Implementation of a Global Semi-Lagrangian Numerical Weather Prediction Model on Parallel Computers

Mikhail A. Tolstykh*

Institute of Numerical Mathematics RAS,
8 Gubkina st., 119991 Moscow, Russia

Key words: global Earth simulation

The numerical weather prediction (NWP) was always the field where high-performance computers are required. To make use of them, the program implementation of the model should be well suited for parallel architecture of these computers. In this paper, parallel implementation of the global 3D semi-Lagrangian NWP model developed in the Institute of Numerical Mathematics, Russian Academy of Sciences (INM RAS) is presented.

The approaches to parallelization described here can be also applied to the problems of incompressible computational fluid dynamics.

The governing equations of the model are essentially the Navier-Stokes-type equations on the rotating sphere with some additional simplifying assumptions (i.e. hydrostatic assumption). These equations for moist atmosphere can be written as

$$\frac{\partial \vec{\varphi}}{\partial t} + A\vec{\varphi} + \boldsymbol{B}\vec{\varphi} = \vec{\boldsymbol{F}}, \qquad (1)$$

where $\vec{\varphi} = \left(u, v, T, q, \ln \pi\right)^T$ is the vector of state of the system, u, v are the components of the horizontal velocity vector, T is the temperature, q is the specific humidity, and π is the surface pressure, A is the 3D advection operator, $\boldsymbol{B}$ represents all other terms of equations without forcing. In the forcing term $\vec{\boldsymbol{F}} = \left(F_u, F_v, F_T, F_q, 0\right)^T$, each component describes the sources and sinks of the corresponding quantity due to subgrid-scale processes (horizontal and vertical diffusion, shortwave and longwave radiation, deep and shallow convection, planetary boundary layer, gravity wave drag).

The forcing term $\vec{\boldsymbol{F}}$ can be computed independently for each point of the horizontal grid, since the computations of this term involve only the points from given vertical column of the grid. The main difficulty in parallelization of atmospheric models lies in the "dynamics" calculation, i.e. time stepping algorithm for the discretized Eq. (1). This is because atmospheric models normally use the semi-implicit treatment of fast gravity waves to increase the time-step. This results in a set of discrete Helmholtz-type equations on the sphere to be solved at each time step.

The semi-Lagrangian method (backward characteristic method) for advection allows to use the time step, which is several times greater than determined by the CFL condition. In this case, the advection operator vanishes in the discretized form of Eq. (1), as the material derivative is discretized as the derivative along the trajectory (see [1] for details). The model under consideration is presented in detail in [2], and numerical methods used for horizontal discretization are given in [3]. The distinct features of this model are the application of the fourth-order compact finite differences on the unstaggered grid for approximation of nonadvective terms combined with the use of the vertical component of the absolute vorticity and the horizontal divergence as prognostic variables.

The system (1) without forcing is discretized using the two-time level scheme with extrapolation of wind and the nonlinear terms to the time level *n+1/2* as

* E-mail: tolstykh@inm.ras.ru
The work was supported by RFBR under Grant No. 01-05-64582.

$$\frac{\vec{\varphi}^{n+1}-\vec{\varphi}_*^{n}}{\Delta t}+\frac{1}{2}\left(\mathbf{L}\vec{\varphi}^{n+1}+\mathbf{L}\vec{\varphi}_*^{n}\right)+\frac{1}{2}\left(\left(\mathbf{N}\vec{\varphi}\right)^{n+1/2}+\left(\mathbf{N}\vec{\varphi}\right)_*^{n+1/2}\right)=0, \quad (2)$$

where **L** is the linear part of the operator ***B***, which is responsible for fast gravity waves, **L**+**N**=***B***, $(\cdot)^{n+1}$ denotes the value of the corresponding equation terms at the arrival point of the trajectory at the new time step, $(\cdot)_*^{n}$ - the value at the corresponding departure point (which is determined by the 3D trajectory). Arrival points are always points of the grid, while the values at departure points are obtained by interpolation using the surrounding points of the grid.

One solves the implicit part of system (2) with the help of a direct solver. To do this, the corresponding part of system (2) is reduced to the single discrete 3D Helmholtz-like equation with respect to the single variable *D*.

$$D^{n+1}-\left(\Delta t\right)^2\mathbf{G}\nabla^2 D^{n+1}=R, \quad (3)$$

where *D* is the horizontal divergence on the model σ-coordinate surface, **G** is a *K* x *K* matrix (*K* - is the number of vertical levels in the model), ∇^2 is the discrete analogue of the horizontal Laplace operator in spherical coordinates, *R* combines all known quantities from system (2) at time levels *n* and *n-1*. The periodical boundary conditions in longitude are imposed on (3) as well as on system (2). The diagonalizing transform applied to matrix **G** allows to uncouple (3) in the vertical and solve it independently for each vertical level by a direct method involving Fourier transforms in longitudinal direction and block-tridiagonal Gaussian elimination in the meridional direction. The necessity to use 2x2 block-tridiagonal inversion is originated from the application of compact finite differences.

Let us consider the computational structure of the model. First the forcing term $\vec{\boldsymbol{F}}$ is computed (Eq. 1) using the values of state variables at time level *n*. This includes the calculation of prognostic variables tendencies due to parameterizations of subgrid-scale processes (solar radiation etc.) One should note that these computations represent 61 % of all computations in terms of the CPU time and can be done in parallel for all points of the horizontal grid. Then the terms **L** at time level *n* and **N** at time level *n+1/2* from (3) are calculated. These calculations for a given grid point (*i,j,k*) require the values from the domain [*i-4:i+4*]x[*j-4:j+4*]x[*1:K*]. The relatively large size of the horizontal stencil is caused by the use of fourth-order compact finite differences to calculate the gradient and divergence operators. They are implemented with the approximate inversion of the corresponding tridiagonal operators.

The next step is to find departure points and interpolate the calculated terms to them. This part is responsible for about 30 % of CPU time per time step. Typical half-width of the dependency domain in our model for the horizontal resolution of 1.5 degrees is 5 points in each horizontal direction.

We restrict ourselves by the 1D partitioning of the computational domain so that each processor performs computations in some band of latitudes (Fig. 1). Note that unlike most Eulerian methods, in the case of 2D partitioning, one would need 8 (and not 4) data exchanges between processors, involving also "corner" processors. In future, we will use OpenMP to increase the degree of parallelism.

Once interpolations and direct fast Fourier transforms in longitude are finished, we have to redistribute data for Fourier space calculations. The calculations for each longitudinal wavenumber require the values of all latitudes (Fig. 1), so we need to provide each processor with the corresponding data using data transposition.

The computations in Fourier space involve the calculation of the RHS and solution of Helmholtz-type equation (3), implicit horizontal fourth-order diffusion and calculation of horizontal velocity components from divergence and vorticity. These computations are performed in parallel for each longitudinal wavenumber. After Fourier-space calculations we again transpose data since we need all longitudinal wavenumbers to perform inverse Fourier transforms.

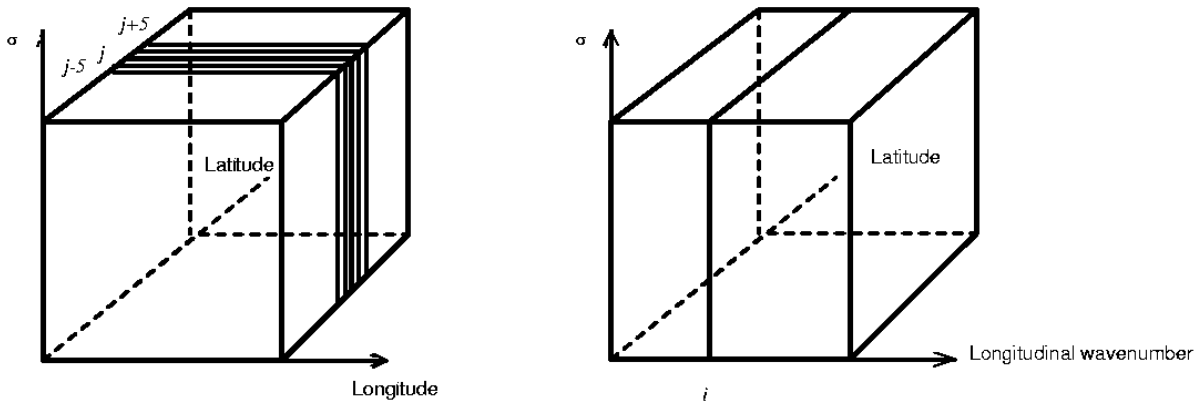

Figure 1. Domain partition in the grid-point and Fourier space computations.

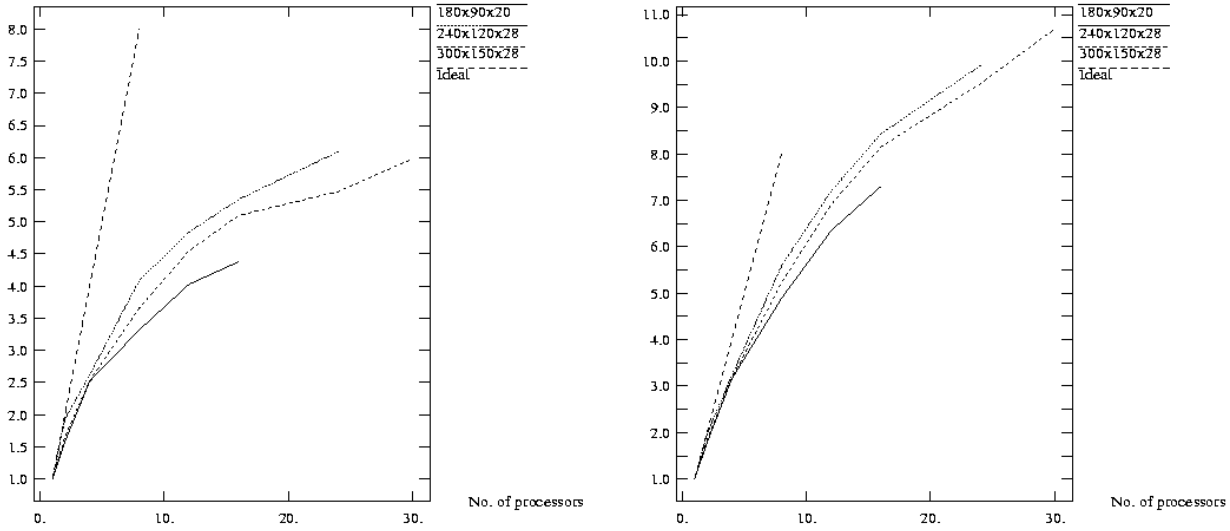

Figure 2. Parallel speedup of semi-Lagrangian NWP model: speedup of the dynamics (left), extrapolated speedup of the whole model with the forcing term (right) for different resolutions of the model

The MPI (Message Passing Interface) is used to handle communications between processors. We use the single program -- multiple data (SPMD) approach with the master process also carrying out I/O and some additional computations. Whenever possible, asynchronous communications are used that permit to overlap the computations and data exchanges.

The benchmarking computations were carried out on the MBC 1000M system installed at Joint Supercomputer Center in Moscow. It consists of 384 biprocessor Alpha21264A 667 MHz nodes connected via Myrinet network having peak output of 210 Mbytes/s and latency time 14 μs over MPI. The peak performance of MBC 1000M is 1 TFlops. Three resolution of the model were tested: 2 degrees in longitude and latitude with 20 vertical levels (this corresponds to the dimensions 180x90x20 in longitude, latitude and vertical coordinate, respectively), 1.5 degrees with 20 levels (240x120x28), and 1.5 degrees with 28 levels (300x150x28). The parallel speedup of the dynamics of the semi-Lagrangian NWP model on MBC 1000M systems is presented in Fig. 2 (left). The right panel of Fig. 2 depicts the theoretical parallel speedup calculated for the whole model with forcing terms.

Finally, one can note that modern atmospheric models apply more and more sophisticated algorithms to calculate the components of forcing term in Eq. (1). This means that parallel efficiency of the models will be further improved without additional efforts, since forcing terms computations do not require additional exchanges between processors.

It is expected that the results for other parallel systems will be available at the time of conference.

References

1. *A. Staniforth and J.Côté,* Semi-Lagrangian integration schemes for atmospheric models - A review. *Mon. Weather Rev.* 119 (1991), 2206-2223.
2. *M.A. Tolstykh,* Semi-Lagrangian high-resolution model of the atmosphere for numerical weather prediction, *Russian Meteorology and Hydrology*, N 4, 2001, 1-9.
3. *Mikhail A. Tolstykh,* Vorticity-divergence semi-Lagrangian shallow-water model on the sphere based on compact finite differences, *J. Comput. Phys.* 179 (2002), 180-200.