DOI: 10.14529/jsfi220102

A Supercomputer-Based Modeling System for Short-Term Prediction of Urban Surface Air Quality

Alexander V. Starchenko[1], ***Evgeniy A. Danilkin***[1], ***Sergei A. Prokhanov***[1], ***Lubov I. Kizhner***[1], ***Elena A. Shelmina***[1,2]

This paper proposes a mathematical model and an effective supercomputer-based numerical method for short-term prediction of extreme meteorological conditions and atmospheric air quality over limited stretches of land encompassing large population centers. The mathematical model includes a pollutant transport model with a reduced chemical mechanism and a non-hydrostatic mesoscale meteorological model with a modern moisture microphysics parametrization scheme. The numerical method relies on the use of the finite volume method and semi-implicit difference schemes of the second order of approximation, which are solved using the TDMA method with a linear dependence of the number of arithmetic operations on the size of the grid. This property of the numerical method ensures high efficiency when parallelized: not less than 70% when using up to 256 computing cores with a horizontal grid size of 0.5–1.0 *km*. Development of parallel programs was carried out using the Message Passing Interface parallel programming technology, two-dimensional decomposition of the grid area along horizontal (west to east and south to north) directions, and introduction of additional fictitious grid nodes along the perimeter of the decomposition subdomains.

Keywords: parallel computations, numerical weather prediction, mesoscale models, urban air quality, MPI.

Introduction

Currently, protection of the environment is becoming one of the most important tasks of science, interest in which is stimulated by the ever-increasing pace of technological progress around the world. Intensive industrial development and the resulting increase in industrial emissions of air pollutants are already starting to have a noticeable effect on the preservation of ecological balance in many regions of our planet. One of the most pressing issues is that of atmospheric air quality deterioration. The significance of this problem is primarily due to the fact that the atmosphere is one of the main vital elements of the environment. Moreover, the quality of the air in the lower part of the atmosphere, the part that is in contact with the Earth's surface, is of particular importance, since it is where the bulk of plant and animal life, including humans, reside. Presently, the deteriorating quality of the air is of particular concern due to changes in its chemical and aerosol composition caused by anthropogenic impact – emissions from industrial enterprises and transport [1].

Recently, mathematical modeling methods have been successfully used to monitor and forecast the ecological state of the urban atmosphere along with instrumental surveys. However, the complexity and interconnectedness of the processes of propagation, dispersion and chemical transformation of pollutant components occurring in the turbulent atmospheric boundary layer make air quality forecasting models cumbersome in mathematical notation and very demanding on computational resources [2]. In addition, reliable calculation of the vertical transport of pollutants, especially within the conditions of the convective boundary layer, requires the use of modern algebraic turbulence models to determine turbulent flows of momentum, mass and heat, which in itself is not an easy task. Thus, the problem of human interaction with the environment

[1]National Research Tomsk State University, Tomsk, Russian Federation
[2]Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russian Federation

currently represents a new and actively developing field of application of mathematical modeling methods.

Currently, several modeling systems have been created and are functioning in the world, operating in real time, which rely on models of the atmospheric boundary layer and pollutant transport. The Enviro-HIRLAM [3] is designed as a fully integrated online system for numerical weather prediction and pollutant transport modeling for study and prediction of meteorological, chemical and biological weather. The integrated modeling system was developed by the Danish Meteorological Institute (DMI) in collaboration with other researchers and is used as the base system of the HIRLAM consortium. Development of the model was started at DMI more than ten years ago and it is now used by several countries. The current version of Enviro-HIRLAM [4] is based on the master version of HIRLAM 7.2 with a more complex and efficient chemistry scheme and aerosol dynamics modules based on a multimode aerosol-radiation and aerosol-cloud microphysics interaction approach. An immediate emergency report model has been developed. The system is optimized for use on vector supercomputers. For example, at the Finnish Meteorological Institute, calculations are carried out on the FMI Cray XC30 supercomputer, which includes 3420 computing units with peak performance of 70 Tflops.

ADMS-Urban is a modeling system developed by Cambridge Environmental Research Consultants with support from the UK Meteorological Office [5]. The system is used to monitor air quality and study complex situations in cities on highways, and in both non-metropolitan areas and large industrial centers. Currently, this model is used in cities in Europe and Asia, including China, as well as in the United States to monitor air pollution levels and ensure their compliance with modern standards. In the UK, over 70 local governments, including London, use ADMS-Urban for monitoring and evaluation, as well as for modeling possible consequences of industrial developments, such as the expansion of airports or construction of highways. The ADMS-Urban Regional Model system runs using the ARCHER UK National Supercomputing Service.

The EURAD-IM (EURopean Air pollution Dispersion-Inverse Model) system [6], created at the University of Cologne Institute for Geophysics and Meteorology, simulates the physical, chemical and dynamic processes that affect the emission, formation, transport and deposition of atmospheric pollutants, and determines the distribution of concentrations in the troposphere over Europe, as well as the dry and wet deposition of air pollutants. This system consists of five submodules. Meteorological data calculations are carried out using the Weather Research and Forecast model [7], operations related to chemical transformations and pollutant transport are carried out using the EURAD-IM model, including the Regional Atmospheric Chemistry Mechanism (RACM) [8], and emission-related operations are carried out using the EURAD Emission Model (EEM). This system is used to monitor the formation of ozone and other photooxidants. The system uses the JURECA supercomputer at the Jülich Supercomputing Centre for Atmospheric air quality calculations.

The aim of the study is to develop efficient parallel numerical methods for a short-term high-resolution ($\sim 1\ km$) modeling system capable of forecasting extreme meteorological conditions and determining urban air quality, aimed at the use of a distributed memory supercomputer.

The article is organized as follows. Section 1 is devoted to mathematical state of the problem. In Section 2 we present a description of numerical method and its parallelization. Conclusion summarizes the study.

1. Description of the Mathematical State of the Problem

To calculate the concentration of pollutant components with regard to the chemical interactions between them, the proposed atmospheric air quality modeling system uses an Eulerian turbulent diffusion model, which includes non-stationary equations describing advection, turbulent diffusion and chemical reactions:

$$\begin{gathered}\frac{\partial \rho C_i}{\partial t}+\frac{\partial \rho u C_i}{\partial x}+\frac{\partial \rho v C_i}{\partial y}+\frac{\partial \rho w C_i}{\partial z}= \\ =\frac{\partial}{\partial x}\left(K_{xy}\frac{\partial C_i}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{xy}\frac{\partial C_i}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_Z\frac{\partial C_i}{\partial z}\right)-\sigma_i C_i+S_i+R_i, \\ i=1\,,..,\,n_s,\end{gathered} \tag{1}$$

$$-L_x/2 \le x \le L_x/2, -L_y/2 \le y \le L_y/2, h(x,y) \le z \le H.$$

Here, C_i is the dimensionless concentration of the i^{th} pollutant component; u and v are horizontal wind speed vector components; w is the vertical velocity component of the pollutant; ρ is the air density; K_{xy} and K_Z are diffusion coefficients; S_i is the intensity of pollutant component emission into the atmosphere from both elevated point and linear land-based sources; R_i describes the formation and transformation of substances through chemical and photochemical reactions involving pollutant components; σ_i is the rate of wet deposition of pollutants through precipitation; n_s is the number of chemical components of the pollutant, the concentration of which to be determined; x and y are horizontal coordinates, the Ox axis is directed to the east, Oy – to the north; z is the vertical coordinate; t is time. The computational domain is parallelepiped-shaped, L_x and L_y are the horizontal dimensions, H is its height, and $h(x,y)$ is the elevation above sea level. The transport model (1) predicts the chemical composition of surface air in an area of 50×50 *km* with a resolution of 500 *m*.

Background component concentration values are used as initial conditions; transport process and chemical reactions are spun up through a 24-hour period before the start of numerical surveys. The upper boundary of the examined area is set at 1 *km*, and the horizontal dimensions $L_x = L_y = 50$ *km*. At the upper boundary, simple gradient conditions for concentrations are applied. At lateral boundaries, computational conditions of the so-called "radiation" type are used [9], providing computational stability of calculations in an open-border area.

At the lower boundary of the examined area, in the center of which the city is located, the conditions of dry deposition of the pollutant due to the resistance of the viscous sublayer, aerodynamic resistance and resistance caused by the distribution of vegetation are set [10].

Primary air pollutant emission sources taken into account are ground sources – motor vehicles – and elevated sources – factory chimneys. Emission parameters for elevated sources were assumed to be constant. For linear land-based sources, the standardized emission rate was set according to the following law:

$$I_{Vehicle}(t)=\begin{cases}0.05+0.95\sin\left(\pi(t-6)/18\right), t\in[6,24],\\ 0.05, t\notin[6,24].\end{cases} \tag{2}$$

Here t is the local time in hours. According to this formula, the maximum intensity of vehicle emissions is observed at about 15:00, local time. From 00:00 to 06:00, the intensity of vehicle emissions is minimal. The integral emission intensity of linear land-based sources and elevated sources per day was calculated based on annual emissions of main pollutants in the Tomsk region for 2019 [11].

In this paper, R_i in (1) is estimated using a kinetic scheme based on two well-tested reduced chemical reaction mechanisms [12, 13]. The General Reaction Set semi-empirical chemical reaction mechanism [12] provides a compact description of the formation of secondary air pollutants ($PM2.5$, $PM10$, RP, H_2O_2, etc). A more detailed representation of tropospheric ozone generation through photochemical reactions was taken from the reduced mechanism [13].

To calculate the speed vector components (u, v, w) and the turbulent diffusion coefficients, (1) uses a non-hydrostatic mesoscale meteorological model [14, 15]. This model uses a system of equations of hydrodynamics and heat and mass transfer in the troposphere, and an upper soil layer heat equation. The meteorological model predicts the wind speed components and temperature and humidity characteristics in the boundary layer of the atmosphere at 50 vertical levels (up to 10 000 m) above a territory of 150 × 150 km and an embedded area with a base of 50 × 50 km (the grid step is 1 km, the grid is centered on the city) for 24 hours. The basic equations of the mathematical representation of the model are presented below [16]:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0, \tag{3}$$

$$\begin{gathered}\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = \\ -\frac{\partial p}{\partial x} + \rho f v + \frac{\partial}{\partial x}\left(K_{xy}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{xy}\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_Z^m\frac{\partial u}{\partial z}\right),\end{gathered} \tag{4}$$

$$\begin{gathered}\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = \\ -\frac{\partial p}{\partial y} - \rho f u + \frac{\partial}{\partial x}\left(K_{xy}\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{xy}\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_Z^m\frac{\partial v}{\partial z}\right),\end{gathered} \tag{5}$$

$$\begin{gathered}\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = \\ -\frac{\partial p}{\partial z} - \rho g + \frac{\partial}{\partial x}\left(K_{xy}\frac{\partial w}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{xy}\frac{\partial w}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_Z^m\frac{\partial w}{\partial z}\right).\end{gathered} \tag{6}$$

Here, t is time, u, v and w are longitudinal, transverse and vertical components of the average wind speed vector in the direction of the Cartesian coordinates x, y, z, respectively, f is the Coriolis parameter, K_{xy} is the horizontal diffusion coefficient, is the momentum vertical diffusion coefficient, g is gravitational acceleration, and p is pressure.

$$\begin{gathered}\rho\left(\frac{\partial \theta}{\partial t} + u\frac{\partial \theta}{\partial x} + v\frac{\partial \theta}{\partial y} + w\frac{\partial \theta}{\partial z}\right) = \\ \frac{\partial}{\partial x}\left(K_{xy}\frac{\partial \theta}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{xy}\frac{\partial \theta}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_Z^h\frac{\partial \theta}{\partial z}\right) + \frac{\theta}{c_p T}\left(Q_{rad} - \rho L_w \Phi\right).\end{gathered} \tag{7}$$

Here, T is the absolute temperature, θ is the potential temperature, $\theta = T(p_0/p)^{R_0/c_p}$, c_p is the air heat capacity at constant pressure, $p_0 = 101300\ N/m^2$, R_0 – the gas constant, Q_{rad} is heating (or cooling) of the atmosphere due to long- and shortwave radiation heat fluxes propagating in a humid atmosphere, $\rho L_w \Phi$ is the temperature change caused by phase transitions of moisture in the atmosphere, K_z^h is the heat and moisture vertical diffusion coefficient, L_w is vaporization heat, and q_V is the relative density of atmospheric steam.

$$p = \rho RT, R = R_0 \left[\frac{1 - q_V}{M_{air}} + \frac{q_V}{M_{H_2O}} \right]. \tag{8}$$

Simulation of water moisture phase transformation processes in the atmosphere in this study is carried out using the WSM6 6-class moisture microphysics scheme [17]. This scheme simulates the processes that occur between the six states of atmospheric moisture (water vapor, cloud water content, rain, ice particles, snow and graupel (hail)). For each of the parameters characterizing the state of moisture in the atmosphere, a transport equation is used, which includes, along with advective transport, various parameterizations of physical processes leading to changes in the phase state of the aforementioned forms of moisture.

To close the system of equations (1), (3)–(8), a turbulence model is used, including an equation for kinetic energy [18], as well as algebraic relations for determining turbulent diffusion coefficients [19]. The horizontal diffusion coefficient is calculated using the Smagorinsky formula [20].

In addition, the model takes into account the following: heat transfer through shortwave and longwave radiation in the examined layer of the atmosphere with regard to clear-sky scattering and attenuation, water vapor absorption, absorption and reflection by clouds [21, 22]; turbulent momentum, heat and moisture exchange with the underlying surface [23, 24]; temperature change in the upper layer of the surface and humidity at the “air-underlying surface” boundary.

The initialization of the mesoscale numerical weather prediction model and its provision with lateral boundary conditions is carried out based on the results of numerical weather forecasting by the Hydrometeorological Centre of Russia’s SL-AV operational global model [25].

2. Numerical Method and its Parallelization

Thus, the basis of the mathematical formulation of the problem in question is a generalized inhomogeneous differential equation of convective diffusion transport:

$$\begin{gathered}\frac{\partial \rho\Phi}{\partial t} + \frac{\partial \rho u\Phi}{\partial x} + \frac{\partial \rho v\Phi}{\partial y} + \frac{\partial \rho w\Phi}{\partial z} = \frac{\partial}{\partial x}\left(K_{xy}\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_{xy}\frac{\partial \Phi}{\partial y}\right) + \\ + \frac{\partial}{\partial z}\left(K_z^{\Phi}\frac{\partial \Phi}{\partial z}\right) + S_\Phi.\end{gathered} \tag{9}$$

Here, Φ represents different sought-for functions in different equations (1), (3)–(7). To account for in relief of the underlying surface $h(x, y)$, equation (9) is brought to the following form:

$$x' = x; y' = y; z' = \frac{z - h(x, y)}{H - h(x, y)} H. \tag{10}$$

As a result of this transformation, mixed derivatives appear. Here, H is the height of the computational domain. Equations such as (9) are solved using the finite volume method using grids with uniform spacings along the horizontal x and y axes and spacings along the vertical z axis getting progressively finer towards the Earth’s surface. The lower boundary is 10 m above the surface. Differential equation (9) is approximated using the finite volume method with second-order spatial variable approximation and explicit-implicit time approximations [26], which also provide second-order accuracy in time.

$$\begin{gathered}\Phi_h^{n+1} = \Phi_h^n + \frac{\Delta t_n}{2}\left(3L_h\left(\Phi_h^n\right) - L_h\left(\Phi_h^{n-1}\right)\right) + \\ + \frac{\Delta t_n}{2}\left(\Lambda_h\left(\Phi_h^{n+1}\right) + \Lambda_h\left(\Phi_h^n\right)\right) + \frac{\Delta t_n}{2}\left(3S_\Phi\left(\Phi_h^n\right) - S_\Phi\left(\Phi_h^{n-1}\right)\right),\end{gathered} \tag{11}$$

where $\Phi_h^n = \left\{\Phi_{i,j,k}^n\right\}$ is the grid function of the scalar Φ for which the differential equation (9) is given; n is the time layer number; $i = 1,\ldots,Nx$ is the number of grid nodes along the x axis; $j = 1,\ldots,Ny$ is the number of grid nodes along the y axis; $k = 1,\ldots,Nz$ is the number of grid nodes along the z axis; L_h is the finite-volume analogue of the convective-diffusive operator in equations (9) except for the vertical diffusion along the z axis, Λ_h is the difference analogue of the differential operator of vertical diffusion $\frac{\partial}{\partial z}\left(K_z\frac{\partial\Phi}{\partial z}\right)$, $S_\Phi(\Phi)$ is the source terms of equation (9). Implicit approximation for vertical diffusion transport, which is important in the boundary layer of the atmosphere, allows to avoid a more rigid time restriction on the integration step. When approximating the convective terms of equation (9), Van Leer's monotonized linear upstream schemes are used [27]. For diffusion terms, ordinary approximations of second-order accuracy are used. These approximations result in a formation of a difference scheme, in which values of the grid function $\left\{\Phi_{i,j,k}^{n+1}\right\}$ on a new time layer $(n+1)$ can be calculated using the tridiagonal matrix algorithm [28] independently along vertical grid lines. As a result of using this method of forming the difference scheme (11), the number of arithmetic operations depends linearly on the grid size and conditions are created for perfectly parallel processing.

To match the finite-difference values of the speed and pressure vector field at each time step, the predictor-corrector scheme is used. Its main idea is that, firstly, the grid values of the speed vector components are predicted using difference schemes of the form (11) for known grid pressure function p_h^n values at the n^{th} time layer. Then an elliptic difference equation is solved for pressure correction $p'_h = p_h^{n+1} - p_h^n$ and the intermediate speed and pressure fields are corrected. This operation is carried out with the requirement that the corrected values of the speed components exactly satisfy the difference analogue of the continuity equation (3). The described scheme includes the following sequence of operations at each n^{th} time step:

1. Approximate speed values $\tilde{u}_h^{n+1}$, $\tilde{v}_h^{n+1}$, $\tilde{w}_h^{n+1}$ are calculated using difference formulas of the type (11) for a known pressure field p_h^n.
2. An equation of the following form for pressure correction p'_h is solved

$$\begin{gathered}ap_{i,j,k}\left(p'\right)_{i,j,k} - ab_{i,j,k}\left(p'\right)_{i,j,k-1} - at_{i,j,k}\left(p'\right)_{i,j,k+1}^{l+1} - ae_{i,j,k}\left(p'\right)_{i+1,j,k}^{l} - \\ -an_{i,j,k}\left(p'\right)_{i,j+1,k}^{l} - aw_{i,j,k}\left(p'\right)_{i-1,j,k}^{l+1} - as_{i,j,k}\left(p'\right)_{i,j-1,k}^{l+1} = b_{i,j,k}({p'}_h), \\ i = \overline{1,Nx}; j = \overline{1,Ny}; k = \overline{1,Nz}.\end{gathered} \tag{12}$$

3. A new pressure field $p_h^{n+1} = p_h^n + p'_h$ is determined.
4. Speed components u_h^{n+1}, v_h^{n+1}, w_h^{n+1} are calculated using speed values $\tilde{u}_h^{n+1}$, $\tilde{v}_h^{n+1}$, $\tilde{w}_h^{n+1}$ and pressure correction value p'_h.

In view of this and the results of conducted computational experiments, a decision was made to calculate pressure correction p'_h using the Gauss-Seidel line by line iteration method [29] with red and black grid node arrangement for each horizontal level k and implicit representation of sought-for values of the grid function p'_h in the nodes $(i, j, k+1)$, (i, j, k), $(i, j, k-1)$, i.e., along the vertical grid lines:

$$\begin{gathered}
-ab_{i,j,k}\left(p'\right)^{l+1}_{i,j,k-1} + ap_{i,j,k}\left(p'\right)^{l+1}_{i,j,k} - at_{i,j,k}\left(p'\right)^{l+1}_{i,j,k+1} = \\
= ae_{i,j,k}\left(p'\right)^{l}_{i+1,j,k} + an_{i,j,k}\left(p'\right)^{l}_{i,j+1,k} + \\
+ aw_{i,j,k}\left(p'\right)^{l+1}_{i-1,j,k} + as_{i,j,k}\left(p'\right)^{l+1}_{i,j-1,k} + b_{i,j,k}(({p'}_h)^l), \\
i = \overline{1, Nx}; j = \overline{1, Ny}; k = \overline{1, Nz},
\end{gathered} \tag{13}$$

l is the number of iteration.

Thus, the main complexity of the program for calculating pollutant concentrations and meteorological fields lies in solving convective-diffusion equations of the form (11) and elliptic equations for pressure field correction (13).

2.1. The Choice of a Parallel Programming Technology

To find the most suitable parallel programming technology for solving the numerical prediction problems in question and implement the chosen numerical method, test calculations were carried out. As one of the subtasks for testing, a mathematical statement with mixed boundary conditions was chosen for one non-stationary inhomogeneous convection diffusion equation (9), the numerical solution of which used difference scheme (11). Open MultiProcessing (OpenMP), Message Passing Interface (MPI), Open Accelerators (OpenACC) and Compute Unified Device Architecture (CUDA) parallel programming technologies [30] were used for test calculations. The calculations used a $256 \times 256 \times 32$ grid and 5000 time steps. Calculations were carried out using Tomsk State University's Cyberia cluster. Each node of the cluster has the following specifications: 48Gb RAM, 2×Intel Xeon X5670 (2.93GHz), 1×GPU NVIDIA GeForce RTX 2080 Ti. Average sequential program runtime for one node is 1733.0 seconds.

When using OpenMP, OpenACC and CUDA technologies, only one node of the cluster with shared RAM for two CPUs and one GPU can be used. For the test case in question and the chosen numerical method, OpenMP programming technology was applied by distributing a set of $Nx \times Ny$ independent subtasks of solving tridiagonal systems between independent parallel threads. To do this, the OpenMP parallel directives were used. As a result of the tests, a speedup of 10.5 was achieved (Fig. 1). It was also found that for the case in question, parallelization of two external loops (i and j) yields the same speedup results as parallelization of just the i loop. Using static, dynamic and guided schedules also does not yield a significant speedup compared to the default iteration distribution.

Attempts were also made to use hybrid CPU/GPU technologies – OpenACC and CUDA – to solve the test problem (9) using a graphics card. When using OpenACC, similarly to OpenMP, compiler directives were used to assign independent subtasks i and j, consisting of solving tridiagonal systems of linear equations (11), to GPU cores. This approach yielded a runtime of 20.4 seconds.

When using CUDA technology, an algorithm for solving a tridiagonal system of linear equations dependent on the indices i and j, was also chosen as the compute kernel. These systems were distributed among GPU cores and solved for each time step with subsequent synchronization of computing processes. The use of CUDA made it possible to obtain a solution to the problem in 16.6 seconds.

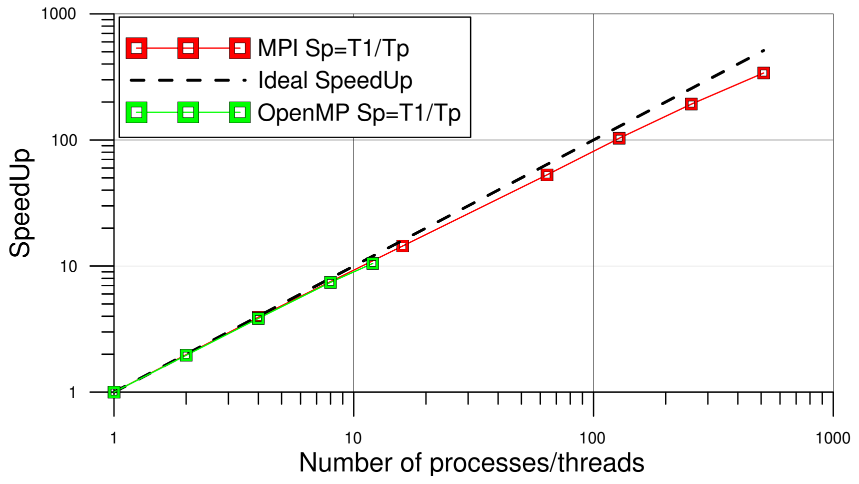


Figure 1. Comparison of execution speedup of parallel programs developed using OpenMP and MPI, tested using TSU's Cyberia cluster

Thus, when the amount of data transmitted between software modules or the number of equations to be solved (9) is small, the use of GPUs for implementation of parallel programming technologies shows more promise.

MPI technology is also often considered for development of parallel programs on multiprocessor distributed-memory systems. For the problem in question (9) and the chosen numerical method (11), this technology was used in combination with a two-dimensional decomposition of the grid area along the grid lines i and j (in the directions of the x and y axes). Blocking communication functions MPI_Sendrecv were used to transmit messages between processes. Replacing these functions with non-blocking communication functions MPI_Isend and MPI_Irecv yielded no more than a 5% decrease in the total execution time of the test task, therefore it was not considered further. To ensure uniformity of parallel computations, two rows of dummy cells were used along the perimeter of each subdomain of the two-dimensional decomposition, due to the use of Van Leer's monotonized upstream scheme [27]. Tridiagonal systems of linear equations (11) were solved simultaneously in each subdomain. With this method of organizing parallel computing using technology on one node of the Cyberia cluster, a speedup of 11.4 was achieved when solving the test problem (9). Note that when performing parallel calculations using the same program on 128 processes, a speedup was obtained by 103 times (average calculation time 16.8 sec), and by 512 – 337 times (average time 5.1 sec).

When developing a parallel method for solving equations (13), it should be noted that the iterative scheme corresponds well to the chosen parallel algorithm for solving transport equations (11) and allows the use of two-dimensional decomposition and asynchronous relaxation [29] with a minimum amount of data exchange to synchronize parallel computations in decomposition subdomains. Unfortunately, at each iteration, such data exchanges must be performed, which makes it relevant to use rapidly converging iterative methods with a minimum number of iterations when they are implemented in parallel.

Thus, if the program code contains several software modules with a large amount of information transmitted between them during computations, which is a property of numerical implementation of mathematical models with a large number of equations (more than ten) of the form (9) and additional algebraic relations, then it is advisable to use the Message Passing Interface technology.

2.2. Parallel Implementation of the Numerical Method for TSUNM3

In view of the results of the computational experiments presented above, the working version of the TSUNM3 model of numerical weather prediction (3)–(8) was parallelized using 2D decomposition and MPI technology. Table 1 shows the resulting speedup values (calculated as the ratio of the execution time of the serial version of the code to the execution time of the parallel program) for the parallel program when performing weather forecast computations for the local research area on two grids for two days: $96 \times 96 \times 50$ nodes (Grid1) and $192 \times 192 \times 50$ nodes (Grid2). Since the plan is to use the TSUNM3 mesoscale meteorological model with a horizontal resolution of 1–2 km, Grid1 corresponds to an industrial area with a large number of enterprises, transport hubs and a large ($\sim$ million residents) population center at the center of a 100–200 km – wide research area. Grid2 can cover an area of 200–400 km and can be used for short-term weather prediction in an administrative area with several large population centers and transport hubs. For the selected grid sizes, mesoscale model calculations were performed on the TSU Cyberia cluster.

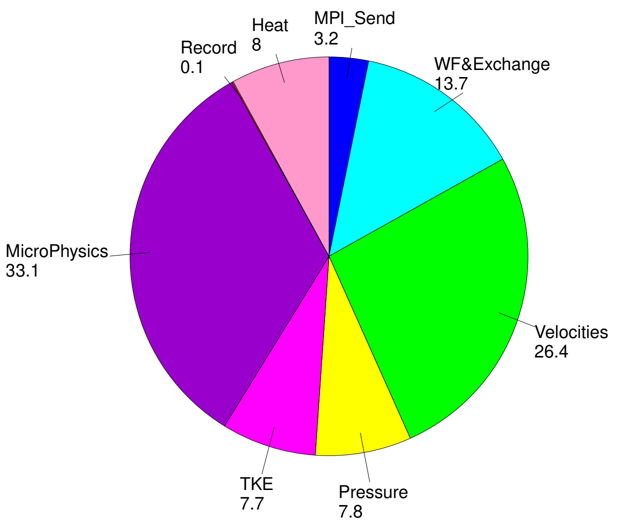
Table 1. TSUNM3 parallel program speedup values for various grids at different process quantities

Processes	4	9	16	36	64	144	256
Grid1	3.9	8.6	15.2	33.8	59.0	119.1	184.7
Grid2	3.9	8.5	14.8	32.1	58.1	129.2	218.8

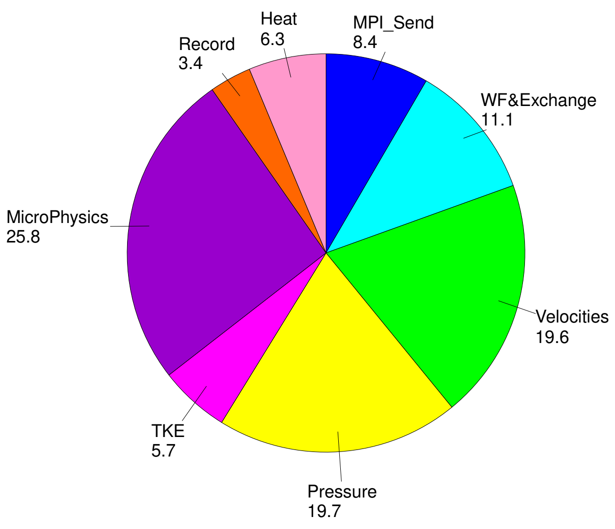
The speedup values for parallel computing programs with an increasing number of processes presented in Tab. 1 show good supercomputer resource utilization efficiency. With the results obtained, it is possible to carry out predictive calculations for the next day in 12 minutes of the program's runtime for Grid1 and 39 minutes for Grid2 using 256 cores of the TSU Cyberia supercomputer. Note that in the calculations, we used a version of the program in which calculations are performed with single precision, which makes it possible to reduce the computation time and the amount of data transmitted between computing nodes.

To clarify the results presented in Tab. 1, diagrams of the relative (in %) time costs of the work of each of the main blocks were constructed and the speedup values of the execution of each of these blocks for the considered number of processes were calculated. The main blocks of the program are: block for calculating the interaction with the underlying surface (friction, heat and mass transfer) and the exchange of values – WF&Exchange; velocity component calculation block – Velocities; block for calculating the hydrostatic and non-hydrostatic parts of pressure – Pressure; block for calculating turbulent kinetic energy and turbulent diffusion – TKE; moisture microphysics calculation block – MicroPhysics; block for solving the equation of heat transfer in the atmosphere – Heat; block for assembling distributed data and writing it to disk – Record. In these blocks, except for the Record block, only the computation time was measured. We also measured the total time required for interprocessor data transfer in 2D decomposition to perform parallel computing (MPI_Send).

Figure 2 shows that a large share in the total amount of computational costs is the execution of blocks for calculating the velocity components and moisture microphysics parameters. This is understandable, because three equations of the form (9) are solved in the first block, and four in the second. That is, on average, the numerical solution of one equation (9) by method (11) takes



(a)



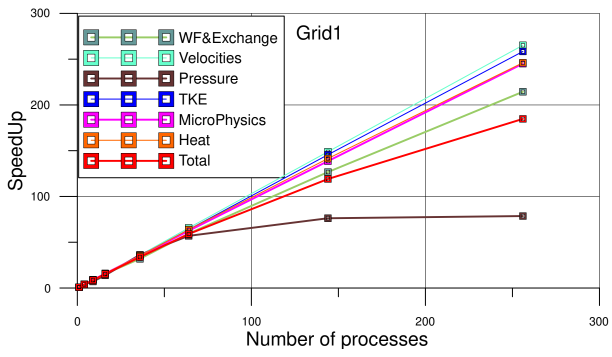
(b)

Figure 2. Diagrams of the relative time spent on the execution of various blocks of the program. (a) is the launch of the program on 36 processes, (b) – on 256. The calculations were performed on the grid Grid1

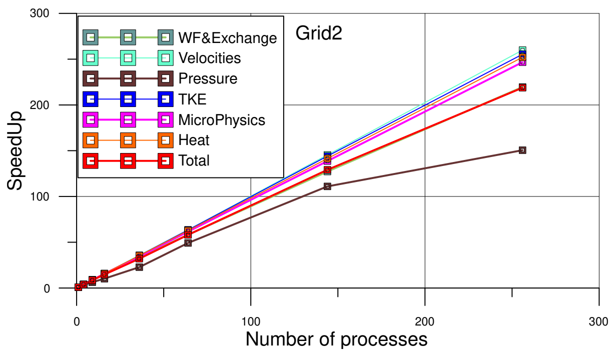
about 8–9% of the total computational time when using up to 36 processes. This is confirmed by the time spent on the implementation of blocks for the numerical solution of the heat transfer equation and the calculation of turbulent characteristics (Fig. 2), where one equation of the form (9) is solved. Note that when solving finite-difference equations of the form (11) numerically, due to the nonlinearity (advection/turbulent diffusion) of equation (9) and the presence of time-consuming source terms (especially for the MicroPhysics, Velocities, Heat blocks), it is required to perform quite a lot of arithmetic operations to calculate the coefficients of equations (11) for each dependent variable. Note also that the WF&Exchange block does not require data exchange and runs with perfect parallelism. Its share in the total cost of the program is more than 10%. The costs of implementing the Pressure block with a relatively small number of processes are commensurate with the costs of numerically solving one transport equation (9). However, with an increase in the degree of decomposition of the grid domain, the rate of convergence of the iterative Gauss-Seidel method slows down and more iterations are required to achieve a given accuracy. This entails an increase in the parallel execution time of the Pressure block (Figs. 2, 3).

After calculating the grid functions of the desired dependent variables for each equation of the form (11), in accordance with the selected grid pattern, the boundary values of the grid subdomain in the xz and yz planes are exchanged. Naturally, with an increase in the number of processes used, the ratio of the computation time for each subdomain to the time of exchange of boundary grid values to ensure the uniformity of parallel computations decreases (Fig. 2).

Figure 3 shows that, due to a fairly large number of auxiliary calculations in calculating the coefficients of finite-difference equations (11), due to its nonlinearity and time-consuming source terms, the speedup of all blocks of the parallel code, except for the pressure calculation block, is close to ideal even with an increase in the fraction time spent on exchange operations at each time step. This character of the speedup of calculations is preserved even with a relatively small amount of grid nodes in the 2D decomposition region. This is confirmed by the above results of speedup in the numerical solution of one convective-diffusion equation (Fig. 1).



(a)



(b)

Figure 3. The speedup of the main blocks of the TSUNM3 program depending on the number of processes used. The calculations use Grid1 (a) and Grid2 (b)

When solving the finite-difference equation (13) by the iterative Gauss-Seidel method at each time step, exchange operations must be performed at each iteration. Therefore, for the considered grids Grid1 and Grid2 and the parallel programming technology, already at 144 processes, speedup saturation for the pressure calculation block is observed and the time spent on its numerical implementation begins to occupy an increasing share in the total amount of calculations. Nevertheless, since the rest of the code blocks demonstrate a high level of speedup, in general, the high efficiency of using a multiprocessor computing system remains for the entire parallel program (Fig. 3, Tab. 1).

To prove the good strong, relative and weak scalability [31] of the developed parallel program, we also present Fig. 4, which shows the graphs of absolute efficiency E, relative efficiency rE and "weak" efficiency wE from the number of processes used, which are calculated by the following formulas:

$$E(p, Grid) = \frac{T(1, Grid)}{pT(p, Grid)}; rE(p, Grid) = \frac{E(p, Grid)}{E(p_{prev}, Grid)}; wE(p) = \frac{T(p, Grid1)}{T(4p, Grid2)}. \tag{14}$$

Here p is the number of processes used, Grid is the size of the grid area, $T(p, Grid)$ is the program computation time on the Grid on p processes, p_{prev} is the number of processes in the sequence of test runs preceding p (see Tab. 1, in accordance with which, for example, at $p = 36$, $p_{prev} = 16$), Grid2 is a grid with 4 times more nodes than Grid1.

Figure 4 shows that the developed computational code has good scalability, since its absolute efficiency is not less than 70% when using up to 256 processes. In addition, the ratio of neighboring values of absolute efficiency starts to decrease below 0.9 only when more than 100 processes are used. The values of weak scalability wE also do not fall below 0.9, which shows a small change in the computation time with a synchronous increase in the grid size and the number of processes used, while maintaining the number of processed grid values in each subdomain of the 2D decomposition.

Thus, the reasons for the high scalability of the developed TSUNM3 program compared to parallel programs for the numerical solution of traditional Navier-Stokes equations with constant coefficients, which consider three equations for the velocity components and an equation for finding pressure (see, for example, [32]), in our opinion, are the following:

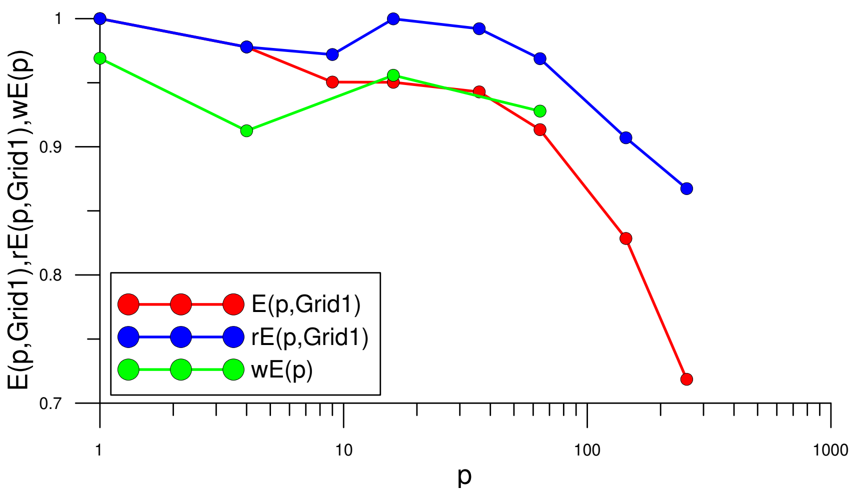


Figure 4. The scalability of the developed program

- in this program, six more three-dimensional non-stationary convective-diffusion equations of the form (9) are additionally solved numerically;
- the developed parallel numerical method for solving the transport equation (9), as shown by the above results, has good scalability up to the use of a relatively small number of nodes in the grid subdomains obtained after applying the two-dimensional decomposition; the latter is associated with a large volume of calculations of the coefficients of difference equations (11) and source terms of equations (9), especially for blocks of microphysics of moisture, turbulence and heat transfer;
- apparently, with more intensive use of fast cache memory with an increase in the degree of decomposition of the finite difference problem with an increase in the number of processes used.

The numerical method (11) and MPI parallel programming technology with two-dimensional decomposition of the grid area along horizontal directions were also used for the pollutant transport model (1), which is applied in off-line mode after calculations using the TSUNM3 numerical weather prediction model. The computations in question include 12 equations of the form (1), which are solved on a $100 \times 100 \times 50$ grid with a horizontal step of 500 m and a time step of ~ 1 sec for a 50×50 km area. Table 2 shows the speedup values for the pollutant transport model (1) parallel program.

Table 2. Pollutant transport parallel program speedup values at different process quantities

Process	4	16	25	100
SpeedUp	3.9	13.6	19.4	70.8

Thus, the table shows that the efficiency of parallel computing for the model of air quality (1) in question does not fall below 70%.

Conclusion

A mathematical model and an effective numerical method for the short-term prediction of dangerous meteorological conditions and atmospheric air quality over limited stretches of land encompassing large population centers designed to be used on multiprocessor distributed-memory systems were developed in the course of this study. The mathematical model includes a pollutant transport model with a reduced chemical mechanism and a non-hydrostatic mesoscale meteorological model with a modern atmospheric moisture microphysics parametrization scheme. The numerical solution of the non-stationary inhomogeneous convection diffusion equations of the modeling system is carried out by means of the transformation of coordinates, the finite volume method and semi-implicit difference schemes of the second order of approximation, which are solved using the TDMA method with a linear dependence of the number of arithmetic operations on the size of the grid. Difference equations used for pressure field computations that ensure correct satisfaction of continuity equations in the difference form are solved using the Gauss-Seidel iteration method and asynchronous relaxation.

Parallel implementation of the chosen numerical scheme for solving the equations of the mathematical model was based on the use of MPI parallel programming technology, two-dimensional decomposition of the grid area along horizontal directions (x and y axes), and introduction of additional fictitious grid nodes along the perimeter of the decomposition subdomains. The results of computational experiments carried out using the TSU Cyberia cluster showed good software scalability and computer resource utilization efficiency were not less than 70% when using up to 256 of the cluster's computing cores. In this case, numerical prediction of local weather conditions for the next 24 hours was carried out in 12 minutes on a $96 \times 96 \times 50$ grid and in 39 minutes on a $192 \times 192 \times 50$ grid with a horizontal resolution of 1 km. Numerical prediction of surface air quality for the next 24 hours was carried out in 23 minutes on a $100 \times 100 \times 50$ grid with a horizontal resolution of 500 m.

Acknowledgements

The work was supported by the Russian Science Foundation (project no. 19-71-20042).

References

1. Sokhi, R., Baklanov, A., Schlünzen, H.: Mesoscale Modelling for Meteorological and Air Pollution Applications. Anthem Press, New York (2018)

2. Dabdub, D., Seinfeld, J.H.: Parallel Computation in Atmospheric Chemical Modeling. Parallel Computing 22, 111–130 (1996). `https://doi.org/10.1016/0167-8191(95)00063-1`

3. Baklanov, A.A., Korsholm, U.S., Mahura, A.G., et al.: Physic a land chemical weather forecasting as a joint problem: two-way interacting integrated modelling. American Meteorological Society, Seattle (2011)

4. Nuterman, R., Korsholm, U., Zakey, A., et al.: New developments in Enviro-HIRLAM online integrated modelling system. Geophysical Research Abstracts 15 (2013)

5. Hood, C., MacKenzie, I., Stocker, J., et al.: Air quality simulations for London using a coupled regional-to-local modelling system. Atmospheric Chemistry and Physics 18, 11221–

11245 (2018). https://doi.org/10.5194/acp-18-11221-2018

6. Strunk, A., Ebel, A., Elbern, H.: A nested application of four-dimensional variational assimilation of tropospheric chemical data. International Journal of Environment and Pollution 46(1–2), 43–60 (2011). https://doi.org/10.1504/IJEP.2011.042607

7. Skamarock, W.C., Klemp, J.B., Dudhia, J., et al.: A Description of the Advanced Research WRF Model Version 4. National Center for Atmospheric Research, Colorado (2021). https://doi.org/10.5065/1dfh-6p97

8. Stockwell, W.R., Kirchner, F., Kuhn, M., Seefeld, S.: A new mechanism for regional atmospheric chemistry modeling. Journal of Geophysical Research Atmospheres 102(22), 25847–25879 (1997). https://doi.org/10.1029/97jd00849

9. Carpenter, K.: Note on the paper 'Radiation conditions for the lateral boundaries of limited-area numerical models'. Quarterly Journal of the Royal Meteorological Society 108(457), 717–719 (1982). https://doi.org/10.1002/qj.49710845714

10. Wesley, M.L.: Parameterisation of surface resistances to gaseous dry deposition in regional-scale numerical models. Atmospheric Environment 23(6), 1293–1304 (1989). https://doi.org/10.1016/0004-6981(89)90153-4

11. Department of Natural Resources and Environmental Protection of the Tomsk Region. https://depnature.tomsk.gov.ru/2019-god, accessed: 2021-10-29

12. Hurley, P.J.: TAPM V4. Part 1: Technical Description. CSIRO Marine and Atmospheric Research, Aspendale (2008). https://doi.org/10.4225/08/585c175bc5884

13. Stockwell, W.R., Goliff, W.S.: Comment on "Simulation of a reacting pollutant puff, using an adaptive grid algorithm" by R.K. Srivastava et al. Journal of Geophysical Research Atmospheres 107(22), 4643–4650 (2002). https://doi.org/10.1029/2002JD002164

14. Starchenko, A.V., Bart, A.A., Kizhner, L.I., Danilkin, E.A.: Mesoscale meteorological model TSUNM3 for the study and forecast of meteorological parameters of the atmospheric surface layer over a major population center. Tomsk State University Journal of Mathematics and Mechanics 66, 35–55 (2020). https://doi.org/10.17223/19988621/66/2

15. Starchenko, A., Prokhanov, S., Danilkin, E., Lechinsky, D.: Numerical Forecast of Local Meteorological Conditions on a Supercomputer. In: Voevodin, V., Sobolev, S. (eds.) Supercomputing. RuSCDays 2020. Communications in Computer and Information Science, vol. 1331, pp. 273–284. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64616-5_24

16. Pielke, R.A.: Mesoscale Meteorological modelling. Academic Press, Orlando (1984)

17. Hong, S., Lim, J.: The WRF single-moment 6-class microphysics scheme (WSM6). Journal of the Korean Meteorological Society 42(2), 129–151 (2006)

18. Andrén, A.: Evaluation of a Turbulence Closure Scheme Suitable for Air-Pollution Applications. Journal of Applied Meteorology and Climatology 29(3), 224–239(1990). https://doi.org/10.1175/1520-0450(1990)029<0224:EOATCS>2.0.CO;2

19. Yamada, T.: Simulations of Nocturnal Drainage Flows by a q^2l Turbulence Closure Model. Journal of Atmospheric Sciences 40(1), 91–106 (1983). `https://doi.org/10.1175/1520-0469(1983)040<0091:SONDFB>2.0.CO;2`

20. Smagorinsky, J.: General Circulation Experiments With the Primitive Equations: Part I. The Basic Experiment. Monthly Weather Review 91(2), 99–164 (1963). `https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2`

21. Mahrer, Y., Pielke, R.A.: A numerical study of the airflow over irregular terrain. Beitr. Phys. Atmosph. 50, 98–113 (1977)

22. Stephens, G.: Radiation Profiles in Extended Water Clouds. Part II: Parameterization Schemes. Journal of the Atmospheric Sciences 35(11), 2123–2132 (1978). `https://doi.org/10.1175/1520-0469(1978)035<2123:RPIEWC>2.0.CO;2`

23. Monin, A.S., Obukhov, A.M.: Basic laws of turbulent mixing in the surface layer of the atmosphere. Tr. Akad. Nauk SSSR Geofiz. 24, 163–187 (1954)

24. Dyer, A.J., Hicks, B.B.: Flux-gradient relationships in the constant flux layer. Quarterly Journal of the Royal Meteorological Society 96(410), 715–721 (1970). `https://doi.org/10.1002/qj.49709641012`

25. Tolstykh, M., Goyman, G., Fadeev, R., Shashkin, V.: Structure and Algorithms of SL-AV Atmosphere Model Parallel Program Complex. Lobachevskii Journal of Mathematics 39(4), 587–595 (2018). `https://doi.org/10.1134/S1995080218040145`

26. Mazumder, S.: Numerical Methods for Partial Differential Equations. Finite Difference and Finite Volume Methods. Academic Press (2016)

27. Van Leer, B.: Towards the ultimate conervative difference scheme. II. Monotonicity and conservation combined in a second order scheme. Journal of Computational Physics 14(4), 361–370 (1974). `https://doi.org/10.1016/0021-9991(74)90019-9`

28. Patankar, S.: Numerical Heat Transfer and Flow. CRC Press (2009). `https://doi.org/10.1201/9781482234213`

29. Ortega, J.O.: Introduction to Parallel and Vector Solution of Linear System. Plenum Press New York, Charlottesville (1988)

30. Starchenko, A.V., Danilkin, E.A., Prohanov, S.A., Leshchinskiy, D.V.: Parallel implementation of a numerical method for solving transport equations for the mesoscale meteorological model TSUNM3. Journal of Physics: Conference Series 1715(1), 012073 (2020). `https://doi.org/10.1088/1742-6596/1715/1/012073`

31. Bruneau, C.-H., Khadra, K.: Highly parallel computing of a multigrid solver for 3D Navier-Stokes equations. Journal of Computational Science 17(1), 35–46 (2016). `https://doi.org/10.1016/j.jocs.2016.09.005`

32. Wang, Y., Baboulin, M., Dongarra, J., et al.: A Parallel Solver for Incompressible Fluid Flows. Procedia Computer Science 18, 439–448 (2013). `https://doi.org/10.1016/j.procs.2013.05.207`