf{P}$ is the preconditioner. Eq. (16) can be rewritten as

$$\mathbf{J}'\delta\mathbf{x}^k_{m\prime} = -F(\mathbf{x}^k), \tag{17}$$

where

$$\mathbf{J}' = \mathbf{J}\mathbf{P}^{-1} \tag{18}$$

and

$$\delta\mathbf{x}^k_{m\prime} = \mathbf{P}\delta\mathbf{x}^k_m, \tag{19}$$

with $\delta\mathbf{x}^k_m$ coming from the following relationship within GMRES

$$\delta\mathbf{x}^k_m = \mathbf{P}^{-1}\delta\mathbf{x}^k_{m\prime} = \mathbf{P}^{-1}\sum_{ik=1}^{m}\mathbf{v}_{\mathrm{gmres}_{ik}}\eta_{ik}, \tag{20}$$

with both the vector $\mathbf{v}_{\mathrm{gmres}}$ and the coefficients η being computed within GMRES. GMRES is iterated until

$$\frac{\|\mathbf{J}'\delta\mathbf{x}^k_{m\prime} - F(\mathbf{x}^k)\|}{\|\mathbf{J}'\delta\mathbf{x}^k_{0\prime} - F(\mathbf{x}^k)\|} < \epsilon_{\mathrm{L}}, \tag{21}$$

where

$$\epsilon_{\mathrm{L}} = \gamma\|F(\mathbf{x}^{\mathbf{k}})\|, \tag{22}$$

with $\gamma = 10^{-2}$. Typically, the value of ϵ_{L} varies from a relatively large tolerance for the first Newton iteration to a rather tight linear tolerance for the last Newton iteration. Hence, this process reduces the amount of work when the nonlinear residual is large and tightens it when linear accuracy helps the convergence of the nonlinear iteration. Only constant values of γ are considered in this study, and these values will be such that true quadratic convergence of Newton's method is not observed. For further discussion on the subject of choosing γ see [15] or [16] and for the effect of different choices of γ on the solution of the Navier–Stokes equations see [17,18].

In addition to a right-hand side, a preconditioned GMRES solver requires both the action of a preconditioner on a vector and the action of a Jacobian matrix on a vector. For example, GMRES [19] requires the following vector be computed:

$$\mathbf{w} = \mathbf{J}'\mathbf{v}_{\text{gmres}} = \mathbf{J}\mathbf{P}^{-1}\mathbf{v}_{\text{gmres}} \tag{23}$$

for each iteration of the solver.

Both of the two required actions can be realized upon breaking Eq. (23) into a two step process with the first step being

$$\delta\mathbf{z} = \mathbf{P}^{-1}\mathbf{v}_{\text{gmres}}, \tag{24}$$

and the second step being

$$\mathbf{w} = \mathbf{J}\delta\mathbf{z}. \tag{25}$$

In the physics-based preconditioner, only the process $\mathbf{P}^{-1}\mathbf{v}_{\text{gmres}}$ is computed and as will be shown in the following section, the process of $\mathbf{P}^{-1}\mathbf{v}_{\text{gmres}}$ involves solving for only one parabolic equation and not the entire system found within $F(\mathbf{x})$. Individual elements of the Jacobian matrix need not be computed because the action of the Jacobian matrix on a vector can be approximated by

$$\mathbf{J}\delta\mathbf{z} \approx \frac{F(\mathbf{x} + \epsilon\delta\mathbf{z}) - F(\mathbf{x})}{\epsilon}, \tag{26}$$

where

$$\epsilon = \frac{\sum_{i=1}^{nmax} a\|x_i\|}{nmax\|\delta\mathbf{z}\|}, \tag{27}$$

with $a = 1 \times 10^{-8}$.

4. Preconditioner

To be consistent with Eq. (12) the preconditioner must be written as solving for an update based on a residual, "δ form." This section will detail the development of a semi-implicit physics-based preconditioner written in the required form. Though used as a preconditioner, the semi-implicit preconditioner could be used to solve the following equation:

$$\delta\mathbf{x} = -P^{-1}F(\mathbf{x}) \tag{28}$$

for $\delta\mathbf{x}$. The above equation is analogous to Eq. (24) with $F(\mathbf{x})$ replacing $\mathbf{v}_{\text{gmres}}$; however, as a solver Eq. (28) is computed only once per time step whereas Eq. (24) is computed for each Krylov iteration. Upon making several approximations in the temporal discretizations of Eqs. (6)–(10), such as advective and diffusion terms being at time level n and a linearization in time of the equation of state, the discretized Navier–Stokes equations used in the semi-implicit preconditioner can be written as follows:

$$\begin{aligned} u\rho_{i,j}^{n+1} - u\rho_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(u\widehat{u\rho}_{i+1/2,j}^{n} - u\widehat{u\rho}_{i-1/2,j}^{n}\right) + \frac{\Delta t}{\Delta y}\left(v\widehat{u\rho}_{i,j+1/2}^{n} - v\widehat{u\rho}_{i,j-1/2}^{n}\right) + \frac{0.5\Delta t}{\Delta x}\left(p'^{n+1}_{i+1,j} - p'^{n+1}_{i-1,j}\right) \\ - \frac{\Delta t}{\Delta x}\left(\kappa\rho\tau_{i+1/2,j}^{11^n} - \kappa\rho\tau_{i-1/2,j}^{11^n}\right) - \frac{\Delta t}{\Delta y}\left(\kappa\rho\tau_{i,j+1/2}^{12^n} - \kappa\rho\tau_{i,j-1/2}^{12^n}\right) = 0, \end{aligned} \tag{29}$$

$$v\rho_{i,j}^{n+1} - v\rho_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(u\widehat{v\rho}_{i+1/2,j}^{n} - u\widehat{v\rho}_{i-1/2,j}^{n}\right) + \frac{\Delta t}{\Delta y}\left(v\widehat{v\rho}_{i,j+1/2}^{n} - v\widehat{v\rho}_{i,j-1/2}^{n}\right) + \frac{0.5\Delta t}{\Delta y}\left(p_{i,j+1}^{n+1} - p_{i,j-1}^{n+1}\right) - \frac{\Delta t}{\Delta x}\left(\kappa\rho\tau_{i+1/2,j}^{21^n} - \kappa\rho\tau_{i-1/2,j}^{21^n}\right) - \frac{\Delta t}{\Delta y}\left(\kappa\rho\tau_{i,j+1/2}^{22^n} - \kappa\rho\tau_{i,j-1/2}^{22^n}\right) = 0, \tag{30}$$

$$\rho\theta_{i,j}^{n+1} - \rho\theta_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(u\rho^{n+1}\hat{\theta}_{i+1/2,j}^{n} - u\rho^{n+1}\hat{\theta}_{i-1/2,j}^{n}\right) + \frac{\Delta t}{\Delta y}\left(v\rho^{n+1}\hat{\theta}_{i,j+1/2}^{n} - v\rho^{n+1}\hat{\theta}_{i,j-1/2}^{n}\right) - \Delta t\left(\frac{\theta\rho f_t}{T}\right)_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(F_{\theta x_{i+1/2,j}}^{n} - F_{\theta x_{i-1/2,j}}^{n}\right) - \frac{\Delta t}{\Delta y}\left(F_{\theta y_{i,j+1/2}}^{n} - F_{\theta y_{i,j-1/2}}^{n}\right) = 0, \tag{31}$$

$$\rho_{i,j}^{n+1} - \rho_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(u\rho_{i+1/2,j}^{n+1} - u\rho_{i-1/2,j}^{n+1}\right) + \frac{\Delta t}{\Delta y}\left(v\rho_{i,j+1/2}^{n+1} - v\rho_{i,j-1/2}^{n+1}\right) = 0, \tag{32}$$

$$p_{i,j}^{n+1} = C_{0i,j}\rho\theta_{i,j}^{n+1}\left(\rho\theta_{i,j}^{n}\right)^{\Gamma-1}. \tag{33}$$

For the advection within the preconditioner an upstream value for the hat quantity was chosen instead of an interpolated quantity, because a previous study [20] has suggested that using a lower-order advection scheme in lieu of a higher-order scheme can be effective in increasing the computational efficiency of a preconditioner.

To rewrite Eqs. (29)–(33) in a form similar to (12), the following expressions:

$$\delta u\rho_{i,j} = u\rho_{i,j}^{n+1} - u\rho_{i,j}^{*}, \tag{34}$$

$$\delta v\rho_{i,j} = v\rho_{i,j}^{n+1} - v_{i,j}^{*}, \tag{35}$$

$$\delta\rho\theta_{i,j} = \rho\theta_{i,j}^{n+1} - \rho\theta_{i,j}^{*}, \tag{36}$$

$$\delta\rho_{i,j} = \rho_{i,j}^{n+1} - \rho_{i,j}^{*}, \tag{37}$$

where the superscript * indicates the current state of the linear solution, are substituted into Eqs. (29)–(33) to obtain the following:

$$\delta u\rho_{i,j} + \frac{0.5\Delta t}{\Delta x}\left(p_{i+1,j} - p_{i-1,j}\right) = v_{\mathrm{gmres}_{u\rho_{i,j}}}, \tag{38}$$

$$\delta v\rho_{i,j} + \frac{0.5\Delta t}{\Delta y}\left(p_{i,j+1} - p_{i,j-1}\right) = v_{\mathrm{gmres}_{v\rho_{i,j}}}, \tag{39}$$

$$\delta\rho\theta_{i,j} + \frac{\Delta t}{\Delta x}\left(\delta u\rho\hat{\theta}_{i+1/2,j}^{n} - \delta u\rho\hat{\theta}_{i-1/2,j}^{n}\right) + \frac{\Delta t}{\Delta y}\left(\delta v\rho\hat{\theta}_{i,j+1/2}^{n} - \delta v\rho\hat{\theta}_{i,j-1/2}^{n}\right) = v_{\mathrm{gmres}_{\rho\theta_{i,j}}}, \tag{40}$$

$$\delta\rho_{i,j} + \frac{\Delta t}{\Delta x}\left(\delta u\rho_{i+1/2,j} - \delta u\rho_{i-1/2,j}\right) + \frac{\Delta t}{\Delta y}\left(\delta v\rho_{i,j+1/2} - \delta v\rho_{i,j-1/2}\right) = v_{\mathrm{gmres}_{\rho_{i,j}}}, \tag{41}$$

$$p_{i,j} = C_{0i,j}\delta\rho\theta_{i,j}\left(\rho\theta_{i,j}^{n}\right)^{\Gamma-1}. \tag{42}$$

Eqs. (38)–(41) are solved in the standard semi-implicit manner, with (38) and (39) being substituted into (40), the energy equation, to obtain an equation of the form,

$$a^{si}_{i,j}\delta\rho\theta_{i-2,j} + b^{si}_{i,j}\delta\rho\theta_{i-1,j} + d^{si}_{i,j}\delta\rho\theta_{i,j} + c^{si}_{i,j}\delta\rho\theta_{i+1,j} + e^{si}_{i,j}\delta\rho\theta_{i+2,j} + f^{si}_{i,j}\delta\rho\theta_{i,j-2} + g^{si}_{i,j}\delta\rho\theta_{i,j-1} + h^{si}_{i,j}\delta\rho\theta_{i,j+1} + i^{si}_{i,j}\delta\rho\theta_{i,j+2} = rhs^{si}_{i,j}, \tag{43}$$

with the coefficients $(a-i)$ being computed numerically using a procedure described in [6]. The variable $rhs^{si}_{i,j}$ is obtained by averaging normalized velocities, $v_{\text{gmres}}u\rho\Delta t/\Delta x$ and $v_{\text{gmres}}v\rho\Delta t/\Delta y$, to the appropriate cell-face, substituting this expression into Eq. (40), and after executing the upstream advection scheme, combining the resulting term with $v_{\text{gmres}}\rho\theta$ to obtain $rhs^{si}_{i,j}$. Note that unlike the original equation set, the above scalar equation (43) is linear and can be approximately inverted by using many linear solvers.

When employing the semi-implicit method as a preconditioner to Eq. (24), the following steps are conducted in each call to the preconditioner:

- If a new time step, then compute the coefficients of Eq. (43) from old time values of $\mathbf{x}$.
- Given $\mathbf{v}_{\text{gmres}}$ construct the $rhs^{si}_{i,j}$ in Eq. (43).
- Approximately solve Eq. (43) with a fixed number of iterations using some linear approach such as a Krylov and/or a multigrid solver.
- Given $\delta\rho\theta$, compute $\delta u\rho$ and $\delta v\rho$ from Eqs. (38) and (39).
- Use the newly computed velocities to solve for $\delta\rho$ in Eq. (41).
- Pass the newly updated vector, $\delta\mathbf{z}$, containing $\delta u\rho$, $\delta v\rho$, $\delta\rho\theta$, and $\delta\rho$ to the Krylov solver.

5. Results

The preconditioner along with the entire JFNK machinery will be used to simulate the bouyant rise of a thermal. The simulations have been designed such that intensity of the thermal produces a velocity field which is at most 5% the speed of sound, hence a low Mach number flow regime. Since this scheme is fully implicit, no stability limit exists; however, to ensure accuracy of the scheme the time step should not exceed the grid resolution divided by speed at which the bubble rises – defined as the dynamical time scale of the problem. Fig. 1 reveals that both the dynamical time scale and the maximum time step allowed by a fully explicit code change little during the middle to later stages of a simulation; whereas, the maximum time step allowed by a code with explicit advection decreases within a simulation and eventually lies below the dynamical time scale. Hence, the possibility exists for a time step to be used by an INC simulation that exceeds the time step allowed by a simulation employing explicit advection.

The domain of the simulations is 1280×1280 m^2 with for example, a domain employing 64×64 gridpoints having a spatial resolution of 20 m. The initial flow is zero with θ and ρ constant and set to 300 K and 1 kg m^{-3}, respectively. The thermal is forced by the following smooth function:

$$f_t = \exp(-f^2_{\text{time}})\exp(-f^2_{\text{space}}), \tag{44}$$

where $f_{\text{time}} = (t - 30\text{ s})/10$ s; and $f_{\text{space}} = \text{dis}(x^1_c, x^2_c)/100$ m with $\text{dis}(x^1_c, x^2_c)$ the distance in meters from the center of the bubble.

5.1. Preconditioner performance

An important component of this work is illustrating the efficiency of the method and in particular the ability of the physics-based preconditioner to significantly reduce computational timings. Another point to be addressed in this section regards the degree of complexity of the linear solver required to approximately

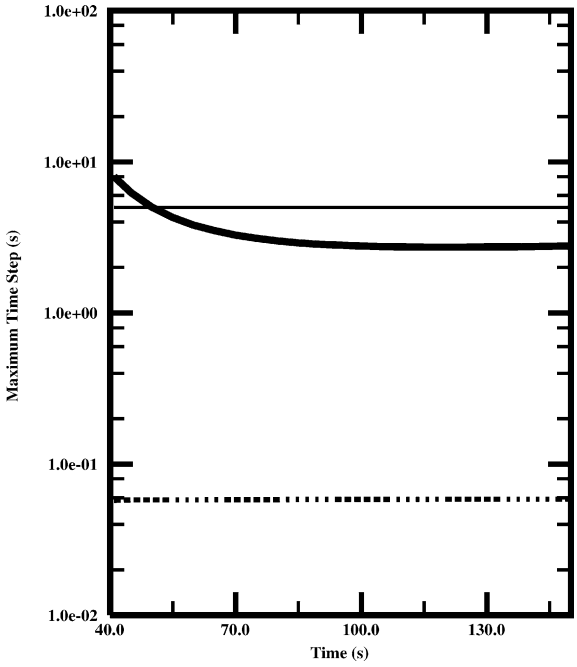

Fig. 1. Approximate maximum time step allowable for the thermal rise problem for a fully explicit method (dashed line), for an approach which steps over fast waves but utilizes explicit advection (thick solid line), and for an INC approach (thin solid line). The time steps were determined by examining the temperature, the velocity fields, and the speed of movement of the bubble from an INC simulation with a constant grid spacing of 20 m and $\Delta t = 0.0025$ s.

invert Eq. (43) and how the complexity of the linear solver used to invert this equation then translates into the efficiency of the overall approach. Since the linear solver could itself employ a preconditioner, a large number of possibilities exist with respect to what type of solver and/or preconditioner could be used to approximately invert Eq. (43). For this study, the following linear solvers within the preconditioner were chosen, a simple multigrid solver utilizing two v-cycles and SSOR as the smoother, a conjugate residual Krylov (CR) solver [21] with and without SSOR preconditioning, and a SSOR solver. Optimization studies in which for example, the number of SSOR iterations were varied, have been conducted with only the "best" possible choice/s in terms of minimum CPU time for a particular linear solver being shown. To understand how the various linear solvers perform with problem dimension and time step size, sets of simulations were run in which either one of those two key parameters was varied. This was achieved by fixing $\Delta t/\Delta x$ to investigate how the solvers scale with problem dimension and then varying Δt to investigate how time step size influences problem scaling. In the performance studies, three grids were used that employed 64×64, 128×128, and 256×256 gridpoints, respectively (Table 1).

Table 2 presents results for a fixed $\Delta t/\Delta x = 0.05$ study (advective cfl ≈ 0.5). Though Table 2 reveals some differences in CPU timings between the various linear solvers used to invert Eq. (43), the most important thing to note from the table is the ability of the physics-based preconditioner to approximately lower CPU time and the number of Krylov iterations by a factor of 10. Scaling with problem size is adequate with the multigrid solver producing the best scaling among the various linear solvers. What is not shown in the table is that each solver is not solving Eq. (43) to a specified tight linear tolerance, but instead solves the equation for a fixed number of iterations. Increasing the number of iterations taken does little to decrease either CPU timings or equivalently the average number of GMRES iterations taken per time step,

Table 1
Definition of labels used in subsequent tables

Label	Definition
Type	Type of linear solver used to invert Eq. (43)
Time	CPU time in seconds for the 64×64 grid
NoPre	No preconditioner used
Mult-v2-2	Multigrid solver using two v-cycles and two SSOR smoothing passes
Mult-v2-4	Multigrid solver using two v-cycles and four SSOR smoothing passes
Kry-4-2	CR solver using four iterations with two SSOR smoothing passes for each iteration
Kry-6-2	CR solver using six iterations with two SSOR smoothing passes for each iteration
Kry-18	CR solver using 18 iterations
Kry-23	CR solver using 23 iterations
SSOR-20	SSOR with 20 smoothing passes
Krylov/Δt	The average number of GMRES iterations per time step averaged over each of the three different grids, 64×64, 128×128, and 256×256.
Ratio 1	Ratio of CPU timings from a 128×128 grid over a 64×64 grid
Ratio 2	Ratio of CPU timings from a 256×256 grid over a 128×128 grid
Ratio $N1$	Ratio of CPU timings from a simulation with an $\epsilon_{\text{nl}} = 1 \times 10^{-6}$ over a simulation with an $\epsilon_{\text{nl}} = 1 \times 10^{-2}$

Table 2
The effect of solver type used to approximately solve the semi-implicit equation set in the physics-based preconditioner for a fixed $\Delta t/\Delta x = 0.05$

Type	Krylov/Δt	Time	Ratio 1	Ratio 2
NoPre	311.2	3944	7.85	–
Mult-v2-2	25.6	348	8.65	10.80
Kry-4-2	23.9	430	9.71	12.73
Kry-18	24.8	580	9.68	11.52
SSOR-20	24.0	356	8.93	12.38
MOA	–	720		

In the table the timing for the MOA approach is for a 64×64 grid from a simulation utilizing a time step that produces an l_2 norm error comparable to that from an INC simulation for $\Delta t/\Delta x = 0.05$ obtained at the end of the simulation. All timings are for the first 30 s of simulation time.

due in large part to differences in the discretizations used in the preconditioner versus the INC equation set. Thus, even if a direct solver could be used to invert Eq. (43), the benefits gained in terms of reduction of CPU timings would probably be small, and could possibly be substantially larger, depending on what type of direct solver is employed to invert Eq. (43). Table 2 also indicated that the explicit MOA approach is almost twice as expensive as the INC simulations for a comparable error measure (to be shown in the following section). Hence, at least for this particular separation of fast and slow time scales (see Fig. 1), the fully implicit INC approach is faster than an explicit approach.

When $\Delta t/\Delta x$ is increased from 0.05 to 0.1 or nearing the maximum allowable from a simulation using explicit advection, CPU timings decrease slightly (see Table 3) and suggest that the physics-based preconditioner scales in a reasonably consistent manner with time step size. The effect of increasing ϵ_{nl} on the timings of the entire solver is shown in Table 4. Increasing ϵ_{nl} does have an appreciable effect on reducing the overall timings with simulations employing an $\epsilon_{\text{nl}} = 1 \times 10^{-2}$ being about 2–4 times faster than simulations employing an $\epsilon_{\text{nl}} = 1 \times 10^{-6}$. Also, CPU timings are smaller for the last 30 s of simulation time than for the first 30 s, possibly indicating that for the first 30 s sound waves produced by the forcing function are leading to an increase in CPU timings.

Table 3
Same as Table 1, except $\Delta t/\Delta x = 0.1$

Type	Krylov/Δt	Time	Ratio 1	Ratio 2
NoPre	543	3961	14.53	–
Mult-v2-2	36	239	13.42	10.19
Mult-v2-4	29	237	11.40	11.43
Kry-4-2	51	438	10.80	13.87
Kry-6-2	37	400	9.35	14.42
Kry-18	48	537	11.33	13.43
Kry-23	38	567	10.08	15.80
SSOR-20	45	317	9.98	15.14

Table 4
The effect of changing $\epsilon_{\rm nl}$ and preconditioner type for a fixed $\Delta t/\Delta x = 0.05$

Type	Time (0–30 s)	Ratio $N1$	Time (120–150 s)	Ratio $N1$
NoPre	3994	4.17	2490	5.56
Mult-v2-2	348	3.34	239	2.23
Kry-4-2	430	2.79	303	2.17
Kry-18	580	3.62	404	3.25
SSOR-20	356	3.12	253	2.14

Simulations time for the first 30 s (0–30 s) and the last 30 s (120–150 s) are shown.

5.2. Accuracy

For the assessment of the accuracy of the INC method three sets of simulations were conducted utilizing grids employing 32×32, 64×64, and 128×128 gridpoints, respectively. For each set, two groups of simulations were performed with the first group of simulations being second order in time INC-JFNK simulations of the Navier–Stokes equations and the second group being identical to the first except explicit advection was employed. Since in the second group of simulations the advected quantity was at the old time level, this group of simulations has been labeled as non-INC. Additionally, for the set employing 64×64 gridpoints a group of simulations employing the MOA approach were also conducted. For a given simulation the time step was fixed and for individual simulations within a group, the time step for a particular simulation varied from that allowed by a fully explicit code to a time step nearing the stability and/or accuracy limit of the given approach.

A primary goal of these sets of simulations is to demonstrate the ability of the INC approach to produce a second order in time solution. The chosen measure of error is the l_2 norm which is defined as $[\sum_{l_2=1}^{ntot}(\theta_{l_2} - \theta_{\rm exact})^2]^{1/2}$ with the potential temperature, $\theta_{\rm exact}$, coming from a resolved simulation employing 256×256 gridpoints and a time step of 0.0025 s or about five times smaller than could be used by a fully explicit code. Additionally for the 64×64 grid, a temporally resolved solution utilizing a time step of 0.0025 s was run. Note, $\theta_{\rm exact}$ was averaged in space for each individual set of simulations. Fig. 2 clearly shows that with a time step of 2.0 s or 40 times larger than allowed by a fully explicit code (see Fig. 1), the fully implicit simulation produces a potential temperature field that visually agrees with the potential temperature field from the resolved solution; whereas, potential temperature fields from the two non-INC simulations reveal small visible differences between the resolved solution and the two non-INC simulations.

There are several ways in which to investigate the behavior of the temporal error of both the INC and non-INC simulation sets. For example, l_2 norm errors (see Fig. 3) computed by using the potential temperature field from the temporally resolved simulation on the 64×64 grid suggest the INC method to be second order in time and the non-INC approach to be first order in time. Unfortunately, it is possible that

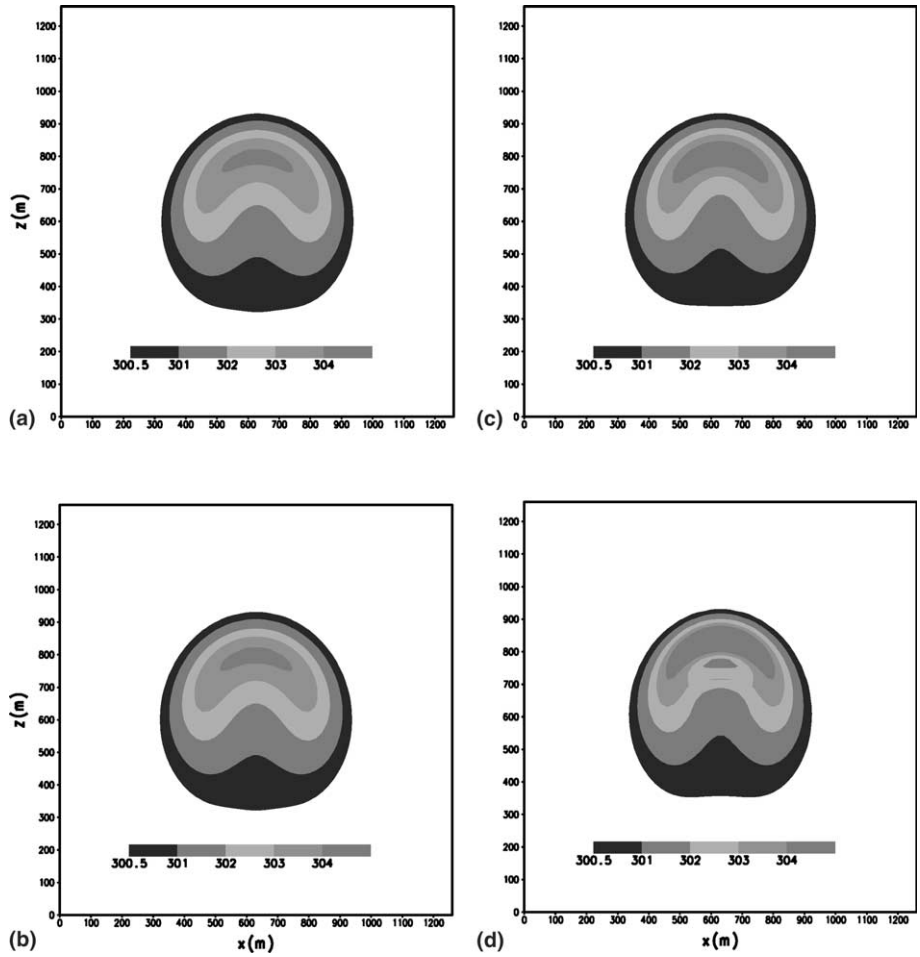

Fig. 2. Potential temperature fields at 150 s from (a) the INC simulation with $\Delta t = 0.0025$ s, (b) the INC simulation with $\Delta t = 2.0$ s, (c) the MOA simulation with $\Delta t = 2.0$ s, and (d) non-INC simulation with $\Delta t = 2.0$ s.

the above error analysis could have been influenced by spatial errors and hence this error analysis may be somewhat misleading. To investigate the interaction between spatial and temporal errors, l_2 norm errors from the various groups of simulations were computed by comparing against the spatially and temporally resolved simulation. Fig. 4 reveals that for a given resolution l_2 norm errors associated with INC simulations are almost independent of time step size; in contrast, l_2 norm errors from the non-INC simulations increase significantly for time steps larger than could be used by a fully explicit code. Additionally, though not shown in Fig. 4, for a non-INC simulation to produce a l_2 norm error of the same magnitude as an INC simulation utilizing a time step near the maximum allowable, requires a time step to be used that is at least an order of magnitude smaller than could be used by a fully explicit code. It is hypothesized that for a given spatial resolution, the absolute difference in l_2 norm errors from two adjacent data points shown in Fig. 4 is primarily due to temporal errors and hence assuming the error data shown in Fig. 3 is mainly temporal error, this new error measure should be of similar magnitude to the errors shown in Fig. 3. Indeed Fig. 5 indicates the l_2 norm errors to be of comparable magnitude, suggesting that temporal errors of a method could be investigated by using either error approach.

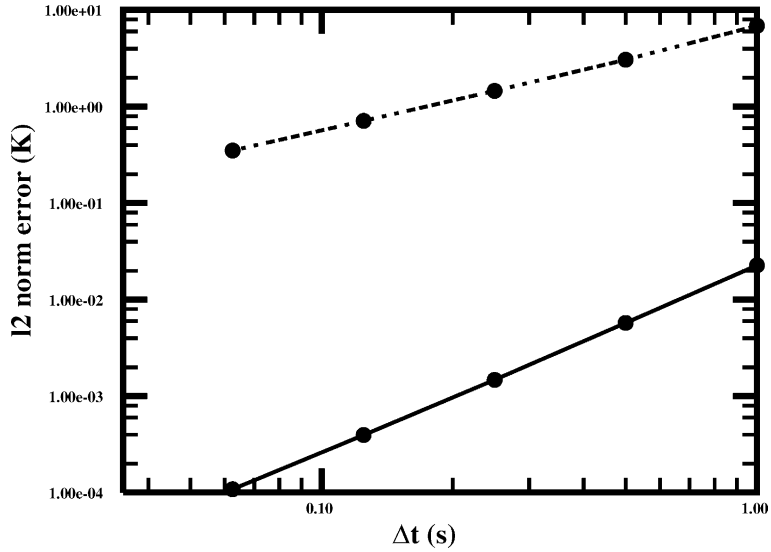

Fig. 3. l_2 norm error versus time step size for potential temperature at 150 s from simulations employing 64×64 gridpoints for INC simulations (solid lines) or non-INC simulations employing explicit advection (dashed lines) with l_2 norm errors computed by comparing against a temporally resolved solution on the 64×64 grid.

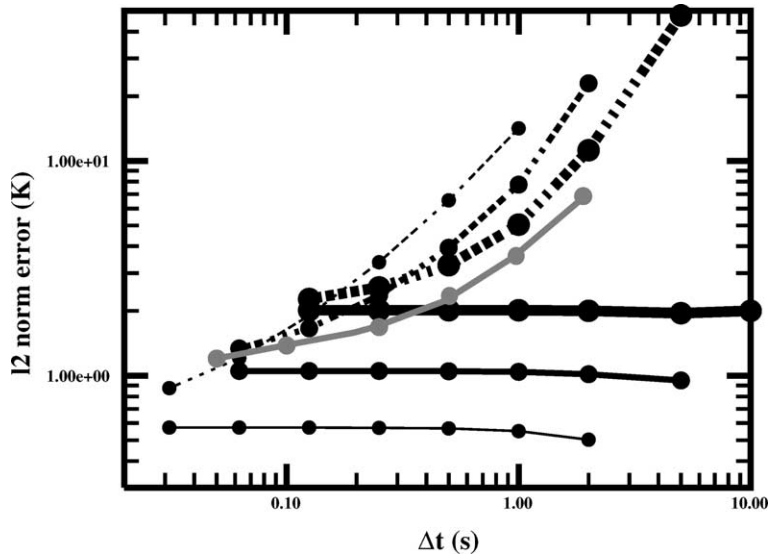

Fig. 4. l_2 norm error versus time step size for potential temperature at 150 s from the three sets of simulations on grids employing either 32×32 (thickest lines), 64×64 (2nd thickest lines), or 128×128 (thinnest lines) gridpoints from either INC simulations (solid lines) or non-INC simulations employing explicit advection (dashed lines). Results from MOA simulations (grey line) employing 64×64 gridpoints are also shown.

6. Concluding remarks

As shown in this paper it is possible to achieve second-order accuracy in time for the Navier–Stokes equation set and still step over the fast wave time scales. A primary component to achieving this accuracy

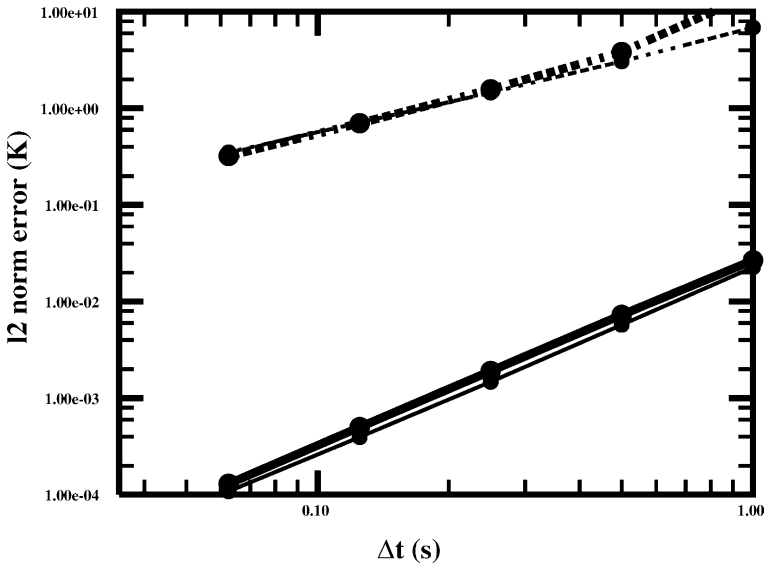

Fig. 5. Same as Fig. 3, except thicker lines computed by subtracting l_2 norm errors between adjacent data points shown in Fig. 4 for the 64×64 grid.

was that the equation set was discretized in a second order in time fully implicit and nonlinearly consistent manner. Our chosen solver for the INC Navier–Stokes equation set utilized the JFNK approach and a key component of this solver was the physics-based preconditioner. Though the physics-based preconditioner was based on solving a semi-implicit discretization and not an INC discretization of the Navier–Stokes equation set, the preconditioner was still able to reduce the total number of Krylov iterations by a factor of 10. This efficiency gain along with the second order in time accuracy should enable the INC-JFNK solution algorithm to remain competitive against the highly efficient explicit approaches commonly used in numerical weather prediction models.

Additionally, this solver is intended to be eventually used in forecasting the movement of wildfires. The physical processes that make up a wildfire model such as combustion, radiation, and diffusion interact in a highly nonlinear manner and splitting these processes could lead to an inaccurate solution algorithm. The promise of the INC approach to eliminate the numerical splitting of physical processes and the ability to accurately step over fast waves could lead to a predictive and efficient solution algorithm for wildfire spread. In addition to being used in a forecast mode, the current solver could be used in determining a steady-state solution for a particular atmospheric event such as a hurricane. The process of ''bogusing'' in a hurricane vortex into an atmospheric prediction model is currently conducted using a relatively crude initialization process, [22,23], and with only a few modifications the current solver should be able to quickly produce a ''realistic'' hurricane vortex which is in relatively good agreement with an observed vortex.

The physics-based preconditioner was tested on a relatively simple problem. For more complex problems that may involve terrain or stretching of the grid, the ability of the linear solver to efficiently solve the semi-implicit matrix will become even more of an issue. For these situations the resulting semi-implicit matrix becomes larger and may require some combination of Krylov and multigrid solvers to efficiently solve the matrix. Additionally, simple solvers such as SSOR that were used in this study may no longer be efficient for the larger semi-implicit matrix. The above topics along with the inclusion of physical parameterizations such as cloud models into the current solver will be investigated in the future.

Acknowledgements

This work was supported under the auspices of the U.S. Department of Energy under DOE contract W-7405-ENG-36 at Los Alamos National Laboratory, LA-UR-01-3593. The authors thank two anonymous reviewers whose helpful comments aided in improving the quality of this manuscript.

References

[1] V.A. Mousseau, D.A. Knoll, J.M. Reisner, An implicit nonlinearly-consistent method for the two-dimensional shallow-water equations with coriolis force, Mon. Weather Rev. 130 (2002) 2611–2625.
[2] W.L. Briggs, A multigrid tutorial, Society for Industrial and Applied Mathematics, Philadelphia, 1987.
[3] D.A. Knoll, W.J. Rider, A multigrid preconditioned Newton–Krylov method, SIAM J. Sci. Comput. 21 (2) (1999) 691–710.
[4] D.A. Knoll, V.A. Mousseau, On Newton–Krylov-multigrid methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 163 (2000) 262–267.
[5] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990) 450–481.
[6] J. Reisner, V. Mousseau, D. Knoll, Application of the Newton–Krylov method to geophysical flows, Mon. Weather Rev. 129 (2001) 2404–2415.
[7] J. Reisner, S. Wynne, L. Margolin, R. Linn, Coupled atmospheric-fire modeling employing the method of averages, Mon. Weather Rev. 128 (2000) 3683–3691.
[8] G.A. Grell, J. Dudhia, D. R. Stauffer, A description of the fifth-generation penn state/NCAR Mesoscale model (MM5), Technical Report NCAR/TN-398 + STR, National Center for Atmospheric Research Boulder, CO, 1994, 121 pp [Available from http://www.mmm.ucar.edu/mm5].
[9] R.A. Pielke, W.R. Cotton, R.L. Walko, C.J. Tremback, W.A. Lyons, L.D. Grasso, M.E. Nicholls, M.D. Moran, D.A. Wesley, T.J. Lee, J.H. Copeland, A comprehensive meteorological modeling system – RAMS, Meteorl. Atmos. Phys. 99 (1992) 69–91.
[10] R. Benoit, M. Desgagne', P. Pellerin, S. Pellerin, Y. Chartier, The Canadian MC2: a semi-Lagrangian, semi-implicit wideband atmospheric model suited for finescale process studies and simulation, Mon. Weather Rev. 125 (1997) 2382–2415.
[11] V. Casulli, D. Greenspan, Pressure method for the numerical solution of transient compressible fluid flows, Int. J. Numer. Methods Fluids 4 (1984) 1001–1012.
[12] B. Leonard, J. Drummond, Why you should not use 'hybrid,' 'power-law' or related exponential schemes for convective modeling – there are better alternatives, Int. J. Numer. Methods Fluids 20 (1995) 421–442.
[13] T.F. Chan, K.R. Jackson, Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms, SIAM J. Sci. Stat. Comput. 5 (1984) 533–542.
[14] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[15] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[16] S.G. Nash, A. Sofer, Assessing a search direction within a truncated-Newton method, Oper. Res. Lett. 9 (1990) 219–221.
[17] P.R. McHugh, D.A. Knoll, Fully implicit finite volume solutions of the incompressible Navier–Stokes and energy equations using inexact Newton's method, Int. J. Numer. Methods Fluids 18 (1994) 439–455.
[18] M. Pernice, H.F. Walker, A Newton iterative solver for nonlinear systems, SIAM J. Sci. Comput. 19 (1998) 302–318.
[19] Y. Saad, M. Shultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM. J. Sci. Comput 7 (1986) 856.
[20] D.A. Knoll, An improved convection scheme applied to recombining divertor plasma flows, J. Comput. Phys. 142 (1998) 473–488.
[21] P.K. Smolarkiewicz, L.G. Margolin, Variational solver for elliptic problems in atmospheric flows, Appl. Math. Comp. Sci. 4 (1994) 527–551.
[22] Y. Kurihara, R. Ross, An initialization scheme of hurricane models by vortex specification, Mon. Weather Rev. 121 (1993) 2030–2045.
[23] Y. Kurihara, R.E. Tuleya, R. Ross, Improvements in the GFDL hurricane prediction system, Mon. Weather Rev. 123 (1995) 2791–2801.