Spectral Preconditioners for Nonhydrostatic Atmospheric Models: Extreme Applications

P.K. Smolarkiewicz[1], C. Temperton[2], S.J. Thomas[1] and A.A. Wyszogrodzki[3]

[1] *National Center for Atmospheric Research, Boulder, Colorado, U.S.A.*
[2] *European Centre for Medium Range Weather Forecasts, Reading, UK.*
[3] *Los Alamos National Laboratory, Los Alamos, New Mexico, U.S.A.*

ABSTRACT

We study the efficacy of spectral preconditioning (of iterative Krylov-subspace solvers) in extreme settings covering a broad range of scales and physical applications, in the context of a massively parallel nonhydrostatic fluid model. We find that while elementary spectral preconditioners offer advantages in certain classes of applications, in general, their performance strongly depends on the integration time step, number of processors versus size of the problem, and the physical problem at hand.

1. Introduction

Elliptic problems arising in numerical simulation of atmospheric/oceanic flows are typically poorly conditioned, nonseparable, and non self-adjoint — features related to domain anisotropy, effects of planetary rotation, ambient stratification, the use of general curvilinear coordinates in the governing equations, or imposing partial-slip conditions along an irregular lower boundary. Such elliptic problems are difficult; that is, a robust, universally effective expert-type approach for their solution does not yet exist. In effect, each particular problem of interest may require the user's intervention in customizing the elliptic solver, in order to achieve a judicious compromise between the accuracy and computational expense of numerical solutions. Although this status quo may be acceptable in research models, it is hardly affordable in community models and production codes (e.g., weather prediction models) that are required to perform reliably for a multitude of users and/or simulated flow conditions.

Among the most effective methods reported for solving difficult elliptic problems are the preconditioned nonsymmetric conjugate-gradient-type (alias Krylov-subspace; hereafter "Krylov", for brevity) iterative schemes — for a concise introduction to Krylov methods, including derivations from variational principles see an earlier ECMWF paper [26]. There exist a number of optional nonsymmetric Krylov solvers, common in computational research and engineering [1][7]. Our method of choice is the restarted generalized conjugate residual GCR(k) algorithm — proven successful in geophysical applications — akin to the popular generalized minimum residual GMRES(k) solver [5][18][23][24][20]. For convenience, GCR(k) is summarized in the following section, where we also introduce necessary terminology, notation, and a notion of preconditioning (left) used throughout this paper; a brief discussion of line-relaxation preconditioners is also included.

Designing an artful preconditioner is important as it can dramatically accelerate solver convergence. In principle, the preconditioner $\mathscr{P}$ can be any linear operator such that $\mathscr{L}\,\mathscr{P}^{-1}$ is definite, where $\mathscr{L}$ symbolizes the original elliptic operator implied by the physical problem at hand. The role of the preconditioner is to substitute the governing elliptic problem $\mathscr{L}(\Psi) - Q = 0$ with an auxiliary problem $\mathscr{P}^{-1}(\mathscr{L}(\Psi) - Q) = 0$ that converges faster (than the original problem) due to a closer clustering of the eigenvalues of the auxiliary elliptic operator

$\mathscr{P}^{-1}\mathscr{L}$; where Ψ and Q symbolize the dependent variable and the rhs, respectively. In other words, the role of the preconditioner is to circumvent the stiffness of the governing elliptic problem. For the preconditioner to be useful, the convergence of the auxiliary problem must be sufficiently rapid to overcome the effort associated with "inverting" the preconditioner itself (i.e., computing $\mathscr{P}^{-1}(\cdot)$). In general, the closer the preconditioner approximates the original operator, the faster the solver converges but the more difficult it is to compute $\mathscr{P}^{-1}(\cdot)$.

There is no general method for designing an optimal preconditioner ([1], section 7). The full complexity of the multi-scale atmosphere and oceans necessarily requires taking into account the anisotropy of the media in the vertical. For instance, the shallowness of the Earth's atmosphere dictates condition numbers $\kappa(\mathscr{L}) \sim \mathscr{O}(10^0)$ for contemporary general circulation models; recall that $\kappa(\mathscr{L})$ can be interpreted as the squared ratio of the longest to the shortest wave present in the system. Because the asymptotic convergence rate of conjugate gradient type methods is proportional to $\kappa(\mathscr{L})^{-1/2}$, a direct preconditioner in the vertical is the 'categorical imperative' of an effective iterative solver for all-scale atmospheric models, [15][20][34][27]. Here, as in earlier works, we take this preconditioning strategy in the vertical for granted, and focus on the horizontal component of $\mathscr{P}$.

Thomas et al. [36] reported advantages of spectral preconditioners, in the context of the serial-code of the Canadian MC2 model (a semi-Lagrangian, semi-implicit elastic, nonhydrostatic all-scale research/weather-prediction type model [34]). Encouraged by this experience we have incorporated spectral preconditioners, in the spirit of [36], in the massively-parallel, nonhydrostatic anelastic, optionally Eulerian or semi-Lagrangian deformable-grid model EULAG for multi-scale research of geophysical flows; cf. [29][17][37] and the references therein. The massive parallelism, flux-form formulation of the Eulerian option, and the grid adaptivity (to either flow features or irregular boundaries of the domain) all affect the performance of spectral preconditioners, thereby making an a priori assessment of their efficacy virtually impossible. Furthermore, the departure of the preconditioner $\mathscr{P}$ from the governing elliptic operator $\mathscr{L}$ can be quite substantial compared to simple line-relaxation schemes,[1] whereupon applications can be encountered for which spectral preconditioners are unsuitable.

In order to learn more about the potential of spectral preconditioners for multiscale problems in geophysical flow simulations, we have tested their performance in extreme settings covering a broad range of scales and physical applications: from a canonical decaying turbulence problem in a triply periodic box, through homogeneous flows past large-amplitude undulating boundaries, mesoscale flows past long winding valleys, to idealized climates. Our present results are far less optimistic than those reported in [36]. In particular, we do not observe universal superiority of spectral preconditioners (over simple line relaxation schemes) found in the context of MC2. While elementary spectral preconditioners offer advantages in some applications, their performance strongly depends on the integration time step (viz. the relative importance of the first guess), number of processors versus size of the problem, and the physical problem at hand.

The paper is organized as follows. The next section contains brief descriptions of the GCR(k) solver and of the line-relaxation and spectral preconditioners. The anelastic model employed in this study is summarized in section 3, including discussions of the analytic formulation, computational approach with implied elliptic problem, and parallelization strategy. Section 4 is devoted to comparative analysis of the efficacy of the spectral and line-relaxation preconditioners for a series of diverse applications. Remarks in section 5 conclude the paper.

[1] Following [36] we assume for $\mathscr{P}$ a separable constant-coefficient approximation to $\mathscr{L}$ by dropping cross derivative terms and averaging metric coefficients across the domain.

2. Elliptic solver

2.1 Generalized conjugate-residual approach

In general, we consider a linear elliptic problem

$$\mathscr{L}(\Psi) \equiv \sum_{I=1}^{M} \frac{\partial}{\partial x^I} \left(\sum_{J=1}^{M} C^{IJ} \frac{\partial \Psi}{\partial x^J} + D^I \Psi \right) - A\Psi = Q \,, \tag{1}$$

with variable coefficients A, C^{IJ}, D^I, Q, and either periodic, Dirichlet, or Neumann boundary conditions; and adopt the following notation. The discrete representation of a field on the grid is denoted by the subscript $\mathbf{i}$; the discrete representation of the elliptic operator on the lhs of (1) is denoted by $\mathscr{L}_{\mathbf{i}}(\Psi)$; and the inner product $\langle \xi \, \zeta \rangle \equiv \sum_{\mathbf{i}} \xi_{\mathbf{i}} \zeta_{\mathbf{i}}$. The preconditioner $\mathscr{P}$ is a linear operator that approximates $\mathscr{L}$ to a greater or lesser degree but is easier to invert. Its role is to substitute (1) with an auxiliary problem that converges faster (than the original problem) due to a closer clustering of the eigenvalues of the auxiliary operator resulting from the superposition of $\mathscr{L}$ and $\mathscr{P}^{-1}$. In this paper, we are primarily concerned with "left" preconditioning that substitutes (1) with an auxiliary problem $\mathscr{P}^{-1}(\mathscr{L}(\Psi) - Q) = 0$. Its accelerated convergence exploits spectral properties of $\mathscr{P}^{-1}\mathscr{L}$. In general, "right" preconditioning is also possible. It augments (1) with $\mathscr{L}\mathscr{P}^{-1}\big(\mathscr{P}(\Psi)\big) = Q$, and its convergence relies on reducing the spectral radius of $\mathscr{L}\mathscr{P}^{-1}$. Left preconditioning assumes $\mathscr{P}$ constant during the iteration process, whereas right preconditioning allows for variable $\mathscr{P}$ as, e.g., in the flexible GMRES (FGMRES) solver of [19].

The GCR(k) method of Eisenstat et al. (1983) may be derived *via* variational arguments (cf. [23][26]). In essence, we augment (1) with a kth-order damped oscillation equation

$$\frac{\partial^k \mathscr{P}(\Psi)}{\partial \tau^k} + \frac{1}{T_{k-1}(\tau)} \frac{\partial^{k-1} \mathscr{P}(\Psi)}{\partial \tau^{k-1}} + ... + \frac{1}{T_1(\tau)} \frac{\partial \mathscr{P}(\Psi)}{\partial \tau} = \mathscr{L}(\Psi) - Q \,; \tag{2}$$

then discretize (2) in a pseudo-time τ to form the affine discrete equation for the progression of the residual errors r; and determine the optimal parameters $T_1, .., T_{k-1}$ and integration increment $\Delta\tau$ (variable in τ) that assure minimization of the residual errors in the norm defined by the inner product $\langle rr \rangle$. This leads to the following algorithm.

For any initial guess $\Psi^0_{\mathbf{i}}$, set $r^0_{\mathbf{i}} = \mathscr{L}_{\mathbf{i}}(\Psi^0) - Q_{\mathbf{i}}$, $p^0_{\mathbf{i}} = \mathscr{P}^{-1}_{\mathbf{i}}(r^0)$; then iterate:

$$\text{For } n = 1, 2, ... \text{until convergence do}$$

$$\text{for } \nu = 0, .., k-1 \text{ do}$$

$$\beta = -\frac{\langle r^\nu \mathscr{L}(p^\nu) \rangle}{\langle \mathscr{L}(p^\nu) \mathscr{L}(p^\nu) \rangle} \,,$$

$$\Psi^{\nu+1}_{\mathbf{i}} = \Psi^\nu_{\mathbf{i}} + \beta p^\nu_{\mathbf{i}} \,,$$

$$r^{\nu+1}_{\mathbf{i}} = r^\nu_{\mathbf{i}} + \beta \mathscr{L}_{\mathbf{i}}(p^\nu) \,,$$

$$\text{exit if } \quad \| r^{\nu+1} \| \leq \varepsilon \,,$$

$$e_{\mathbf{i}} = \mathscr{P}^{-1}_{\mathbf{i}}(r^{\nu+1}) \,,$$

$$\mathscr{L}_{\mathbf{i}}(e) = \left[\sum_{I=1}^{M} \frac{\partial}{\partial x^I} \left(\sum_{J=1}^{M} C^{IJ} \frac{\partial e}{\partial x^J} + D^I e \right) - Ae \right]_{\mathbf{i}} \,,$$

$$\forall_{l=0,\nu} \quad \alpha_l = -\frac{\langle \mathscr{L}(e) \mathscr{L}(p^l) \rangle}{\langle \mathscr{L}(p^l) \mathscr{L}(p^l) \rangle} \,,$$

$$p_{\mathbf{i}}^{\nu+1} = e_{\mathbf{i}} + \sum_{l=0}^{\nu} \alpha_l p_{\mathbf{i}}^{l} \,,$$

$$\mathscr{L}_{\mathbf{i}}(p^{\nu+1}) = \mathscr{L}_{\mathbf{i}}(e) + \sum_{l=0}^{\nu} \alpha_l \mathscr{L}_{\mathbf{i}}(p^{l}) \,,$$

end do ,

reset $[\Psi,\ r,\ p,\ \mathscr{L}(p)]_{\mathbf{i}}^{k}$ to $[\Psi,\ r,\ p,\ \mathscr{L}(p)]_{\mathbf{i}}^{0}$,

end do .

For convergence, the GCR(k) algorithm above requires $\mathscr{L}\mathscr{P}^{-1}$ to be negative definite[2] but not necessarily self-adjoint[3]. Direct evaluation of the elliptic operator on the grid takes place only once per iteration following the preconditioning $e = \mathscr{P}^{-1}(r^{\nu+1})$ that provides an estimate of the solution error $e^{\nu+1} = \Psi^{\nu+1} - \Psi_{\text{exact}}$.

2.2 Line-relaxation preconditioners

A distinctive feature of meteorological flows is their anisotropy in the vertical direction — the larger the ratio of the horizontal scale of the problem to the fluid depth, the stiffer the elliptic problem. A class of simple yet effective preconditioners that mitigate this aspect (of elliptic problems in meteorology) derives from an implicit Richardson iteration [26]

$$\frac{e^{\mu+1} - e^{\mu}}{\Delta\widetilde{\tau}} = \mathscr{P}^{h}(e^{\mu}) + \mathscr{P}^{z}(e^{\mu+1}) - r^{\nu+1} \,, \tag{3}$$

a realization of the preconditioning step $e = \mathscr{P}^{-1}(r^{\nu+1})$ of the GCR(k) solver summarized in section 2.1. Here, $\mathscr{P}^{h}$ and $\mathscr{P}^{z}$ are the "horizontal" and the "vertical" counterparts of the operator $\mathscr{P}$, respectively; $\Delta\widetilde{\tau}$ is a parameter of the iteration (a pseudo-time step) based on spectral properties of $\mathscr{P}^{h}$ [*viz.*, linear stability analysis of (3)], μ numbers successive Richardson iterations, and ν numbers the outer iterations of the Krylov solver.

Assuming $\mathscr{P}$ identical to the governing operator $\mathscr{L}$ in (1) except for neglected cross derivative terms,[4] and a standard discretization in the spirit of the Arakawa A or C grid, the equation (3) leads to a straightforward linear problem

$$\left(\mathscr{I} - \Delta\widetilde{\tau}\mathscr{P}^{z}\right) e^{\mu+1} = \widetilde{R}^{\mu} \,, \tag{4}$$

where $\widetilde{R}^{\mu} \equiv e^{\mu} + \Delta\widetilde{\tau}\left(\mathscr{P}^{h}(e^{\mu}) - r^{\nu+1}\right)$, that can be solved readily using the tridiagonal (Thomas) algorithm. Alternating implicit discretization between the vertical and horizontal counterparts of $\mathscr{P}$ in fractional steps of $\widetilde{\tau}$ leads to alternating-direction-implicit (ADI) preconditioners of [20].

The implicit Richardson preconditioner in (3) can be further improved. Consider extending the Richardson diffusion scheme with respect to $\mathscr{P}^{h}$ to the diagonally-preconditioned Duffort-Frankel type implicit algorithm

$$\frac{\mathscr{D}e^{\mu+1} - \mathscr{D}e^{\mu}}{\Delta\widetilde{\tau}} = \mathscr{P}^{h}(e^{\mu}) - \mathscr{D}(e^{\mu+1} - e^{\mu}) + \mathscr{P}^{z}(e^{\mu+1}) - r^{\nu+1} \,, \tag{5}$$

where $(-1)\mathscr{D}$ stands for the diagonal coefficient embedded within the matrix representing $\mathscr{P}^{h}$ on the grid. Note that adding the relaxation term on the rhs of (3) has the effect of replacing the $\mathscr{D}e^{\mu}$ term with $\mathscr{D}(e^{\mu+1})$ in $\mathscr{P}^{h}(e^{\mu})$ without complicating flux boundary conditions imposed in constructing $\mathscr{P}^{h}(e)$ (cf. [26] for discussions). In the limit $\Delta\widetilde{\tau} \to \infty$, (5) is equivalent to the block Jacobi preconditioner proven effective in meteorological applications [34][35].

[2] An operator $\mathscr{A}$ is said to be definite if $\langle \xi \mathscr{A}(\xi) \rangle$ is either strictly positive (positive definite) or strictly negative (negative definite) for all ξ.

[3] An operator $\mathscr{A}$ is said to be self-adjoint if $\langle \xi \mathscr{A}(\zeta) \rangle = \langle \zeta \mathscr{A}(\xi) \rangle$ for all ξ and ζ.

[4] Formally, this amounts to replacing the coefficients C^{IJ} in (1) by their products with the Kronecker delta δ_{IJ}.

2.3 Spectral preconditioning

The motivation behind spectral preconditioning is to allow for the unconditionally stable pseudo-time discretization also in the horizontal counterpart of the preconditioning operator on the rhs of (3), and to converge with $\delta\widetilde{\tau} \to \infty$; so the resulting fully-implicit Richardson iteration

$$\frac{e^{\mu+1} - e^{\mu}}{\delta\widetilde{\tau}} = \mathscr{P}^h(e^{\mu+1}) + \mathscr{P}^z(e^{\mu+1}) - r^{\nu+1} \tag{6}$$

becomes

$$\mathscr{P}^h(e^{\nu+1}) + \mathscr{P}^z(e^{\nu+1}) = r^{\nu+1} \,, \tag{7}$$

i.e., iteration free.

Assuming in (7) an appropriate Fourier-series representation[5] of the already-updated residual error $r^{\nu+1}$ and the yet-unknown estimate of the solution error $e^{\nu+1}$

$$\begin{aligned} e^{\nu+1} &= \sum_{k,l} \hat{e}_{k,l}(z) \exp[i(k\cdot x + l\cdot y)] \\ r^{\nu+1} &= \sum_{k,l} \hat{r}_{k,l}(z) \exp[i(k\cdot x + l\cdot y)] \end{aligned} \tag{8}$$

leads to

$$\sum_{k,l} \left\{ \mathscr{C}_{k,l}(z)\hat{e}_{k,l} + \mathscr{B}_{k,l}(z)\frac{d^2\hat{e}_{k,l}}{dz^2} - \hat{r}_{k,l} \right\} \exp[i(k\cdot x + l\cdot y)] \equiv 0 \,, \tag{9}$$

and to the corresponding set of independent tridiagonal problems in Fourier space,

$$\forall_{k,l} \quad \left(\mathscr{C}_{k,l}(z) + \mathscr{B}_{k,l}(z)\frac{\delta^2}{\delta z^2} \right) \hat{e}_{k,l}(z) = \hat{r}_{k,l}(z) \,. \tag{10}$$

In (9) and (10), $\mathscr{B}_{k,l}(z)$ and $\mathscr{C}_{k,l}(z)$ symbolize the resulting Fourier-space coefficients of the preconditioning problem in (7). In order to make the approach practical, we *homogenize* (here, average in the horizontal) the coefficients of the elliptic problem in (7).[6] In general, this may lead to substantial departures of the preconditioner $\mathscr{P}$ from the governing operator $\mathscr{L}$. Predicting the impact of the coefficient homogenization on the solvers' performance is difficult, and its a posteriori assessment is one of the goals of this work.

3. Anelastic model

3.1 Analytic Formulation

The scope of this paper justifies merely a brief symbolic description of the governing anelastic model; for physical motivation, thorough mathematical exposition, and a complementary summary of the employed numerical approach refer to the article by Smolarkiewicz and Prusa [31] in this same proceedings volume, and references therein.

To address a broad class of geophysical flows in a variety of domains — with, optionally, Dirichlet, Neumann, or periodic boundaries in each direction —we formulate (and solve) the governing equations in transformed time-dependent curvilinear coordinates

$$(\bar{t}, \bar{\mathbf{x}}) \equiv \big(t, \mathscr{F}(t, \mathbf{x})\big) \tag{11}$$

[5] Particular realizations depend on the assumed boundary conditions and discretization of the governing model equations.

[6] Otherwise, posing the problem entirely in the Fourier space would require transforming the coefficients themselves, and evaluating the products of the resulting series, while dramatically increasing the computational cost of the preconditioner.

The key simplifying assumptions are that: i) both physical and transformed domains are topologically either cuboidal, toroidal, or spheroidal; ii) the coordinates $(t,\mathbf{x})$ of the physical domain are orthogonal and stationary; iii) time flow is the same in both domains; and iv) the transformed horizontal coordinates $(\overline{x},\overline{y})$ are independent of the vertical coordinate z. Given the transformation in (11), the anelastic equations of Lipps and Hemler [13] can be compactly written as follows

$$\overline{\nabla}\bullet(\rho^*\overline{\mathbf{v}}^s)=0\ . \tag{12}$$

$$\frac{d\mathbf{v}}{d\overline{t}}=-\widetilde{\mathbf{G}}(\overline{\nabla}\pi')-\mathbf{g}\frac{\theta'}{\theta_b}-\mathbf{f}\times\mathbf{v}'+\mathbf{M}'+\mathbf{D}\ , \tag{13}$$

$$\frac{d\theta'}{d\overline{t}}=-\overline{\mathbf{v}}^s\bullet\overline{\nabla}\theta_e+\mathscr{H}\ , \tag{14}$$

where, because of the coordinate transformation, the physical and geometrical aspects intertwine each other. Insofar as the physics is concerned: $\mathbf{v}$ denotes the *physical velocity* vector; θ, ρ, and π denote potential temperature, density, and a density-normalized pressure; $\mathbf{g}$ is the acceleration of gravity, and $\mathbf{f}$ the vector of "Coriolis parameter"; $\mathbf{M}$ symbolizes the inertial forces of geospherical metric accelerations; whereas $\mathbf{D}$ and $\mathscr{H}$ symbolize viscous dissipation of momentum and diffusion of heat, respectively. Primes denote deviations from the geostrophically-balanced ambient (alias, environmental) state $\mathbf{v}_e$, θ_e, and the subscript $_b$ refers to the basic state, i.e., a horizontally homogeneous constant-stability hydrostatic reference state (cf. section 2b in [3]).

The geometry of the coordinates in (11) enters the governing equations as follows: in the mass continuity equation (12), $\rho^*\equiv\rho_b\overline{G}$ with $\overline{G}$ denoting the Jacobian of the transformation; whereas in the momentum equation (13), $\widetilde{\mathbf{G}}$ symbolizes the renormalized Jacobi matrix of the transformation coefficients $\sim(\partial\overline{\mathbf{x}}/\partial\mathbf{x})$; $\overline{\nabla}\bullet\equiv\partial/\partial\overline{\mathbf{x}}\bullet$, and the total derivative is given by $d/d\overline{t}=\partial/\partial\overline{t}+\overline{\mathbf{v}}^*\bullet\overline{\nabla}$, where $\overline{\mathbf{v}}^*\equiv d\overline{\mathbf{x}}/d\overline{t}\equiv\dot{\overline{\mathbf{x}}}$ is the *contravariant velocity*. Appearing in the continuity (12) and potential temperature (14) equations is the *solenoidal velocity* (so named for distinction, because of the form continuity takes with it),

$$\overline{\mathbf{v}}^s\equiv\overline{\mathbf{v}}^*-\frac{\partial\overline{\mathbf{x}}}{\partial t}\ , \tag{15}$$

that readily follows — given $\rho_b=\rho_b(\mathbf{x})$, and the time-independent coordinate system in the physical space — from the generic (tensor invariant) form of anelastic continuity

$$\overline{G}^{-1}\left(\frac{\partial\rho^*}{\partial\overline{t}}+\overline{\nabla}\bullet(\rho^*\overline{\mathbf{v}}^*)\right)\equiv 0\ . \tag{16}$$

Use of the solenoidal velocity facilitates the solution procedures because it preserves the incompressible character of numerical equations. While numerous relationships can be derived that express any velocity (solenoidal, contravariant, or physical) in terms of the other, in either transformed or physical coordinate system [17], a particularly useful transformation

$$\overline{\mathbf{v}}^s=\widetilde{\mathbf{G}}^T\mathbf{v}\ . \tag{17}$$

relates the solenoidal and physical velocities directly. For further details of the metric and transformation tensors as well as formulating viscous and dissipative terms in the governing equations, the interested reader is referred to [31] and the references therein.

3.2 Numerical Approximations

Given (16), each prognostic equation that forms the anelastic system (13) and (14) can be written either as a Lagrangian evolution equation

$$\frac{d\psi}{d\overline{t}}=R\ , \tag{18}$$

or Eulerian conservation law

$$\frac{\partial \rho^* \psi}{\partial \bar{t}} + \overline{\nabla} \bullet (\rho^* \overline{\mathbf{v}}^* \psi) = \rho^* R \,. \tag{19}$$

Here ψ symbolizes components of $\mathbf{v}$ or θ', and R denotes the associated rhs.

We approximate either (19) or (18) to second-order accuracy in space and time using the nonoscillatory forward-in-time (NFT) approach — see [29] for a review. A particular NFT algorithm employed here can be formally written as

$$\psi_{\mathbf{i}}^{n+1} = LE_{\mathbf{i}}(\tilde{\psi}) + 0.5\Delta t R_{\mathbf{i}}^{n+1} \;\equiv\; \widehat{\psi}_{\mathbf{i}} + 0.5\Delta t R_{\mathbf{i}}^{n+1}; \tag{20}$$

where $\psi_{\mathbf{i}}^{n+1}$ is the solution sought at the grid point $(\bar{t}^{n+1}, \overline{\mathbf{x}}_{\mathbf{i}})$, $\tilde{\psi} \equiv \psi^n + 0.5\Delta t R^n$, and LE denotes a two-time-level either advective semi-Lagrangian [21] or flux-form Eulerian [22] NFT transport operator (viz. advection scheme).

Equation (20) represents a system implicit with respect to all dependent variables in (13) and (14), because all principal forcing terms are assumed to be unknown at $n+1$.[7] For the physical velocity vector $\mathbf{v}$, it can be written compactly as

$$\mathbf{v}_{\mathbf{i}} = \widehat{\mathbf{v}}_{\mathbf{i}} - 0.5\Delta t \Big(\widetilde{\mathbf{G}}(\overline{\nabla}\pi') \Big)_{\mathbf{i}} + 0.5\Delta t \mathbf{R}_{\mathbf{i}}(\mathbf{v}, \widehat{\theta}) \,, \tag{21}$$

where

$$\mathbf{R}_{\mathbf{i}}(\mathbf{v}, \widehat{\theta}) \equiv -\big(\mathbf{f} \times (\mathbf{v} - \mathbf{v}_e)\big)_{\mathbf{i}} - \mathbf{g} \frac{1}{\theta_b} \Big(\widehat{\theta}_{\mathbf{i}} + 0.5\Delta t \big((\widetilde{\mathbf{G}}^T \mathbf{v}) \bullet \overline{\nabla} \theta_e \big)_{\mathbf{i}} \Big) \tag{22}$$

accounts for the implicit representation of the buoyancy via (14), and the superscript $n+1$ has been dropped as there is no ambiguity. On grids unstaggered with respect to all prognostic variables (e.g., A and B Arakawa grids), (21) can be inverted algebraically to construct expressions for the solenoidal velocity components that are subsequently substituted into (12) to produce

$$\left\{ \frac{\Delta t}{\rho^*} \overline{\nabla} \cdot \rho^* \widetilde{\mathbf{G}}^T \Big[\big(\mathbf{I} - 0.5\Delta t \mathbf{R} \big)^{-1} \big(\widehat{\mathbf{v}} - 0.5\Delta t \widetilde{\mathbf{G}}(\overline{\nabla}\pi') \big) \Big] \right\}_{\mathbf{i}} = 0 \,; \tag{23}$$

that is, an elliptic equation for pressure

$$\left\{ \frac{\Delta t}{\rho^*} \overline{\nabla} \cdot \rho^* \widetilde{\mathbf{G}}^T \Big[\widehat{\widehat{\mathbf{v}}} - \big(\mathbf{I} - 0.5\Delta t \mathbf{R} \big)^{-1} \widetilde{\mathbf{G}}(\overline{\nabla}\pi'') \Big] \right\}_{\mathbf{i}} = 0 \,, \tag{24}$$

where $\widehat{\widehat{\mathbf{v}}} - \big(\mathbf{I} - 0.5\Delta t \mathbf{R} \big)^{-1} \widetilde{\mathbf{G}}(\overline{\nabla}\pi'') \equiv \overline{\mathbf{v}}^s$ defined in (15); cf. [17] for the complete development. Boundary conditions imposed on $\overline{\mathbf{v}}^s \bullet \mathbf{n}$, subject to the integrability condition $\int_{\partial\Omega} \rho^* \overline{\mathbf{v}}^s \bullet \mathbf{n} d\sigma = 0$, imply the appropriate boundary conditions on π'' [17, 37]. The resulting boundary value problem is solved using a preconditioned nonsymmetric GCR(k) solver discussed in section 2 of this paper. Given the updated pressure, and hence the updated solenoidal velocity, the updated physical and contravariant velocity components are constructed from the solenoidal velocities using transformations (17) and (15), respectively.

3.3 Massively parallel implementation

The massively-parallel message-passing FORTRAN program of the anelastic model outlined — named EULAG for its Eulerian/semi-Lagrangian optional integration schemes — and its performance across various platforms have been discussed in [16]. The parallelization strategy adopted in EULAG exploits the notion that

[7] Nonlinear terms in R^{n+1} (e.g., metric terms arising on the globe) may require outer iteration of the system of equations generated by (20) [27]; when included, diabatic, viscous, and subgrid-scale forcings may be first-order-accurate and explicit, e.g., assume $SGS(\psi^{n+1}) = SGS(\psi^n) + \mathcal{O}(\Delta t)$ in R^{n+1}, thereby contributing to the rhs of the resulting elliptic problem; for extensions to moist processes, see [8].

a great majority of numerical simulations of atmospheric/oceanic flows address problems in thin shells. This characteristic anisotropy of the media favors program parallelization based on 2D domain decomposition, with one-to-one mapping of the horizontally-partitioned 3D subdomains onto the 2D processor array, while maintaining the vertical direction unpartitioned. The design of the spectral preconditioner (7) in section 2.3 adopts, essentially, this same strategy based on the horizontal decomposition of the spectral space [2]. In principle, physical and spectral spaces can be decomposed independently. However computationally the most efficient is to distribute both spaces analogously. Each single subdomain is assigned statically to only one processor responsible for the calculations in this subdomain. In general, the physical space requires interprocessor communication for discrete differentiation in the horizontal as well as exchange of global information, viz. reduction operations such as sums and extrema; whereas in the spectral space, only reduction operations are required. The interprocessor communication employs Message Passing Interface (MPI) or Shared Memory (SHMEM) parallel libraries.

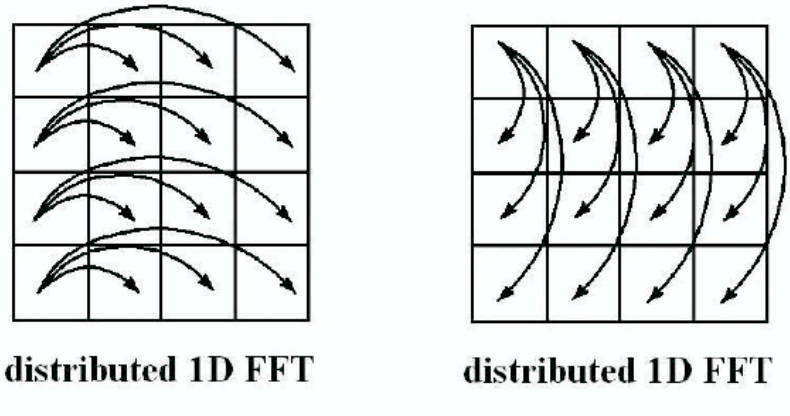

Figure 1: Static block distribution (SBD) method for computing tensor-product Fourier transforms.

The domain decomposition algorithm adopted in EULAG allows for an arbitrary rectangular processor array to cover the entire simulated domain. This embodies two optional strategies of the processor (memory) distribution [2]. The static block distribution (SBD) method is based on two-dimensional equal partition (roughly) of spectral and physical space in both horizontal directions. In the second method, known as the static local distribution (SLD) approach, entire columns (or rows) of the spectral- and physical-space matrices are distributed locally on each processor. In the latter case, each processor independently computes 1D spectral transforms on locally allocated columns (or rows) with the coefficient serial/vector algorithms (sequential Fourier transforms). Then for each individual row (or column) 1D Fourier transform is computed in the parallel mode (distributed transform). In this approach only one parallel phase is computed but the tradeoff between serial- and parallel-phase work strongly depends on the number of processors used. For the same domain size as in the SBD method the smaller number of processors can be used. Furthermore, increasing the number of processors decreases the time of computing the sequential phase but significantly increases the parallel work in the distributed phase.

4. Results

A suite of model runs[8] has been adopted from diverse research projects (documented in the literature) to explore the efficacy of spectral preconditioning outlined in section 2.3. The coefficient homogenization assumed in our preconditioner leaves no doubt that the approach cannot be universal, and will turn out to be ineffective at some point. Unless the numerical model at hand is based on spectral transforms, developing a spectral preconditioner within a massively parallel code is not an effortless task. Thus, to assist the reader in deciding

[8] All calculations reported were performed on unstaggered grids, with 64-bit real arithmetic.

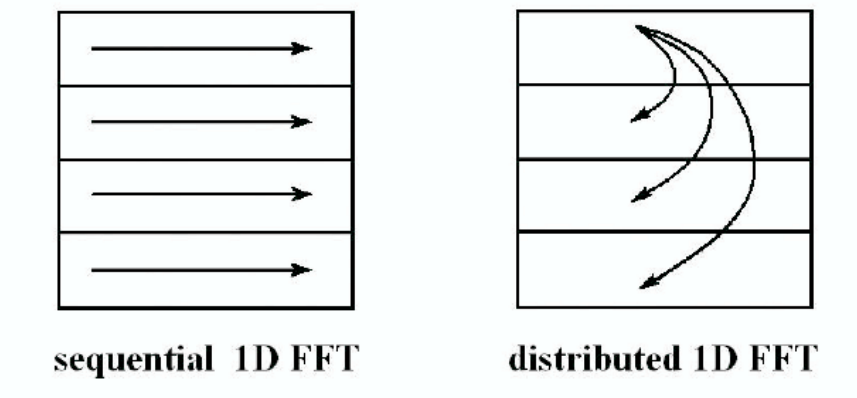

Figure 2: Static local distribution (SLD) method for computing tensor-product Fourier transform.

whether the approach is worth further consideration, we show both its strengths and weaknesses. In contrast to the suite of problems benchmarked in [36], the numerical experiments discussed here tend to extremes rather than being representative of contemporary meteorological applications. We begin with the large-time-step semi-Lagrangian simulation of a flow past rapidly oscillating large-amplitude membranes. Then we follow with the analysis of transient decaying turbulence in a triply periodic box, addressing both the efficacy and scalability issues. The third application is a simulation of a moist rotating stratified flow past a long winding valley in curvilinear domain following the valley, and the last run is a large-eddy simulation of an idealized climate. Each numerical experiment emphasizes different aspects of the elliptic problem in (24), and of the preconditioned GCR(k) solver.

4.1 Flapping Membranes

Here, we consider the numerical simulation of a flow of an ideal 3D homogeneous Boussinesq fluid past oscillating membranes. This test has been presented originally in [37], to validate the correctness of implementation of Dirichlet solenoidal velocity boundary conditions (implying Neumann boundary conditions for pressure) ensuring the integrability condition of the elliptic pressure equation; recall the discussion that follows (24). The membranes form impermeable free-slip upper and lower boundaries, Fig. 3, and their shape is prescribed, respectively, as

$$z_s(r(x,y),t) = \begin{cases} z_{s0}\cos^2(\pi r/2L)\sin(2\pi t/T) & \text{if } r/L \le 1, \\ 0 & \text{otherwise}, \end{cases} \qquad (25)$$
$$H(x,y,t) = H_0 - z_s(x,y,t)\ ,$$

with $r=\sqrt{x^2+y^2}$, oscillation period $T=48\Delta t$, amplitude $z_{s0}=48\Delta z$, the membranes' half-width $L=48\Delta x$, where $\Delta x=\Delta y=\Delta z$, and $\overline{z}(t,x,y,z)=H_0\ (z-z_s)/(H-z_s)$ in (11). Our computational domain consists of $151\times151\times120$ grid intervals, in the horizontal and vertical, respectively; and the LE operator in (20) is semi-Lagrangian. The domain deformation is significant, since at $t=T/4$, the upper and lower boundaries are separated merely by one fifth of the vertical extent of the model. The magnitude of the induced flow and its variation is approximately 5 and 0.5, respectively, as measured by $\mathscr{C}\equiv\|\ \Delta\overline{t}\overline{\mathbf{v}}^*/\Delta\overline{\mathbf{x}}\ \|$ and $\mathscr{L}\equiv\|\ \Delta t\partial\overline{\mathbf{v}}^*/\partial\overline{\mathbf{x}}\ \|$ — the (maximal) Courant and "Lipschitz" numbers (cf. [21] for a discussion).

Lacking diabatic forces, boundary friction, and buoyancy, the experimental setup implies a potential-flow solution with zero integral pressure force on the bounding walls. Indeed, the authors have verified in [37] that the pressure drag is on the order of round-off errors. The supplementary vorticity analysis (validating indeed the vorticity evaluation in curvilinear coordinates) can be found in [31]. Consistent with the accuracy threshold

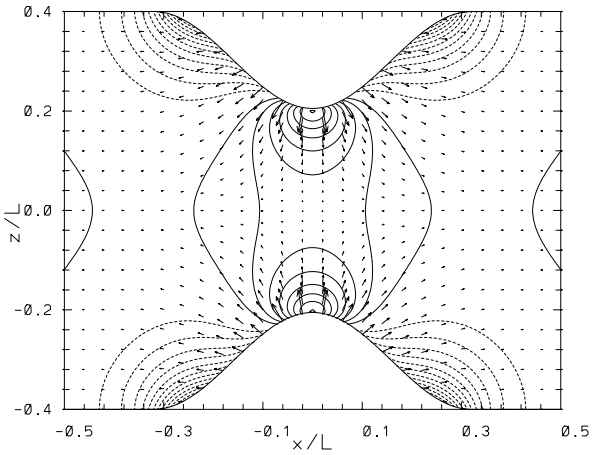

Figure 3: Potential flow simulation past 3D undulating boundaries. Flow vectors and isobares are shown in the central xz plane.

specified for the step "exit if $\| r^{\nu+1} \| \leq \varepsilon$" of the GCR($k$) solver, the quoted (in [31]) value of the divergence of the solenoidal vorticity ($\times \delta t$) is 7 orders smaller than $\mathscr{C}$.

We repeated the experiment as described above, using 16 processors of the IBM SP RS/6000, NCAR "Black-forest" machine. The GCR(4) solver has been employed with either spectral (SP) or line-relaxation (LR) preconditioner. For LR two iterations of the block-Jacobi scheme [34][35] were used. The stopping criterion in the "exit" step of the solver has been set at $\| (\delta t/\rho^*)\overline{\nabla} \bullet \rho^* \overline{\mathbf{v}}^s \|_\infty < 10^{-5}$. The wallclock time (the ultimate measure of the solver performance) for simulated time $t = 22\delta t$, and the average (over the simulation) number of GCR evaluations of the elliptic operator $\mathscr{L}_{\mathbf{i}}(e)$ (viz. solver iterations) per model time step ($\overline{NI}$) are listed in Table 1.

Table 1: Solver performance; flapping membranes

preconditioner	$\overline{NI}$	wallclock time
SP	17	0:18:16
LR	105	0:42:24

The results listed in Table 1 speak for themselves. However, they provide an opportunity to comment on a few aspects of the iterative solvers, that every once in a while appear misconstrued in the literature. Note first that with the simple block Jacobi preconditioner, the number of solver iterations appears discouragingly high. However, this is a special experiment where, due to advantageous large-time-step semi-Lagrangian integration, most of the computational work is in the elliptic solver. In principle, one could design an entirely implicit integration scheme and seek the solution after an appropriately large yet single time step. Then, the entire work would be in the elliptic solver, and the number of iterations accordingly higher. Second, although the spectral preconditioner reduced the number of the solver's iterations sixfold, the wallclock time was reduced only twice. This shows that the computational cost of a spectral preconditioner is substantially higher (several times) than of the simple line-relaxation scheme. In principle, any preconditioned converging iterative solver, can be set to achieve the specified accuracy threshold in merely a few iterations (e.g., consider increasing the number of block-Jacobi passes in the LR preconditioner). These observations illustrate the thesis that judging iterative solvers by the number of iterations required for convergence is misleading. However, reporting the number of iterations is informative, as it aids estimating the comparative solver's cost independent of the machine and implementation.

4.2 Decaying turbulence

Our second benchmark is a simulation of the decaying turbulence of a homogeneous incompressible fluid in a triply-periodic cube — a canonical problem in turbulence studies. The assumed homogeneity of the thermodynamics, and the lack of near-wall effects, focus the problem on the nonlinearity of the convective derivatives $\mathbf{v}\nabla\mathbf{v}$ in the momentum equation, i.e., the 'categorical imperative' of turbulence. Originally, the problem was set forth by Herring and Kerr [11], to investigate the rapid enstrophy growth (blowup) in solutions of the incompressible Euler equations. Recently, it has been employed in several studies to document implicit turbulence modeling property of NFT schemes [28][29][14][9]. The initial condition is posed in the spectral space, such that the energy is equipartitioned among the largest eddies with the wave number value $\| \mathbf{k} \|_2 \leq 4$. For illustration, Fig. 2 shows the isolines of vertical velocity in the center *xy*-plane for the initial condition, and after a non-dimensional unit time.

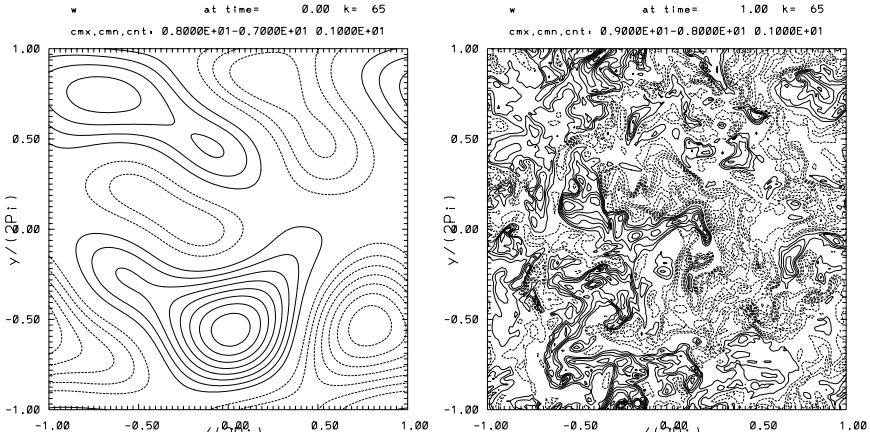

Figure 4: Decaying turbulence problem

For the physical scenario considered — the problem is posed entirely on a Cartesian domain, and is free of forces other than the pressure gradient — the preconditioner $\mathscr{P}$ in (7) is identical to the governing elliptic operator $\mathscr{L}$ in (24). The latter together with direct evaluation of $\mathscr{P}^{-1}$, via (10) and synthesis of e in (8), makes our spectral preconditioner a direct solver of the governing boundary value problem (7). Thus, one might anticipate the GCR solver to converge to machine precision in a single iteration [36]. This is indeed the case for the analogous problem but with impermeable boundaries in the vertical. The periodic boundaries assumed in conjunction with the unstaggered grid discretization employed add some complexity.

In general, periodic tridiagonal systems are more complicated and require more work than their nonperiodic equivalents [33][32]. For the A-grid discretization of the pressure gradient forces in (21) the resulting tridiagonal system (in the vertical) is indeed pentadiagonal and the corresponding Thomas algorithm (cf. section 3.4 in [32]) requires solving not three (like for the compact C-grid type discretization) but five nonperiodic-type systems as well as inverting not a 2×2 but 4×4 linear problem to synchronize cyclic boundaries. Furthermore, on the A grid "2Δ" modes belong to the null space of the discrete gradient operator, whereupon the resulting elliptic operator $\mathscr{L}$ is only semi-definite. Altogether, this leads to a substantial accumulation of the roundoff error while solving (10), especially as the size of the problem increases, and an effective departure of $\mathscr{P}$ from $\mathscr{L}$. In effect, it takes not 1 but 3 GCR iterations (i.e., evaluations of $\mathscr{L}_{\mathbf{i}}(e)$) to solve (24).

In the discussion that follows, we address the efficiency of implementation of the parallel SBD and SLD decompositions of the spectral preconditioner algorithm on a distributed memory IBM SP6000 supercomputer with portable MPI library. We investigate the scalability of the spectral algorithm and compare the results with a non- and LR-preconditioned (using a single pass of the block-Jacobi scheme) GCR solver. A series of tests, using the Eulerian (finite-volume wise) option of the model algorithm, employs two different grid resolutions 128^3 and 256^3 grid points, with constant grid intervals $\delta x = \delta y = \delta z$, and non-dimensional time step $\delta t = 5 \cdot 10^{-4}$. The

selected solver option is GCR(2), equivalent to the classical conjugate-residual scheme of Hestenes and Stiefel [12] — for the problem at hand there are no asymmetries of finite differencing formulae in the computational domain, whereupon $\mathscr{L}$ is self-adjoint and using GCR(2) suffices for the optimal convergence.

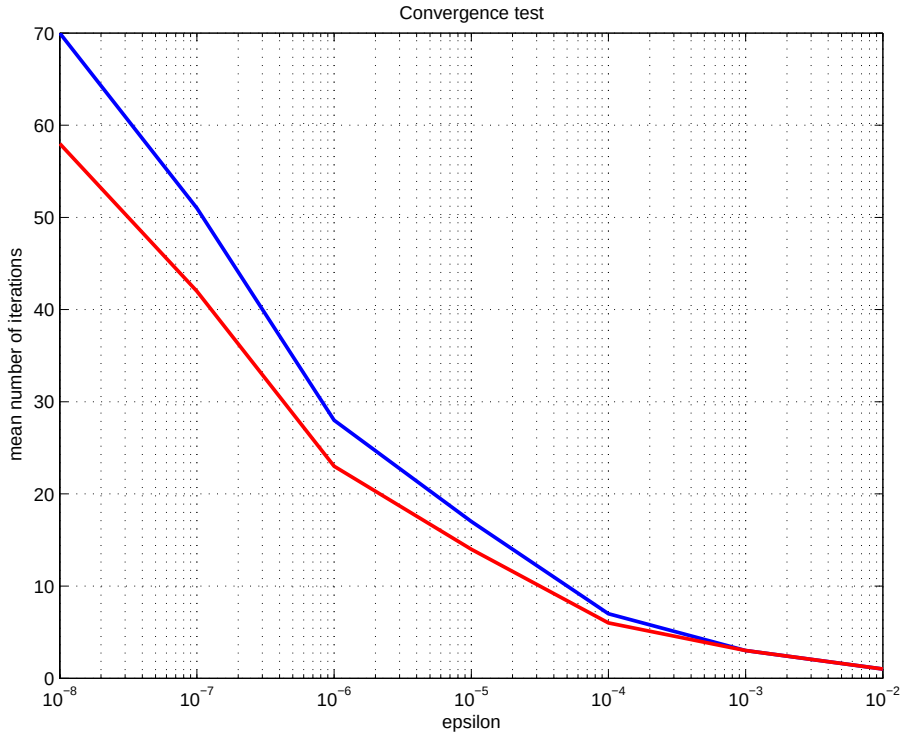

Figure 5: Number of iterations required for convergence

Figure 5 shows the number of solver iterations required to reach the specified convergence threshold $\varepsilon = \delta t \parallel \nabla \mathbf{v} \parallel_\infty$ for the 128^3 resolution. The upper and lower curves correspond to the non- and LR-preconditioned GCR(2) solver. Although the preconditioned solver converges faster, it is somewhat ($\approx 1.2\times$) more computationally expensive than the pure GCR scheme, due to a relatively-high cost of the cyclic tridiagonal algorithm. Since we observe the spectrally-preconditioned GCR(2) to reach $\varepsilon \lesssim 10^{-7}$ in 3 iterations, regardless of the specified threshold in the "exit" step of the algorithm, in all subsequent tests we set $\varepsilon = 10^{-7}$ for the convergence threshold of the LR-preconditioned scheme as well, to assure the fairness (and meaningfulness) of comparisons. This results in about 40 iterations of the LR-preconditioned solver.

Table 2 summarizes the relative cost of spectral preconditioner components for the different number of SBD processors (PE) and 128^3 resolution. The tridiagonal solver requires no parallel work, so it scales well for any number of processors. The Fourier synthesis (FT^{-1}) tends to be less computationally expensive than the analysis (FT) due to a smaller number of global sums (8 for the analysis but only 5 for the synthesis). In both cases the algorithm scales up to 16 processors; above 16 PE, the efficiency of the parallel work deteriorates.

Table 2: Relative cost of the SP components

PE	FT	3dg-solv.	FT^{-1}
1	17.36	4.63	18.62
4	2.37	1.04	2.23
8	1.38	0.43	1.08
16	1.04	0.17	0.63
32	1.33	0.06	0.76
64	1.79	0.03	0.59

Table 3 compares the LR and SP preconditioners for the 128^3 grid resolution runs. For affordability, the numerical data from only the first 10 time steps of each run are analyzed; which is representative, because the work required to solve the elliptic equation (24) remains (roughly) constant over the course of the simulation, due to the transience of the physical problem at hand. Because the number and length of sums invoked in Fourier transformations — and thus the efficiency of spectral preconditioning — depend on the boundary conditions imposed, the comparative tests are also included that assume Neumann boundaries (for pressure) in x or y; labeled X0 and Y0, respectively, for distinction from cyclic X1 and Y1. This comparison documents that the spectral preconditioner is substantially cheaper for Neumann boundaries, the sole option of the development in [36]. This contrasts with the performance of the LR preconditioner, which appears more efficient for periodic boundaries. The increase in execution time for the single-processor run with the LR preconditioner (entries 4650, 7920, 7535, and 10659 in Table 3) was accompanied by the increasing number of iterations (respectively, 112, 179, 178, and 241). Such an unusually high number of iterations is due to the small value of the convergence threshold assumed ($\varepsilon = 10^{-7}$; unnecessary from the physical accuracy viewpoint [24]) as well as the enforcement of the Neumann boundaries, leading to development of the near-boundaries discontinuities in the initially periodic fields.[9] Because of the minor relevance and high computational expense, only the single processor runs were conducted for the LR-preconditioner with Neumann boundaries; mostly for the purposes of illustration. The numbers PX and PY show the horizontal distribution of all processors (PE) in x and y, respectively.

Table 3: Parallel performance of LR and SP preconditioners; 128^3 grid

PE	PX	PY	X1Y1	X1Y0	X0Y1	X0Y0	X1Y1	X1Y0	X0Y1	X0Y0
1	1	1	4650	7920	7535	10659	591	519	459	503
4	2	2	648	-	-	-	123	115	111	79
8	2	4	336	-	-	-	93	61	53	45
8	4	2	348	-	-	-	121	57	62	46
16	4	4	214	-	-	-	54	36	32	28
32	4	8	163	-	-	-	40	29	25	29
32	8	4	179	-	-	-	45	29	28	25
64	8	8	92	-	-	-	47	44	34	36

The results in Table 3 show that the parallel program with the spectral preconditioner scales up to 32 processors, while the scaling for the program with the line-relaxation preconditioner is good for any number of the processors. As the size of the problem increases (Table 4) the scaling of the spectral preconditioner improves.

Table 4: Parallel performance of LR and SP preconditioners; 256^3 grid

PE	PX	PY	X1Y1	X1Y1	X0Y0
64	8	8	1553	380	252
128	16	8	961	350	212
256	16	16	529	358	212

Spectral methods relying on global basis functions are often criticized as being inappropriate for modern, distributed-memory parallel architectures. Certainly, our results show that relatively poor scaling of Fourier transforms degrades the scaling properties of the entire model code. Nonetheless, regardless of a poorer scaling, the parallel program with the spectral preconditioner is up to several times faster, for the problem at hand, than the equivalent program that uses the LR preconditioner with good parallel performance. This illustrates that the scaling arguments should be taken with caution when assessing the utility of spectral methods.

[9] Since the spectral preconditioner is a direct solver, it is essentially insensitive to the spectral composition of the elliptic problem's rhs.

4.3 Mesoscale valley flow

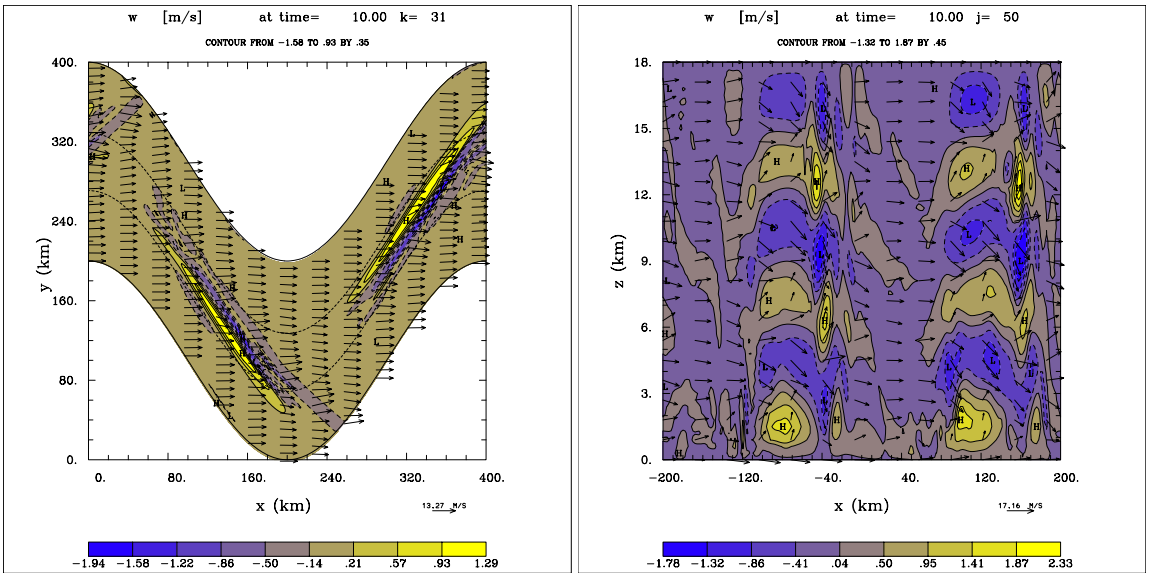

Figure 6: Valley flow. Vertical velocity contours in xy cross section at z=9kmi (left pannel) and on the vertical ribbon aligned with the center of the valley.

The two applications discussed in the preceding subsections both showed advantages of spectral preconditioning. Here, we address a rotating, stratified, mesoscale type flow with substantial variability of ambient parameters in all three directions. The particular problem at hand is a dry inviscid variant of the long-winding valley problem, employed in [31] to illustrate grid adaptivity via a continuous mapping approach. The horizontal model domain in the physical is bounded by two sinusoids of the same x-wavelength $L_x = 400$ km, separated by constant increment 200 km in y. A cosine-shaped valley with the depth and half-width 0.8 km and 30 km, respectively, is centered in the model domain. Here, the vertical domain is 18 km deep. The ambient wind is $(U,0,0)$ with $U = 10$ m/s, and the buoyancy frequency $N = 0.012$ s^{-1}. Boundary conditions are periodic in both horizontal directions. The transformed model domain is covered with $200 \times 100 \times 60$ grid increments. The simulation covers 10 h of physical time with $\Delta t = 60$ s. Figure 4 displays the Eulerian model solution after 10 h.

Table 5 compares four numerical experiments, all performed on the IBM SP RS/6000 machine. The GCR(4) solver has been employed with either spectral or line-relaxation preconditioner. For the line relaxation, the one pass of the block-Jacobi scheme was used. The stopping criterion in the "exit" step of the solver has been set at $\| (\delta t/\rho^*)\overline{\nabla} \bullet \rho^* \overline{\mathbf{v}}^s \|_\infty < \varepsilon$. For each selection of the preconditioner SP or LR, the table lists the accuracy threshold ε, number of processors used PE, the average (over the simulation) number of GCR evaluations of the elliptic operator $\mathscr{L}_{\mathbf{i}}(e)$ per model time step ($\overline{NI}$), and the wallclock time for simulated time $t = 1200\delta t$.

Table 5: Solver performance; mesoscale valley flow

preconditioner	ε	PE	$\overline{NI}$	wallclock time
SP	10^{-5}	20	4	2:54:33
LR	10^{-5}	20	10	1:42:24
SP	10^{-7}	40	13	4:46:21
LR	10^{-7}	40	87	2:42:02

The outcome of the experiments is clear. For a standard model set up ($\varepsilon = 10^{-5}$, cf. [24] for discussion), the simple line relaxation scheme results in a nearly a two times ($1.7\times$) faster model, in spite of a 2.5 larger number

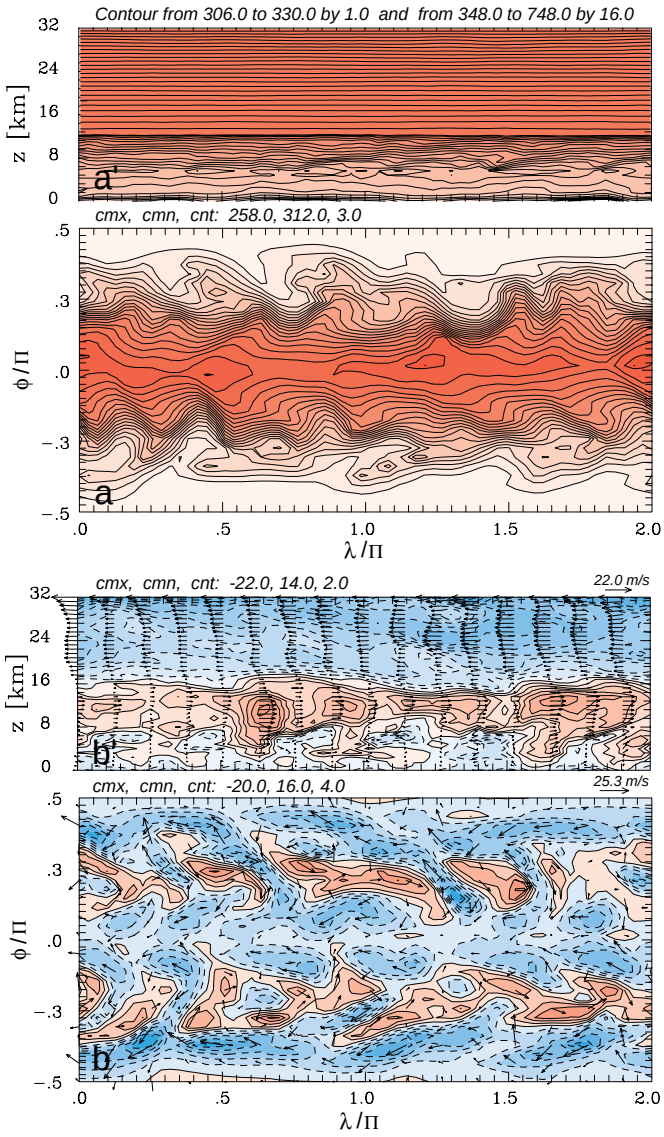

Figure 7: Instantaneous solutions of the idealized climate problem after 3 years of simulation.

of solver iterations. There is no accuracy trade-off, as tightening the threshold by two orders of magnitude only worsens the ratio (to 1.8×) although the iteration count ratio improves from 0.4 to 0.15 in favor of the SP preconditioner.

4.4 Idealized climate

Our last benchmark — the idealized climate problem of Held and Suarez [10] — has been broadly discussed in [27]. In general terms, it represents thermally forced baroclinic instability on the rotating sphere. In a sense, it bears a resemblance to LES studies of decaying turbulence in section 4.2, where small differences in model setups can lead to totally different instantaneous flow realizations. In other words, these simulated flows are both turbulent and stochastic. Figure 7 illustrates the overall complexity of the flow. It shows instantaneous vertical cross-sections in the equatorial plane and surface plots of the isentropes θ and isolines of zonal velocity u, after 3 years of simulated flow. The results displayed typify the response of an initially stagnant and uniformly stratified fluid to a diabatic forcing that mimics the long-term thermal and frictional forcing in the Earth atmosphere.

Here, the globe is covered by a uniform spherical mesh with $nx \times ny = 65 \times 32$ grid intervals (no grid points at the poles); the $H = 32 \cdot 10^3$ m deep atmosphere is resolved with $nz = 40$ uniform grid intervals; and the time step of integration is $\Delta t = 900$ s. The advection scheme is Eulerian, and the elliptic solver is GCR(4), with the line-relaxation preconditioner using 8 iterations of the implicit Richardson iteration (3); see [27] for further details of the simulation performed. To assess the relative efficiency of the LR and SP preconditioner, we have run the experiment (as described) on 24 PE of the IBM SP RS/6000 for 2304 time steps, with the convergence threshold $\| (\delta t/\rho^*)\overline{\nabla} \cdot \rho^* \overline{\mathbf{v}}^s \|_\infty < 10^{-5}$. With LR preconditioner, the solver achieved the threshold in 11 iterations per model time step (on average during the 2304 time steps) and the wallclock time was 0:25:04. In the run with the spectral preconditioner the elliptic solver failed to converge, stalling at $\| r \|_\infty \sim 10^{-3}$ after performing hundreds of iterations within a few initial time steps.

The observed failure of the spectral preconditioner amplifies the result reported in [26], where the implicitness of the Richardson scheme (3) has been extended, by means of spectral decomposition, only in the zonal direction. There, the relatively more relaxed coefficient homogenization sufficed for the solver's convergence on the globe, yet at the rate substantially slower than with the line-relaxation scheme. These results are not unique to the global problems, and examples can be given of a similar behavior for small-scale thermally homogeneous flows.

5. Remarks

Our results corroborate the conclusions of [36] that spectral preconditioning provides a useful alternative to line relaxation schemes. For some problems, a fairly simple spectral approach can accelerate solvers' convergence manifold, leading to substantially faster models. On the other hand, the utility of spectral preconditioners is particular as the *coefficient homogenization* assumed in the (spectral) preconditioner, for the sake of simplicity and computational economy, can be detrimental to the solver convergence in problems with substantial variability of the model coefficients. In other words, spectral preconditioning may turn out well suited for one single-application model, while inappropriate for another.

The relative efficiency of the line-relaxation and spectral preconditioners is problem dependent. Nonetheless, some guidelines, for the worthwhileness of the spectral approach, can be offered based on the results reported. In particular, since spectral preconditioners have substantial overhead compared to the line-relaxation schemes, it appears counterproductive to use them in problems where the line-relaxation preconditioned iterative solver converges in several iterations. A broad class of applications falling under this category, are problems with convergence toward a steady state, where the number of solver iterations typically decreases in the course of the simulation. Conversely, spectral preconditioners may win big in inherently transient problems, where line-relaxation schemes cannot take advantage of slow variability of the solution in a portion of the spectral range (and thus the additivity of error reduction with solver iterations). In general, spectral preconditioners will tend to be more advantageous in large-time-step integrations requiring numerous solver iterations, or in applications where the preconditioner is either identical to, or very closely approximates, the governing elliptic operator.

It should be noted that the two-dimensional Fourier transforms employed in the preconditioners exploited the tensor product nature of the transform, but did not use fast (FFT) techniques in the constituent one-dimensional transforms. Hence, our conclusions on the relative efficiency of spectral preconditioners may be unduly pessimistic. We hope to revisit this question in future work.

Acknowledgements. This work was supported in part by the Department of Energy "Climate Change Prediction Program" (CCPP) research initiative.

References

[1] Axelsson 0, *Iterative Solution Methods*, Cambridge University Press, 1994, pp. 654.

[2] Calvin C. Implementation of parallel FFT algorithms on distributed memory machines with a minimum overhead of communication. *Parallel Computing* 1996; 22: 1255-1279

[3] Clark TL, Farley RD. Severe downslope windstorm calculations in two and three spatial dimensions using anelastic interactive grid nesting: A possible mechanism for gustiness. *J. Atmos. Sci.* 1984; **41**:329–350.

[4] Domaradzki JA, Xiao Z, Smolarkiewicz PK. Effective eddy viscosities in implicit large eddy simulations of turbulent flows. *Phys. Fluids* 2003; **15**:3890–3893.

[5] Eisenstat SC, Elman HC, and Schultz MH, Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.* 1983, **20**, 345–357.

[6] Gal-Chen T, Somerville CJ. On the use of a coordinate transformation for the solution of the Navier-Stokes equations. *J. Comput. Phys.* 1975; **17**:209–228.

[7] Greenbaum A, it Iterative Methods for Solving Linear Systems, SIAM, 2002, pp. 220.

[8] Grabowski WW, Smolarkiewicz PK. A multiscale anelastic model for meteorological research, *Mon. Weather Rev.* 2002; **130**:939–956.

[9] Hahn M, Drikakis D. Large Eddy Simulation of Compressible Turbulence using High-Resolution Methods. *Int. J. Num. Meth. Fluids* 2004, submitted.

[10] Held IM, Suarez MJ. A proposal for intercomparison of the dynamical cores of atmospheric general circulation models, *Bull. Amer. Meteor. Soc.*, 1994; **75**: 1825-1830.

[11] Herring JR, Kerr RM. Development of enstrophy and spectra in numerical turbulence. *Phys. Fluids*, 1993; A **5**: 2792-2798.

[12] Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *J. Res. NBS* 1952; **49**: 409-436.

[13] Lipps FB, Hemler RS. A scale analysis of deep moist convection and some related numerical calculations. *J. Atmos. Sci.* 1982; **39**:2192–2210.

[14] Margolin LG, Smolarkiewicz PK, Wyszogrodzki AA. Implicit turbulence modeling for high Reynolds number flows. *J. Fluids Eng.* 2002; **124**:862–867.

[15] Marshall J, Hill C, Perelman L, Adcroft A. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. J. Geoph. Res. 1997 **102**(C3), 5733–5752.

[16] Prusa JM, Smolarkiewicz PK, Wyszogrodzki AA. Simulations of gravity wave induced turbulence using 512 PE CRAY T3E. *Int. J. Applied Math. Comp. Science*, 2001; **11**: No. 4, 101-115.

[17] Prusa JM, Smolarkiewicz PK. An all-scale anelastic model for geophysical flows: dynamic grid deformation. *J. Comput. Phys.* 2003; **190**:601–622.

[18] Saad Y, Schultz M, A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 1986; **7**: 856–869

[19] Saad Y, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Stat. Comput.* 1993; **14**: 461–469

[20] Skamarock WC, Smolarkiewicz PK, Klemp JB, Preconditioned conjugate-residual solvers for Helmholtz equations in nonhydrostatic models, *Mon. Wea. Rev.* **125**, (1997) 587–599.

[21] Smolarkiewicz PK, Pudykiewicz JA. A class of semi-Lagrangian approximations for fluids. *J. Atmos. Sci.* 1992; **49**:2082–2096.

[22] Smolarkiewicz PK, Margolin LG. On forward-in-time differencing for fluids: extension to a curvilinear framework. *Mon. Weather Rev.* 1993; **121**:1847–1859.

[23] Smolarkiewicz PK, Margolin LG, Variational solver for elliptic problems in atmospheric flows, *Applied Math. Comp. Sci.* **4**, (1994) 527–551.

[24] Smolarkiewicz PK, Grubišić V, and Margolin LG, On forward-in-time differencing for fluids: stopping criteria for iterative solutions of anelastic pressure equations, *Mon. Wea. Rev.*, **125**, 647-654 (1997).

[25] Smolarkiewicz PK, Margolin LG. MPDATA: A finite difference solver for geophysical flows. *J. Comput. Phys.* 1998; **140**:459–480.

[26] Smolarkiewicz PK, Margolin LG. Variational methods for elliptic problems in fluid models. *Proc. ECMWF Workshop on Developments in numerical methods for very high resolution global models* 5-7 June 2000; Reading, UK, ECMWF, 137–159.

[27] Smolarkiewicz PK, Margolin LG, Wyszogrodzki AA. A class of nonhydrostatic global models. *J. Atmos. Sci*. 2001; **58**:349–364.

[28] Smolarkiewicz PK, Prusa JM, VLES modeling of geophysical fluids with nonoscillatory forward-in-time schemes, *Int. J. Num. Meth. Fluids* 2002; **39**: 799-819.

[29] Smolarkiewicz PK, Prusa JM. Forward-in-Time Differencing for Fluids: Simulation of geophysical turbulence. In *Turbulent Flow Computation* Drikakis D, Guertz BJ (eds). Kluwer Academic Publishers, 2002; 207–240.

[30] Smolarkiewicz PK, Prusa JM. Toward mesh adaptivity for geophysical turbulence. *Int. J. Numer. Meth. Fluids* 2004, submitted.

[31] Smolarkiewicz PK, Prusa JM. Dynamic grid deformation: Continuous mapping approach. *ibid.*

[32] Strikwerda JC, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, 1989, pp. 386.

[33] Temperton C. Algorithms for the solution of cyclic tridiagonal systems. *J. Comput. Phys.* 1975; **19**:317–323.

[34] Thomas SJ, Girard C, Benoit R, Desgagné M, Pellerin P, A new adiabatic kernel for the MC2 model, *Atmos.-Ocean.* 1998 **36**:241–270.

[35] Thomas SJ, Girard C, Doms G, Schättler U, Semi-implicit scheme for the DWD Lokal-Modell, *Meteor. Atmos. Phys.* 2000 **73**:105–125.

[36] Thomas SJ, Hacker JP, Smolarkiewicz PK, Stull RB, Spectral preconditioners for nonhydrostatic atmospheric models, *Mon. Wea. Rev.*, 2003 **131**: 2464–2478.

[37] Wedi NP, Smolarkiewicz PK. Extending Gal-Chen & Somerville terrain-following coordinate transformation on time-dependent curvilinear boundaries. *J. Comput. Phys.* 2004; **193**:1–20.