INFORMATICA, 2022, Vol. 33, No. 3, 477–497
© 2022 *Vilnius University*
DOI: https://doi.org/10.15388/22-INFOR481

Parallel 3D ADI Scheme for Partially Dimension Reduced Heat Conduction Problem

Raimondas ČIEGIS[1], Remigijus ČIEGIS[2], Olga SUBOČ[1,*]

[1] *Vilnius Gediminas Technical University, Saulėtekio av. 11, LT-10223 Vilnius, Lithuania*
[2] *Vilnius University Kaunas Faculty, Muitinės st. 8, LT-44280 Kaunas, Lithuania*
e-mail: rc@vgtu.lt, remigijus.ciegis@knf.vu.lt, os@vgtu.lt

Received: November 2021; accepted: April 2022

Abstract. This model describes the heat equation in 3D domains, and this problem is reduced to a hybrid dimension problem, keeping the initial dimension only in some parts and reducing it to one-dimensional equation within the domains in some distance from the base regions. Such mathematical models are typical in industrial installations such as pipelines. Our aim is to add two additional improvements into this methodology. First, the economical ADI type finite volume scheme is constructed to solve the non-classical heat conduction problem. Special interface conditions are defined between 3D and 1D parts. It is proved that the ADI scheme is unconditionally stable. Second, the parallel factorization algorithm is proposed to solve the obtained systems of discrete equations. Due to both modifications the run-time of computations is reduced essentially. Results of computational experiments confirm the theoretical error analysis and scalability estimates of the parallel algorithm.

Key words: hybrid dimension model, ADI scheme, parallel factorization algorithm, heat conduction, piplines.

1. Introduction

An important trend in development of mathematical models, simulating various real world applications, deals with new type nonlocal boundary and conjugation conditions. The nonlocality mechanism helps to simulate important physical processes more accurately. Still the existence and uniqueness of the solution, its stability is very sensitive to approximation of such nonlocal boundary conditions and require development and analysis of special discrete approximation techniques and modifications of implementation algorithms. As one important example we mention the simulation of enzyme-catalysed glucose oxidation and redox reactions over inhomogeneous surfaces (Skakauskas *et al.*, 2018; Čiegis *et al.*, 2018).

It is a standard situation in applications of the mathematical modelling technique that systems of nonstationary 3D differential equations should be solved. Even if state of the art high-order discrete approximations are constructed on adaptive meshes, the total complexity of the obtained computational algorithms is very large. It restricts our possibility to

[*] Corresponding author.

include such models into general optimization routines or to use them in real time decision making applications.

Recently, a new important technique has been proposed to solve such applied problems more efficiently. A model reduction technique is applied to avoid the effects of curse of dimensionality. The given 3D heat conduction problem is reduced to a hybrid dimension problem, keeping the initial dimension only in some parts of the domain and reducing it to one-dimensional equation within the domains in some distance from the base regions. Such mathematical models are typical in industrial installations such as pipelines.

Our aim is to add two additional improvements into this methodology. First, the economical ADI type finite volume scheme is constructed to solve the non-classical heat conduction problem. It is proved that the ADI scheme is unconditionally stable. Second, the parallel factorization algorithm is proposed to solve the obtained systems of discrete equations. Due to both modifications the run-time of computations is reduced essentially.

The method of partial dimension reduction was first introduced in Panasenko (1998) and then developed in Panasenko (2005). It is based on the asymptotic analysis of a solution of a partial differential equation set in a thin domain and on a projection of the variational model formulation on a subspace of functions having a form of the asymptotic expansion in all zones of regular behaviour of the solution. Then the dimension of the problem is reduced within the big part of the domain keeping the full dimension description only in small zones of a singular behaviour of the solution.

Applying this approach a model with nonlocal junctions of one dimensional and full dimensional equations is obtained. Note that nonlocal conjugation conditions make the main challenges in development of effective parallel numerical methods for this class of problems of hybrid dimension. Let us mention some papers where different techniques were used to discretize the obtained hybrid mathematical model. A finite volume setting was studied in Panasenko and Viallon (2015). The method of asymptotic partial decomposition of the domain was compared with several existing methods of dimension reduction in Amar and Givoli (2018) and its effectiveness with respect to the precision and time of the execution is shown. In a recent paper (Čiegis *et al.*, 2020) ADI type finite volume scheme was considered for a 2D heat conduction problem.

However, the question about the most effective parallel numerical solvers for such problems of hybrid dimension is still an important topic. Non-classical conjugation conditions require to develop special solvers to get the discrete solutions. A straightforward way is to solve the obtained systems of linear equations by using general iterative solvers, e.g. well-known multigrid solvers. For multidimensional parabolic problems an alternative and most popular way is to use splitting methods (Hundsdorfer and Verwer, 2003; Samarskii, 2001). It is well-known that in many cases ADI type methods are the best selection.

Thus in this paper we have developed and analysed special efficient ADI schemes to solve the hybrid 3D partial dimension reduction models for a nonstationary heat conduction problem. Our second goal is to propose efficient parallel implementations of the constructed ADI schemes. All main topics required in the analysis of discrete schemes, i.e. the approximation, positivity and stability of the solution are investigated. The accuracy of the reduced dimension model and the scalability of the parallel algorithms are tested in a series of computational experiments.

The approximation error is investigated quite straightforwardly, since the regularity of the solution and dependence of it on the reduction parameter δ is known from results given in Amosov and Panasenko (2018), Panasenko (2005). A more complicated step is to investigate the stability of proposed parallel ADI schemes in the case of new non-local conjugation conditions.

The rest of the paper is organized in the following way. In Section 2 the problem is formulated. The nonstationary heat conduction equation is formulated in the 3D parallelepiped and the initial and boundary conditions are specified. In this section also, the partially reduced dimension model is constructed and an approximate solution is defined by considering a simplified domain when 3D small size subregions are coupled with a 1D line. The non-local conjugation conditions are defined at the truncations of the tube. In Section 3 the full problem in the 3D domain is solved by using the ADI stability correction scheme. It is proved that this scheme is unconditionally stable. Its implementation requires solving many linear systems with tridiagonal matrix and the classical factorization algorithm is used. The Wang parallel factorization algorithm also can be used for parallel computers. In Section 4 the modified ADI stability correction scheme is constructed for the reduced dimension heat conduction problem. It is proved that the unique solution of the obtained system of linear equations exists for each time level. The efficient factorization algorithm is defined, it is based on algorithms presented in Čiegis *et al.* (2020). A parallel version of the modified factorization algorithm for implementation of the ADI scheme approximating the reduced dimension heat conduction problem is constructed in Section 5. A theoretical scalability analysis of this parallel algorithm is done. In Section 6 results of computational experiments are presented and analysed. Some final conclusions are done in Section 7.

2. Problem Formulation

Let us consider a parallelepiped $\Omega \subset \mathbb{R}^3$, which has the following form $\Omega = (0, X_1) \times D$, where D is a rectangle $D = \{(x_2, x_3) : 0 < x_j < X_j,\ \ j = 2, 3\}$.

We are interested to solve a linear heat equation in $\Omega \times (0, T]$:

$$\frac{\partial u}{\partial t} = \sum_{j=1}^{3} \frac{\partial^2 u}{\partial x_j^2} + f(x_1, x_2, x_3, t), \quad (x_1, x_2, x_3, t) \in Q_T = \Omega \times (0, T], \tag{1}$$

$$\begin{aligned} &u(0, x_2, x_3, t) = g_1(x_2, x_3, t), \quad u(X_1, x_2, x_3, t) = g_2(x_2, x_3, t),\\ &\quad (x_2, x_3, t) \in \bar{D} \times (0, T], \end{aligned} \tag{2}$$

$$\frac{\partial u}{\partial \eta} = 0, \quad (x_1, x_2, x_3, t) \in \partial\Omega_1 \times (0, T], \tag{3}$$

$$u(x_1, x_2, x_3, 0) = u^0(x_1, x_2, x_3), \quad (x_1, x_2, x_3) \in \Omega, \tag{4}$$

where $\partial\Omega_1 = \partial\Omega \setminus (D \times \{x_1 = 0\} \cup \{x_1 = X_1\})$ denotes a part of the boundary, where the Neumann boundary condition is specified.

Next, following Amosov and Panasenko (2018), we consider a modified approximate problem. First, let $S(u)$

$$S(u) = \frac{1}{|D|} \int_0^{X_2} \int_0^{X_3} u(x_1, x_2, x_3, t)\, dx_2\, dx_3$$

denote the averaging operator. We assume that the initial condition u^0 and source function f satisfy the relations

$$u^0(x_1, x_2, x_3) = S\big(u^0\big), \qquad f(x_1, x_2, x_3, t) = S(f), \quad (x_1, t) \in (0, X_1) \times (0, T].$$

It means that u^0 and f don't depend on x_2, x_3 within Ω.

Denote a reduced dimension domain $\Omega_\delta = \{(x_1, x_2, x_3) \in (\delta, X_1 - \delta) \times D\}$. Function U is called an approximate solution to problem (1)–(4) if it satisfies the following problem (Amosov and Panasenko, 2018)

$$\frac{\partial U}{\partial t} = \sum_{j=1}^{3} \frac{\partial^2 U}{\partial x_j^2} + f(x_1, t), \quad (x_1, x_2, x_3, t) \in (\Omega \setminus \Omega_\delta) \times (0, T], \tag{5}$$

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x_1^2} + f(x_1, t), \quad (x_1, x_2, x_3, t) \in \Omega_\delta \times (0, T], \tag{6}$$

$$\begin{aligned} &U(0, x_2, x_3, t) = g_1(x_2, x_3, t), \quad U(X_1, x_2, x_3, t) = g_2(x_2, x_3, t), \\ &\quad (x_2, x_3, t) \in \bar{D} \times (0, T], \end{aligned} \tag{7}$$

$$\frac{\partial U}{\partial \eta} = 0, \quad (x_1, x_2, x_3, t) \in \partial\Omega_{\delta,1} \times (0, T], \tag{8}$$

$$U(x_1, x_2, x_3, 0) = u^0(x_1, x_2, x_3), \quad (x_1, x_2, x_3) \in \Omega, \tag{9}$$

where $\partial\Omega_{\delta,1} = \partial\Omega_\delta \setminus (D \times \{x_1 = \delta\} \cup \{x_1 = X_1 - \delta\})$.

In $\Omega_\delta \times (0, T]$ the solution U doesn't depend on x_2, x_3, i.e.

$$U(x_1, x_2, x_3, t) = S(U), \quad (x_1, x_2, x_3, t) \in \Omega_\delta \times (0, T].$$

Thus in the reduced domain it is sufficient to find 1D space function $U(x_1, t)$.

By analysing the weak form of the heat equation, it is shown in Amosov and Panasenko (2018) that the following conjugation conditions are valid at the truncation points of the space domain

$$U|_{x_1=\delta-0} = U|_{x_1=\delta+0}, \qquad U|_{x_1=X_1-\delta-0} = U|_{x_1=X_1-\delta+0}, \tag{10}$$

$$\left.\frac{\partial S(U)}{\partial x_1}\right|_{x_1=\delta-0} = \left.\frac{\partial U}{\partial x_1}\right|_{x_1=\delta+0}, \qquad \left.\frac{\partial U}{\partial x_1}\right|_{x_1=X_1-\delta-0} = \left.\frac{\partial S(U)}{\partial x_1}\right|_{x_1=X_1-\delta+0}. \tag{11}$$

The conditions (10) are classical and mean that U is continuous at the truncation points. While the remaining two conditions (11) are nonlocal and they define the conservation of full fluxes along the separation planes.

3. ADI Scheme for the 3D Differential Problem

The uniform spatial mesh $\bar{\Omega}_h = \bar{\omega}_1 \times \bar{\omega}_2 \times \bar{\omega}_3$ is defined as

$$\bar{\omega}_k = \{x_{kj} : x_{kj} = jh_k,\ j = 0, \dots, J_k\}, \qquad x_{kJ_k} = X_k, \quad k = 1, 2, 3.$$

For simplicity of notations we also consider a uniform time mesh:

$$\bar{\omega}_t = \left\{t^n : t^n = n\tau,\ n = 0, \dots, N\right\}, \quad t^N = T.$$

Let U^n_{ijk} be a numerical approximation to the exact solution $u(x_{1i}, x_{2j}, x_{3k}, t^n)$ of problem (1)–(4) at the grid point $(x_{1i}, x_{2j}, x_{3k}, t^n)$.

The following standard operators are defined for discrete functions:

$$\partial_{x_1} U^n_{ijk} := \frac{U^n_{ijk} - U^n_{i-1,j,k}}{h_1},$$

$$\partial_{x_2} U^n_{ijk} := \frac{U^n_{ijk} - U^n_{i,j-1,k}}{h_2}, \qquad \partial_{x_3} U^n_{ijk} := \frac{U^n_{ijk} - U^n_{i,j,k-1}}{h_3},$$

$$A^h_1 U^n_{ijk} := -\frac{1}{h_1}\big(\partial_{x_1} U^n_{i+1,j,k} - \partial_{x_1} U^n_{ijk}\big), \quad 0 < i < J_1$$

$$A^h_2 U^n_{ijk} := \begin{cases} -\frac{2}{h_2}\partial_{x_2} U^n_{i1k}, & j = 0, \\ -\frac{1}{h_2}\big(\partial_{x_2} U^n_{i,j+1,k} - \partial_{x_2} U^n_{ijk}\big), & 0 < j < J_2, \\ \frac{2}{h_2}\partial_{x_2} U^n_{iJ_2k}, & j = J_2, \end{cases}$$

$$A^h_3 U^n_{ijk} := \begin{cases} -\frac{2}{h_3}\partial_{x_3} U^n_{ij1}, & k = 0, \\ -\frac{1}{h_3}\big(\partial_{x_3} U^n_{i,j,k+1} - \partial_{x_3} U^n_{ijk}\big), & 0 < k < J_3, \\ \frac{2}{h_3}\partial_{x_3} U^n_{ijJ_3}, & k = J_3. \end{cases}$$

The discrete approximation of diffusion operators is obtained by using the finite volume method.

Then the heat conduction problem (1)–(4) is approximated by the following alternating direction implicit (ADI) stability correction scheme (Hundsdorfer and Verwer, 2003)

$$\frac{U^{n+\frac{1}{4}}_{ijk} - U^n_{ijk}}{\tau} + \sum_{l=1}^{3} A^h_l U^n_{ijk} = f^{n+\frac{1}{2}}_{ijk}, \quad (x_{1i}, x_{2j}, x_{3k}) \in \omega_1 \times \bar{\omega}_2 \times \bar{\omega}_3,$$

$$\frac{U^{n+\frac{l}{4}}_{ijk} - U^{n+\frac{l-1}{4}}_{ijk}}{\tau} + \frac{1}{2} A^h_{5-l}\big(U^{n+\frac{l}{4}}_{ijk} - U^n_{ijk}\big) = 0, \quad l = 2, 3, 4. \tag{12}$$

At the first step the heat conduction problem (1)–(4) is approximated by the explicit Euler scheme. It is well known that this scheme is only conditionally stable. The next three steps of scheme (12) guarantee the unconditional stability of the discrete scheme and they increase the accuracy of approximation in time until the second order.

At all three steps 3D systems are split into a set of 1D problems along x_3, x_2 and x_1 coordinates respectively, and each subproblem is solved by using the classical fast factorization algorithm.

Lemma 1. *If a solution of the problem* (1)–(4) *is sufficiently smooth, then the approximation error of ADI scheme* (12) *is* $O(\tau^2 + h_1^2 + h_2^2 + h_3^2)$.

Proof. For a self-completeness of the text and in order to explain the approximation property of the ADI scheme (12) we present a short proof of this result (for a detailed analysis see Hundsdorfer and Verwer, 2003). By adding all equations (12) the following discrete equation is obtained

$$\frac{U_{ijk}^{n+1} - U_{ijk}^n}{\tau} + \sum_{l=2}^{4} \frac{1}{2} A_{5-l}^h \big(U_{ijk}^{n+\frac{l}{4}} + U_{ijk}^n\big) = f_{ijk}^{n+\frac{1}{2}}. \tag{13}$$

In order to compute the approximation error of (12), we use the relations

$$U_{ijk}^n \sim u\big(x_{1i}, x_{2j}, x_{3k}, t^n\big), \qquad U_{ijk}^{n+\frac{l}{4}} \sim u\big(x_{1i}, x_{2j}, x_{3k}, t^{n+1}\big), \quad l = 2, 3, 4.$$

Thus the approximation error of ADI scheme (12) with respect to time coordinate is the same as of the classical Crank-Nicolson method. The second order accuracy of the constructed approximations of space derivatives is shown, e.g. in Hundsdorfer and Verwer (2003), Samarskii (2001). □

It is a straightforward task to show the result presented in the following lemma.

Lemma 2. *The discrete operators* A_1^h, A_2^h *and* A_3^h *are symmetric,* A_1^h *is positive definite and* A_2^h, A_3^h *are non-negative operators.*

All operators A_1^h, A_2^h and A_3^h commute, thus there exists a complete system of functions

$$\big\{\phi_{1i}(x_1)\phi_{2j}(x_2)\phi_{3k}(x_3), \ 1 \leqslant i \leqslant J_1 - 1, \ 0 \leqslant j \leqslant J_2, \ 0 \leqslant k \leqslant J_3\big\},$$

where $\phi_{lm}(x_l)$ are eigenvectors of the operator A_l^h, $l = 1, 2, 3$:

$$A_l^h \phi_{lm} = \lambda_{lm} \phi_{lm}, \quad l = 1, 2, 3$$

and λ_{lm} are eigenvalues

$$\lambda_{1i} \geqslant \lambda_{11} > 0, \qquad \lambda_{lj} \geqslant 0, \quad l = 2, 3. \tag{14}$$

Lemma 3. *ADI scheme* (12) *is unconditionally stable.*

Proof. For the given normal operators A_l^h, $l = 1, 2, 3$ the Fourier stability analysis can be used. Let us consider the solution of ADI scheme (12) in the case when boundary conditions $g_j = 0$, $j = 1, 2$ and $f(x, t) = 0$. The solution of (12) can be written as

$$U_{ijk}^n = \sum_{l=1}^{J_1-1} \sum_{m=0}^{J_2} \sum_{r=0}^{J_3} c_{lmr}^n \phi_{1l}(x_{1i}) \phi_{2m}(x_{2j}) \phi_{3r}(x_{3k}).$$

Substituting this formula into equations (12), we obtain for each mode the stability equations

$$c_{lmr}^{n+1} = q_{lmr} c_{lmr}^n,$$

where the discrete stability function q_{lmr} is defined as

$$q_{lmr} = \frac{(1 - \frac{\tau}{2}(\lambda_{1l} + \lambda_{2m} + \lambda_{3r}) + \frac{\tau^2}{4}(\lambda_{1l}\lambda_{2m} + \lambda_{1l}\lambda_{3r} + \lambda_{2m}\lambda_{3r}) + \frac{\tau^3}{8}\lambda_{1l}\lambda_{2m}\lambda_{3r})}{(1 + 0.5\tau\lambda_{1l})(1 + 0.5\tau\lambda_{2m})(1 + 0.5\tau\lambda_{3r})}.$$

Since eigenvalues satisfy estimates (14), then $|q_{lmr}| < 1$ and the ADI scheme (12) is unconditionally stable in the L_2 norm. □

It is easy to see that the discrete stability function approximates the continuous stability function of the differential equation (1) with the second order accuracy.

4. The ADI Scheme for the Partially Dimension Reduced Problem

Let K_1 and K_2 define the truncation points of the domain $x_{1K_1} = \delta$, $x_{1K_2} = X_1 - \delta$. Then the spatial mesh ω_1 is splitted into three parts:

$$\begin{aligned}
\omega_{11} &= \{x_{1i} : x_{1i} = ih_1, \ i = 1, \dots, K_1 - 1\}, \\
\bar{\omega}_{12} &= \{x_{1i} : x_{1i} = ih_1, \ i = K_1, \dots, K_2\}, \\
\omega_{13} &= \{x_{1i} : x_{1i} = ih_1, \ i = K_2 + 1, \dots, J_1 - 1\}.
\end{aligned}$$

We note that more general meshes are used in Viallon (2013); Panasenko and Viallon (2015), where some atypical cells along the interface are constructed, and that lead to non-admissible meshes. In this paper we use the finite volume approach. For more general non-uniform meshes it can be recommended to use a general framework of finite element and finite difference schemes for construction of discrete approximations on non-matching grids (Eymard *et al.*, 2000; Faille, 1992). A numerical comparison of these methods for 1D-2D discrete schemes is done in Viallon (2013).

The heat conduction problem (5)–(11) is approximated by the modified version of the ADI scheme

$$
\begin{aligned}
&\frac{U^{n+\frac{1}{4}}_{ijk} - U^n_{ijk}}{\tau} + \sum_{l=1}^{3} A^h_l U^n_{ijk} = f^{n+\frac{1}{2}}_{ijk}, \quad (x_{1i}, x_{2j}, x_{3k}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3,\\
&\frac{U^{n+\frac{1}{4}}_{i00} - U^n_{i00}}{\tau} + A^h_1 U^n_{i00} = f^{n+\frac{1}{2}}_{i00}, \quad x_{1i} \in \bar{\omega}_{12} \setminus \{x_{1K_1}, x_{1K_2}\},\\
&\frac{U^{n+\frac{1}{4}}_{K_100} - U^n_{K_100}}{\tau} + \frac{1}{h_1^2}\big(-S_h\big(U^n_{K_1-1}\big) + 2U^n_{K_100} - U^n_{K_1+1,0,0}\big) = f^{n+\frac{1}{2}}_{K_100},\\
&\frac{U^{n+\frac{1}{4}}_{K_200} - U^n_{K_200}}{\tau} + \frac{1}{h_1^2}\big(-S_h\big(U^n_{K_2+1}\big) + 2U^n_{K_200} - U^n_{K_2-1,0,0}\big) = f^{n+\frac{1}{2}}_{K_200},
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
&\frac{U^{n+\frac{l}{4}}_{ijk} - U^{n+\frac{l-1}{4}}_{ijk}}{\tau} + \frac{1}{2} A^h_{5-l}\big(U^{n+\frac{l}{4}}_{ijk} - U^n_{ijk}\big) = 0, \quad l = 2, 3,\\
&\quad (x_{1i}, x_{2j}, x_{3k}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3,
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
&\frac{U^{n+1}_{ijk} - U^{n+\frac{3}{4}}_{ijk}}{\tau} + \frac{1}{2} A^h_1\big(U^{n+1}_{ijk} - U^n_{ijk}\big) = 0,\\
&\quad (x_{1i}, x_{2j}, x_{3k}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3,
\end{aligned}
\tag{17}
$$

$$
\frac{U^{n+1}_{i00} - U^{n+\frac{3}{4}}_{i00}}{\tau} + \frac{1}{2} A^h_1\big(U^{n+1}_{i00} - U^n_{i00}\big) = 0, \quad x_{1i} \in \bar{\omega}_{12} \setminus \{x_{1K_1}, x_{1K_2}\},
\tag{18}
$$

$$
\begin{aligned}
&\frac{U^{n+1}_{K_100} - U^{n+\frac{3}{4}}_{K_100}}{\tau} + \frac{1}{2h_1^2}\big(-S_h\big(U^{n+1}_{K_1-1}\big) + 2U^{n+1}_{K_100} - U^{n+1}_{K_1+1,0,0}\\
&\quad + S_h\big(U^n_{K_1-1}\big) - 2U^n_{K_100} + U^n_{K_1+1,0,0}\big) = 0,
\end{aligned}
\tag{19}
$$

$$
\begin{aligned}
&\frac{U^{n+1}_{K_200} - U^{n+\frac{3}{4}}_{K_200}}{\tau} + \frac{1}{2h_1^2}\big(-S_h\big(U^{n+1}_{K_2+1}\big) + 2U^{n+1}_{K_200} - U^{n+1}_{K_2-1,0,0}\\
&\quad + S_h\big(U^n_{K_2+1}\big) - 2U^n_{K_200} + U^n_{K_2-1,0,0}\big) = 0.
\end{aligned}
\tag{20}
$$

Here S_h denotes the discrete averaging operator:

$$
S_h\big(U^n_i\big) = \frac{1}{|D|} \sum_{j=0}^{J_2} \sum_{k=0}^{J_3} d_{2j} d_{3k} U^n_{ijk} h_2 h_3,
$$

where $d_{lm} = 1$, if $0 < m < J_l$ and $d_{lm} = 0.5$, if $m = 0, J_l$, $l = 2, 3$. Notation of the argument U^n_i indicates that the two dimensional average of discrete function U^n is computed at the grid point x_{1i}.

Note that equations (19), (20) approximate the nonlocal flux conjugation conditions (11).

We define two discrete meshes $\Omega_{h,RD} = (\bar{\omega}_2 \times \bar{\omega}_3 \times (\omega_{11} \cup \omega_{13})) \cup \bar{\omega}_{12}$ and $\bar{\Omega}_{h,RD} = \Omega_{h,RD} \cup (\bar{\omega}_2 \times \bar{\omega}_3 \times (x_{10} \cup x_{1J_1}))$. Let us consider discrete functions $U_{ijk} = U(x_{1i}, x_{2j}, x_{3k})$ defined on the spatial mesh $\bar{\Omega}_{h,RD}$. We denote the set of such vectors by D_h.

Let us define three operators for $U \in D_h$:

$$\mathcal{A}_2^h U = \begin{cases} A_2^h U, & (x_{1i}, x_{2j}, x_{3i}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3, \\ 0, & x_{1i} \in \bar{\omega}_{12}, \end{cases}$$

$$\mathcal{A}_3^h U = \begin{cases} A_3^h U, & (x_{1i}, x_{2j}, x_{3i}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3, \\ 0, & x_{1i} \in \bar{\omega}_{12}, \end{cases}$$

$$\mathcal{A}_1^h U = \begin{cases} A_1^h U, & (x_{1i}, x_{2j}, x_{3i}) \in (\omega_{11} \cup \omega_{13}) \times \bar{\omega}_2 \times \bar{\omega}_3, \\ A_1^h U_{i00}, & x_{1i} \in \bar{\omega}_{12} \setminus \{x_{1K_1}, x_{1K_2}\}, \\ \frac{1}{h_1^2}(-S_h(U_{K_1-1}) + 2U_{K_100} - U_{K_1+1,0,0}), & x_1 = K_1, \\ \frac{1}{h_1^2}(-S_h(U_{K_2+1}) + 2U_{K_200} - U_{K_2-1,0,0}), & x_1 = K_2. \end{cases}$$

Then we can write the ADI scheme as

$$\frac{U^{n+\frac{1}{4}} - U^n}{\tau} + \big(\mathcal{A}_1^h + \mathcal{A}_2^h + \mathcal{A}_3^h\big)U^n = f^{n+\frac{1}{2}}, \tag{21}$$

$$\frac{U^{n+\frac{l}{4}} - U^{n+\frac{l-1}{4}}}{\tau} + \frac{1}{2}\mathcal{A}_{5-l}^h\big(U^{n+\frac{l}{4}} - U^n\big) = 0, \quad l = 2, 3, 4. \tag{22}$$

Next, our goal is to prove that the discrete operators $\mathcal{A}_1^h$, $\mathcal{A}_2^h$ and $\mathcal{A}_3^h$ are symmetric.

For $U, V \in D_h$, such that $U_{0jk} = U_{J_1jk} = 0$, $V_{0jk} = V_{J_1jk} = 0$, $(x_2, x_3) \in \bar{\omega}_2 \times \bar{\omega}_3$ the formulas

$$\begin{aligned} (U, V) = {} & \sum_{j=0}^{J_2} \sum_{k=0}^{J_3} d_{2j} d_{3k} \Bigg(\sum_{i=1}^{K_1-1} U_{ijk} V_{jk} + \sum_{i=K_2+1}^{K-1} U_{ijk} V_{ijk} \Bigg) h_1 h_2 h_3 \\ & + |D| \sum_{i=K_1}^{K_2} U_{i00} V_{i00} h_1, \\ \|U\| = {} & (U, U)^{1/2} \end{aligned}$$

define a scalar product and a norm in this vector space. We remind that $d_{lm} = 1$, if $0 < m < J_l$ and $d_{lm} = 0.5$, if $m = 0, J_l, l = 2, 3$.

Lemma 4. *The discrete operators $\mathcal{A}_1^h$, $\mathcal{A}_2^h$ and $\mathcal{A}_3^h$ are symmetric, $\mathcal{A}_1^h$ is positive definite and $\mathcal{A}_2^h$, $\mathcal{A}_3^h$ are non-negative operators.*

Proof. First, we investigate the operator $\mathcal{A}_2^h$. Applying the summation by part formula, we get

$$\begin{aligned}
\big(\mathcal{A}_2^h U, V\big) &= \sum_{j=0}^{J_2}\sum_{k=0}^{J_3} d_{2j}d_{3k}\Bigg(\sum_{i=1}^{K_1-1} A_2^h U_{ijk}V_{ijk} + \sum_{i=K_2+1}^{K-1} A_2^h U_{ijk}V_{ijk}\Bigg)h_1h_2h_3 \\
&\quad + \sum_{j=1}^{J_2}\sum_{k=0}^{J_3} d_{3k}\Bigg(\sum_{k=1}^{K_1-1} \partial_{x_2}U_{ijk}\partial_{x_2}V_{ijk} + \sum_{k=K_2+1}^{K-1} \partial_{x_2}U_{ijk}\partial_{x_2}V_{ijk}\Bigg)h_1h_2h_3 \\
&= \big(U, \mathcal{A}_2^h V\big), \quad \big(\mathcal{A}_2^h U, U\big) \geqslant 0.
\end{aligned}$$

In a similar way we get the estimates

$$\begin{aligned}
\big(\mathcal{A}_3^h U, V\big) &= \sum_{j=0}^{J_2}\sum_{k=0}^{J_3} d_{2j}d_{3k}\Bigg(\sum_{i=1}^{K_1-1} A_3^h U_{ijk}V_{ijk} + \sum_{i=K_2+1}^{K-1} A_3^h U_{ijk}V_{ijk}\Bigg)h_1h_2h_3 \\
&\quad + \sum_{j=0}^{J_2}\sum_{k=1}^{J_3} d_{2k}\Bigg(\sum_{k=1}^{K_1-1} \partial_{x_3}U_{ijk}\partial_{x_3}V_{ijk} + \sum_{k=K_2+1}^{K-1} \partial_{x_3}U_{ijk}\partial_{x_3}V_{ijk}\Bigg)h_1h_2h_3 \\
&= \big(U, \mathcal{A}_3^h V\big), \quad \big(\mathcal{A}_3^h U, U\big) \geqslant 0.
\end{aligned}$$

It follows from the obtained estimates that $\mathcal{A}_2^h$ and $\mathcal{A}_3^h$ are symmetric and non-negative definite operators.

Next, we investigate the operator $\mathcal{A}_1^h$. Applying the summation by parts formula and taking into account the boundary conditions for vectors U, V, and the nonlocal conjugation conditions (19), (20) we get

$$\begin{aligned}
&\big(\mathcal{A}_1^h U, V\big) \\
&\quad = \sum_{j=0}^{J_2}\sum_{k=0}^{J_3} d_{2j}d_{3k}\Bigg(\sum_{i=1}^{K_1-1} A_1^h U_{ijk}V_{ijk} + \sum_{i=K_2+1}^{K-1} A_1^h U_{ijk}V_{ijk}\Bigg)h_1h_2h_3 \\
&\qquad + |D|\Bigg(\sum_{i=K_1+1}^{K_2-1} A_1^h U_{i00}V_{i00}h_1 + \frac{1}{h_1}\big(-S_h(U_{K_1-1}) + 2U_{K_100} - U_{K_1+1,0,0}\big)V_{K_100} \\
&\qquad + \frac{1}{h_1}\big(-S_h(U_{K_2+1}) + 2U_{K_200} - U_{K_2-1,0,0}\big)V_{K_200}\Bigg) \\
&\quad = \sum_{j=0}^{J_2}\sum_{k=0}^{J_3} d_{2j}d_{3k}\Bigg(\sum_{i=1}^{K_1} \partial_{x_1}U_{ijk}\partial_{x_1}V_{ijk} + \sum_{i=K_2+1}^{K} \partial_{x_1}U_{ijk}\partial_{x_1}V_{ijk}\Bigg)h_1h_2h_3 \\
&\qquad + |D|\sum_{i=K_1+1}^{K_2} \partial_{x_1}U_{i00}\,\partial_{x_1}V_{i00}\,h_1 = \big(U, \mathcal{A}_1^h V\big), \quad \big(U, \mathcal{A}_1^h U\big) > 0.
\end{aligned}$$

Here we use the notation $U_{K_1jk} = U_{K_100}$, $U_{K_2jk} = U_{K_200}$ for $(x_{2j}, x_{3k}) \in \bar{\omega}_2 \times \bar{\omega}_3$.

It follows from the obtained estimates that $\mathcal{A}_1^h$ is a symmetric and positive definite operator. □

Due to nonlocal conjugation conditions (19), (20) the classical factorization algorithm should be modified in order to solve 1D subproblems (17)–(20).

Lemma 5. *The unique solution of the linear system of equations* (15)–(20) *exists and it can be computed by using the efficient factorization algorithm.*

Proof. The given proof also defines the constructive algorithm to solve (17)–(20). Since the first part (15) of the discrete ADI scheme defines an explicit algorithm and 1D subproblems (16) in the second part of the scheme are efficiently solved by using the classical factorization algorithm, then it is sufficient to consider in detail the subproblem (17)–(20). For each $(x_{2j}, x_{3k}) \in \bar{\omega}_2 \times \bar{\omega}_3$ the solution is factorized separately in subdomains ω_{11}, ω_{12} and ω_{13}. Let us write equations of the system (17)–(20) as

$$-a_{ijk}U_{i-1,j,k}^{n+1} + c_{ijk}U_{ijk}^{n+1} - b_{ijk}U_{i+1,j,k}^{n+1} = d_{ijk},$$
$$a_{ijk}, b_{ijk}, c_{ijk} \geqslant 0, \quad c_{ijk} \geqslant a_{ijk} + b_{ijk}.$$

1. *Domain* ω_{11}. The solution is presented in the following form:

$$U_{ijk}^{n+1} = \alpha_{ijk}U_{i+1,j,k}^{n+1} + \gamma_{ijk}, \quad 0 \leqslant i < K_1, \tag{23}$$
$$\alpha_{0jk} = 0, \quad \gamma_{0jk} = g_1\big(x_{2j}, x_{3k}, t^{n+1}\big),$$
$$\alpha_{ijk} = \frac{b_{ijk}}{c_{ijk} - a_{ijk}\alpha_{i-1,j,k}}, \qquad \gamma_{ijk} = \frac{d_{ijk} + a_{ijk}\gamma_{i-1,j,k}}{c_{ijk} - a_{ijk}\alpha_{i-1,j,k}}. \tag{24}$$

By induction it can be proved that the estimates $0 \leqslant \alpha_{ijk} \leqslant 1$ are valid.

2. *Domain* ω_{12}. The solution is presented in the following form:

$$U_{i00}^{n+1} = \alpha_{i00}U_{K_100}^{n+1} + \beta_{i00}U_{K_200}^{n+1} + \gamma_{i00}, \quad K_1 < i < K_2. \tag{25}$$

This factorization is done in two steps. First, the solution is written in the form

$$U_{i00}^{n+1} = \tilde{\alpha}_{i00}U_{K_100}^{n+1} + \tilde{\beta}_{i00}U_{i+1,0,0}^{n+1} + \tilde{\gamma}_{i00}, \quad K_1 < i < K_2, \tag{26}$$
$$\tilde{\alpha}_{K_1+1,0,0} = \frac{a_{K_1+1,0,0}}{c_{K_1+1,0,0}}, \qquad \tilde{\beta}_{K_1+1,0,0} = \frac{b_{K_1+1,0,0}}{c_{K_1+1,0,0}}, \qquad \tilde{\gamma}_{K_1+1,0,0} = \frac{d_{K_1+1,0,0}}{c_{K_1+1,0,0}},$$
$$\tilde{\alpha}_{i00} = \frac{a_{i00}}{c_{i00} - a_{i00}\tilde{\beta}_{i-1,0,0}}\tilde{\alpha}_{i-1,0,0}, \qquad \tilde{\beta}_{i00} = \frac{b_{i00}}{c_{i00} - a_{i00}\tilde{\beta}_{i-1,0,0}},$$
$$\tilde{\gamma}_{0k} = \frac{a_{i00}\tilde{\gamma}_{i-1,0,0} + d_{i00}}{c_{i00} - a_{i00}\tilde{\beta}_{i-1,0,0}}. \tag{27}$$

Let us assume that $0 \leqslant \tilde{\alpha}_{i-1,0,0}, \tilde{\beta}_{i-1,0,0} \leqslant 1$ and $\tilde{\alpha}_{k-1,0,0} + \tilde{\beta}_{k-1,0,0} \leqslant 1$. Then it follows from the given formulas that $0 \leqslant \tilde{\beta}_{i00} \leqslant 1$ and $\tilde{\alpha}_{i00} \geqslant 0$. It remains to prove that $\tilde{\alpha}_{i00} \leqslant 1$

and $\tilde{\alpha}_{i00} + \tilde{\beta}_{i00} \leqslant 1$. It follows from simple inequalities

$$a_{i00}\tilde{\alpha}_{i-1,0,0} + b_{i00} + a_{i00}\tilde{\beta}_{i-1,0,0} = a_{i00}(\tilde{\alpha}_{i-1,0,0} + \tilde{\beta}_{i-1,0,0}) + b_{i00} \\ \leqslant a_{i00} + b_{i00} \leqslant c_{i00},$$

that $a_{i00}\tilde{\alpha}_{i-1,00} + b_{i00} \leqslant c_{i00} - a_{i00}\tilde{\beta}_{i-1,0,0}$. Thus we get the estimate

$$0 \leqslant \tilde{\alpha}_{i00} + \tilde{\beta}_{i00} \leqslant 1. \tag{28}$$

The proof by induction is completed.

Next, we compute coefficients α_{i00}, β_{i00} and γ_{i00} in formula (25)

$$\alpha_{K_2-1,0,0} = \tilde{\alpha}_{K_2-1,0,0}, \qquad \beta_{K_2-1,0,0} = \tilde{\beta}_{K_2-1,0,0}, \qquad \alpha_{i00} = \tilde{\alpha}_{i00} + \tilde{\beta}_{i00}\alpha_{i+1,0,0}, \\ \beta_{i00} = \tilde{\beta}_{i00}\beta_{i+1,0,0}, \qquad \gamma_{i00} = \tilde{\gamma}_{i00} + \tilde{\beta}_{i00}\gamma_{i+1,0,0}, \quad i = K_2 - 2, \dots, K_1 + 1. \tag{29}$$

From estimates for $\tilde{\alpha}_{0k}$, $\tilde{\beta}_{0k}$ and (29) it follows that

$$0 \leqslant \alpha_{i00}, \beta_{i00} \leqslant 1, \quad 0 \leqslant \alpha_{i00} + \beta_{i00} \leqslant 1.$$

3. *Domain* ω_{13}. For each $(x_{2j}, x_{3k}) \in \omega_2 \times \omega_3$ the solution is presented in the following form:

$$U_{ijk}^{n+1} = \beta_{ijk}U_{i-1,j,k}^{n+1} + \gamma_{ijk}, \quad K_2 < i \leqslant J_1, \tag{30}$$

$$\beta_{J_1jk} = 0, \qquad \gamma_{J_1jk} = g_2\big(x_{2j}, x_{3k}, t^{n+1}\big),$$

$$\beta_{ijk} = \frac{a_{ijk}}{c_{ijk} - b_{ijk}\beta_{i+1,j,k}}, \qquad \gamma_{ijk} = \frac{d_{ijk} + b_{ijk}\gamma_{i+1,j,k}}{c_{ijk} - b_{ijk}\beta_{i+1,j,k}}.$$

By induction it can be proved that the estimates $0 \leqslant \beta_{ijk} \leqslant 1$ are valid.

Substituting (23)–(30) into equations (19) and (20) we get a linear system of two equations to find $U_{K_100}^{n+1}$, $U_{K_200}^{n+1}$:

$$\begin{cases} A_{11}U_{K_100}^{n+1} + A_{12}U_{K_200}^{n+1} = B_1, \\ A_{21}U_{K_100}^{n+1} + A_{22}U_{K_200}^{n+1} = B_2, \end{cases} \tag{31}$$

where coefficients are defined by

$$A_{11} = \frac{1}{\tau} + \frac{1}{2h_1^2}\big(2 - \alpha_{K_1+1,0,0} - S_h(\alpha_{K_1-1})\big), \quad A_{12} = -\frac{1}{2h_1^2}\beta_{K_1+1,0,0},$$

$$A_{21} = -\frac{1}{2h_1^2}\alpha_{K_2-1,0,0}, \quad A_{22} = \frac{1}{\tau} + \frac{1}{2h_1^2}\big(2 - \beta_{K_2-1,0,0} - S_h(\beta_{K_2+1})\big),$$

$$B_1 = \frac{U^{n+\frac{3}{4}}_{K_1 00}}{\tau} - \frac{1}{2h_1^2}\big(S_h\big(U^n_{K_1-1}\big) - 2U^n_{K_1 00} + U^n_{K_1+1,0,0} + \gamma_{K_1+1,0,0} - S_h(\gamma_{K_1-1})\big),$$

$$B_2 = \frac{U^{n+\frac{3}{4}}_{K_2 00}}{\tau} - \frac{1}{2h_1^2}\big(S_h\big(U^n_{K_2+1}\big) - 2U^n_{K_2 00} + U^n_{K_2-1,0,0} + \gamma_{K_2-1,0,0} - S_h(\gamma_{K_2+1})\big).$$

From the estimates given above we have that

$$\begin{aligned} &2 - \alpha_{K_1+1,0,0} - S_h(\alpha_{K_1-1}) \geqslant \beta_{K_1+1,0,0}, \\ &2 - \beta_{K_2-1,0,0} - S_h(\beta_{K_2+1}) \geqslant \alpha_{K_2-1,0,0}, \end{aligned}$$

thus the determinant of the matrix of system (31) is positive and the unique solution $U^{n+1}_{K_1 00}$, $U^{n+1}_{K_2 00}$ exists. Then the backward factorization step is applied and U^{n+1}_{ijk} are computed. □

It is noted in Hundsdorfer and Verwer (2003) that the stability analysis of such ADI schemes for three dimensional cases is far from easy and even the linear commuting case needs a closer attention. Since in our case operators $\mathcal{A}^h_1$, $\mathcal{A}^h_2$, $\mathcal{A}^h_3$ don't commute, we can't apply the spectral stability analysis.

Here we restrict to writing the stability matrix

$$R = \left(I + \frac{\tau}{2}\mathcal{A}^h_1\right)^{-1}\left(I + \frac{\tau}{2}\mathcal{A}^h_2\right)^{-1}\left(I + \frac{\tau}{2}\mathcal{A}^h_3\right)^{-1}\left(I - \frac{\tau}{2}\big(\mathcal{A}^h_1 + \mathcal{A}^h_2 + \mathcal{A}^h_3\big)\right).$$

It follows from Lemma 4, that

$$\left\| I + \frac{\tau}{2}\mathcal{A}^h_j \right\| \leqslant 1, \quad j = 1, 2, 3.$$

Then it is sufficient to assume that the discrete problem is such that the stability estimate

$$\|R^n\| \leqslant C, \quad n \geqslant 1 \tag{32}$$

is satisfied. The validity of this assumption should be tested for each case of the discrete problem experimentally.

5. Parallel Algorithm

In this section a parallel version of the ADI algorithm (15)–(20) is presented. We note that development of efficient parallel algorithms for implementation of ADI type solvers is quite a non-trivial task. First the factorization algorithm for 3D problems severely depends on data movement in the memory of computers (from and to in different levels of the memory). Only few arithmetical operations are done with each small size portion of data, therefore cost of data movement is very important issue (see, Guo and Lu, 2016; Otero *et al.*, 2020). Second, nonlocality of some parts of the discrete scheme lead to modifications of the standard factorization algorithm and these changes should be resolved efficiently by the parallel algorithm (see Čiegis *et al.*, 2014, where a parallel ADI algorithm is constructed to solve multidimensional parabolic problems with nonlocal boundary conditions). Third, different approaches of parallelization can be used for different architectures of parallel computers. At least three main classes can be considered, including GPU processors (Imankulov *et al.*, 2021; Xue and Feng, 2018; Otero *et al.*, 2020), shared memory processors (multicore processors) and general global memory parallel machines, when implementation of parallel algorithms can be done, e.g. by using MPI library (Čiegis *et al.*, 2014). In this paper we restrict to the construction of parallel algorithms for general distributed memory computers. It is obvious that these algorithms can be efficiently used also for multicore computers and distributed clusters of multicore nodes.

First, in defining a parallel version of the ADI algorithm for the reduced dimension model we restrict to a a simple modification of the standard sequential factorization algorithm. Only two parts are used to decompose the domain Ω_h in each dimension. Therefore the maximum number of parallel processes is restricted to eight processes.

We start by defining a parallel algorithm for the solution of a tridiagonal system of linear equations:

$$\begin{cases} c_0 U_0 - b_0 U_1 = F_0, \\ -a_j U_{j-1} + c_j U_j - b_j U_{j+1} = F_j, \quad j = 1, \dots, J-1, \\ -a_J U_{J-1} + c_J U_J = F_J. \end{cases} \tag{33}$$

Let us assume that two processes are used to solve it in parallel. System (33) is partitioned among both processes. The first process is responsible for the first $(J+1)/2$ equations $0 \leqslant j < (J+1)/2$ and the second process solves the remaining equations. Both processes implement the classical sequential factorization algorithm in parallel but in different directions. The first process applies this algorithm from left-to-right:

$$U_j = \alpha_j U_{j+1} + \beta_j, \quad j = 0, \dots, (J+1)/2 - 1$$

while the second process computes in the opposite direction:

$$U_j = \alpha_j U_{j-1} + \beta_j, \quad j = J, \dots, (J+1)/2.$$

Then both processes exchange coefficients $(\alpha_{\frac{J+1}{2}-1}, \beta_{\frac{J+1}{2}-1})$, $(\alpha_{\frac{J+1}{2}}, \beta_{\frac{J+1}{2}})$, solve 2×2 dimension systems and find solutions $U_{\frac{J+1}{2}-1}$, $U_{\frac{J+1}{2}}$. The second backward part of the factorization algorithm is again implemented in parallel.

Now we are ready to describe the main decomposition details of the parallel version of ADI algorithm (15)–(20).

1. *The case of $p = 2$ processes.* The discrete grid $\bar{\omega}_1$ is splitted into two parts $\bar{\omega}_1 = \omega_{1,p0} \cup \omega_{1,p1}$, where

$$\omega_{1,p0} = \omega_{11} \cup \bar{\omega}_{12}, \qquad \omega_{1,p1} = \omega_{13}.$$

The first process computes the solution on the discrete grid $(\omega_{11} \times \bar{\omega}_2 \times \bar{\omega}_3) \cup \bar{\omega}_{12}$, and the second process computes the solution on the grid $\omega_{13} \times \bar{\omega}_2 \times \bar{\omega}_3$.

Before computations of the explicit step (21), the first process sends the value $U^n_{K_2 00}$ to the second process, and the second process computes the value of $S(U^n_{K_2+1})$ and sends it to the first process. Then both processes compute their parts of $U^{n+\frac{1}{4}}$ in parallel.

The $l = 2, 3$ steps of the ADI scheme (22) don't require any communication between processes and can be done in parallel.

The last step $l = 4$ of the ADI scheme (22) is implemented in the following way. The first process computes the factorization coefficients of (23) and (25), and the second process computes coefficients of (30). These computations can be done in parallel. Then the second process computes the values of $S(\beta^n_{K_2+1})$, $S(\gamma^n_{K_2+1})$ and sends them to the first process. The latter solves system (31) and sends the value $U^{n+1}_{K_2 00}$ to the second process. Then both process compute their parts of U^{n+1} in parallel.

A simple but still accurate complexity model of the parallel ADI algorithm can be constructed

$$T_2 = c\frac{J_1 J_2 J_3}{2} + 2(\alpha + 2\beta), \tag{34}$$

where T_2 is computation time of the one time step of the parallel algorithm, α is data communication start-up time, β defines time required to send one element.

We note that this model takes into account only main computational and data communication costs. In real computations the values of these parameters can vary. One more important detail, i.e. automatic memory caching is also not estimated in this formula. Thus efficiency estimates of such models are only asymptotically valuable, but exactly such information is sufficient in most cases.

Let us define the speed-up S_p and efficiency E_p of the parallel algorithm

$$S_p = \frac{T_0}{T_p}, \qquad E_p = \frac{S_p}{p},$$

where p is the number of processes, T_0 is the computation time of the sequential algorithm.

Since $T_0 = cJ_1 J_2 J_3$ and additional data communication costs are constant, then even for moderate size grids $\omega_1 \times \omega_2 \times \omega_3$ it follows from (34) that

$$S_2 \approx 2, \qquad E_2 \approx 1,$$

i.e. the parallel algorithm is scalable.

2. *The case of $p = 4$ processes.* In addition to splitting the discrete grid $\bar{\omega}_1$, the grid $\bar{\omega}_2$ is also divided into two parts $\bar{\omega}_2 = \omega_{2,p0} \cup \omega_{2,p1}$, where

$$\omega_{2,p0} = \left\{x_{2j},\ 0 \leqslant j \leqslant \frac{J_2}{2}\right\}, \qquad \omega_{2,p1} = \left\{x_{2j},\ \frac{J_2}{2} + 1 \leqslant j \leqslant J_2\right\}.$$

The first process computes the solution on the grid $(\omega_{11} \times \bar{\omega}_{2,p0} \times \bar{\omega}_3) \cup \bar{\omega}_{12}$, the second process computes the solution on the grid $\omega_{13} \times \bar{\omega}_{2,p0} \times \bar{\omega}_3$, the third process computes the solution on the grid $(\omega_{11} \times \bar{\omega}_{2,p1} \times \bar{\omega}_3) \cup \bar{\omega}_{12}$, and the fourth process computes the solution on the grid $\omega_{13} \times \bar{\omega}_{2,p1} \times \bar{\omega}_3$.

Before parallel computations of the explicit step (21), the first process sends the value $U^n_{K_200}$ to the second and fourth process, these two processes compute their part of $S(U^n_{K_2+1})$ and send the obtained values to the first process. In a similar way the first process sends the value $U^n_{K_100}$ to the third process, which computes its part of $S(U^n_{K_1+1})$ and sends the obtained value to the first process.

The second part of data communication is done among two pairs of processes: the first and third processes exchange the values of $U^n_{i,J_2/2,k}$ and $U^n_{i,J_2/2+1,k}$, $0 < i < K_1$, $0 \leqslant k \leqslant J_3$, and the second and fourth processes exchange the values of $U^n_{i,J_2/2,k}$ and $U^n_{i,J_2/2+1,k}$, $K_2 < i < J_1$, $0 \leqslant k \leqslant J_3$. These communications are done in parallel. Then all four processes compute their parts of $U^{n+\frac{1}{4}}$ in parallel.

The $l = 2$ step of the ADI scheme (22) don't require any communication between processes and can be done in parallel.

The $l = 3$ step of the ADI scheme (22) is implemented by using the parallel factorization algorithm, which is described above for the system of linear equations (33). The data communication part between the given pairs of the processes is equivalent to the one defined for the explicit part of the ADI algorithm (21). This step again can be done in parallel.

The last step $l = 4$ of the ADI scheme (22) is implemented as for $p = 2$ processes with one small modification. The second and fourth process compute the values of $S(\beta^n_{K_2+1})$, $S(\gamma^n_{K_2+1})$ and the third process computes the values of $S(\alpha^n_{K_1-1})$, $S(\gamma^n_{K_1-1})$. Then they send these values to the first process. It solves system (31) and sends the values $U^{n+1}_{K_100}$, $U^{n+1}_{K_200}$ to the remaining three process. Then all four process compute their parts of U^{n+1} in parallel.

The complexity model of the parallel ADI algorithm for $p = 4$ processes is given by

$$T_4 = c\frac{J_1J_2J_3}{4} + 4\alpha + \frac{J_1J_3}{2}\beta. \tag{35}$$

It follows from this theoretical model, that now the data communication costs depend linearly on the size of J_1J_3. Thus only for sufficiently large J_2 these costs can be neglected and estimates

$$S_4 \approx 4, \qquad E_4 \approx 1$$

are valid. We get that the parallel algorithm is scalable for sufficiently large J_2 and this specific size depends on both parameters – computational complexity c and data communication costs β.

3. *The case of* $p = 8$ *processes.* In addition to splitting discrete grids $\bar{\omega}_1$, $\bar{\omega}_2$ the discrete grid $\bar{\omega}_3$ is also divided into two parts $\bar{\omega}_3 = \omega_{3,p0} \cup \omega_{3,p1}$, where

$$\omega_{3,p0} = \left\{x_{3k},\ 0 \leqslant k \leqslant \frac{J_3}{2}\right\}, \qquad \omega_{3,p1} = \left\{x_{3k},\ \frac{J_3}{2} + 1 \leqslant k \leqslant J_3\right\}.$$

The parallel algorithm is modified to include data distribution steps in the x_3 dimension, all communications are defined by using analogous formulas given for $p = 4$ case in the x_2 dimension.

The complexity model of the parallel ADI algorithm for $p = 8$ processes is given by

$$T_8 = c\frac{J_1 J_2 J_3}{8} + 4\alpha + \left(\frac{J_1 J_3}{4} + \frac{J_1 J_2}{4}\right)\beta. \tag{36}$$

It follows from this model that for sufficiently large grid sizes, when $J_1 J_2 J_3 \gg J_1(J_2 + J_3)$, the estimates

$$S_8 \approx 8, \qquad E_8 \approx 1$$

are valid and the parallel algorithm is scalable.

Remark 1. Let us consider the case when the number of processors/cores is larger than 8. In order to use all p processors, we modify the constructed parallel algorithm. We assume that p is an even number and factorize it as $p = 2 \times p_2 \times p_3$. The decomposition of the discrete grid Ω_h is done in two steps. First, the $\bar{\omega}_1$ grid is divided into two parts following the algorithm defined for $p \leqslant 8$ processes. Thus the implementation of the parallel algorithm in x_1 dimension remains unchanged. Next, we divide the grids $\bar{\omega}_j$ into p_j, $j = 2, 3$ parts. If $p_j > 2$, then in this dimension the proposed parallel factorization algorithm is changed to a more expensive but general algorithm. We recommend to use Wang's or Cyclic reduction parallel factorization algorithms (Kumar *et al.*, 1994; Imankulov *et al.*, 2021).

6. Computational Experiments

Consider the problem (1)–(4) in the domain Ω with geometry parameters $X_1 = 6$, $X_2 = 1$, $X_3 = 1$. We solve this problem till $T = 1$ with given initial, boundary conditions and the zero source function

$$u^0(x_1, x_2, x_3) = 0, \quad g_1(x_2, x_3, t) = 5te^{-10((x_2-0.5)^2+(x_3-0.5)^2)},$$
$$g_2(x_2, x_3, t) = 2te^{-6((x_2-0.5)^2+(x_3-0.5)^2)}, \quad f(x_1, x_2, x_3, t) = 0.$$

The accuracy of time integration of ADI schemes. First, we have tested the time integration accuracy of the ADI solver for the full dimension model. A uniform space grid Ω_h with $J_1 = 120$, $J_2 = 20$, $J_3 = 20$ is used. Table 1 gives for a sequence of decreasing time step widths τ the errors $e(\tau)$ and the experimental convergence rates $\rho(\tau)$ of the discrete solution for ADI scheme (12) in the maximum norm:

$$e(\tau) = \max_{(x_{1i}, x_{2j}, x_{3k}) \in \Omega_h} \left|U_{ijk}^N - U(x_{1i}, x_{2j}, x_{3k}, T)\right|, \quad \rho(\tau) = \log_2\bigl(e(2\tau)/e(\tau)\bigr),$$

Table 1
Errors $e(\tau)$ and experimental convergence rates $\rho(\tau)$ for the discrete solution of ADI scheme (12) for a sequence of time steps τ.

τ	$e(\tau)$	$\rho(\tau)$
0.002	3.0352×10^{-5}	2.016
0.001	7.5272×10^{-6}	2.012
0.0005	1.8549×10^{-6}	2.021
0.00025	4.411×10^{-7}	2.072

Table 2
Errors $e(\delta)$ of the discrete solution of the reduced dimension model (5)–(11) for a sequence of truncation parameter δ.

	$\delta = 1$	$\delta = 1.5$	$\delta = 2$	$\delta = 2.8$
$e(\delta)$	4.304×10^{-3}	1.748×10^{-4}	7.201×10^{-6}	4.522×10^{-8}
CPU	19.4	29.1	40.8	60.2

where the reference solution $U(x_{1i}, x_{2j}, x_{3k}, T)$ is computed by using the very small time step $\tau = 6.25 \times 10^{-5}$. Then the integration error introduced by the ADI scheme can be measured accurately.

It follows from the presented results, that the accuracy of ADI scheme agrees well with the theoretical prediction.

The accuracy of the reduced dimension model. Next we investigate the accuracy of the reduced dimension model (5)–(11). For the space discretization a uniform grid Ω_h with $J_1 = 300$, $J_2 = 50$, $J_3 = 50$ is used, and integration in time is done with $\tau = 0.001$.

The numerical tests have been performed on the computer with Intel® Xeon® processor E5-2670, 8GB RAM, and the algorithms have been implemented using C++ language.

We note that CPU time for computing the full model solution by using the ADI scheme (12) is 55.9 seconds.

Table 2 gives for a sequence of reduction parameter δ errors $e(\delta)$

$$e(\delta) = \max_{(x_{1i}, x_{2j}, x_{3k}) \in \Omega_h h} \left| U_{ijk}^N - U_{ijk}^N(\delta) \right|$$

of the reduced dimension model (5)–(11) discrete solution in the maximum norm, where U_{ijk}^N is the solution of the ADI scheme (12) and $U_{ijk}^N(\delta)$ is the solution of the reduced dimension ADI scheme (15)–(20). CPU times are also presented for the scheme (15)–(20).

It follows from the presented results, that starting from $\delta = 1$ the accuracy of the reduced dimension model is sufficient for most real world applications, while CPU time for the reduced dimension model is reduced up to 3 times.

The efficiency of the parallel ADI scheme (15)–(20). In this section we present results of the parallel scalability tests. All parallel numerical tests in this work were performed on the computer cluster "HPC Vanagas" at the High Performance Computing Laboratory

Table 3
Scalability analysis of the parallel algorithm. The speed-up S_p and efficiency E_p coefficients for two problems of dimension $450 \times 75 \times 75$ and $600 \times 100 \times 100$.

	$T_p(450)$	$S_p(450)$	$E_p(450)$	$T_p(600)$	$S_p(600)$	$E_p(600)$
$p = 1$	77.4	–	–	254.1	–	–
$p = 4$	17.5	4.42	1.12	57.79	4.40	1.10
$p = 8$	9.85	7.86	0.97	29.2	8.70	1.08

of Vilnius Gediminas technical university. We have used up to 8 cores of Intel® Xeon® processor E5-2630 with 20 cores (2.20 GHz) and 32 GB DDR4 of RAM. The parallel algorithms have been implemented using *MPI* library.

Our goal is to investigate the efficiency of the proposed parallel version of the ADI scheme (15)–(20). We have solved two problems with different sizes of the discrete meshes $450 \times 75 \times 75$ and $600 \times 100 \times 100$. The CPU time T_p, speed-up S_p and efficiency E_p coefficients are presented in Table 3.

Two conclusions follow from the presented results of computational experiments. First, the constructed parallel algorithm scales well even for problems of moderate sizes.

Second, MPI library enables efficient usage of the allocated local memory for each process, thus the efficiency of the parallel algorithm E_p over 1 is achieved.

As a future task, it is interesting to consider the scalability of the modified parallel algorithm from Remark 1, when the number of processes is larger than 8.

7. Conclusions

In this paper we have applied the new efficient approach how to solve the heat equation in 3D domains. This problem is reduced to a hybrid dimension problem, keeping the initial dimension only in some parts and reducing it to one-dimensional equation within the domains in some distance from the base regions. Such mathematical models are typical in industrial installations such as pipelines. We added two additional improvements into this methodology. First, the economical ADI type finite volume scheme is constructed to solve the non-classical heat conduction problem. A finite volume method is used to approximate space differential operators with non classical conjugation conditions between the 3D and 1D parts. It is proved that the ADI scheme is unconditionally stable.

Second, the parallel factorization algorithm is proposed to solve the obtained systems of discrete equations. The ADI scheme leads to non-iterative implementation algorithm and a set of one dimensional linear systems are solved by using the parallel versions of the factorization algorithm for the parallel solving tridiagonal equations systems in each direction. An efficient modification of the basic factorization algorithm is developed to resolve non-local conjugation conditions.

The presented results of numerical experiments confirm both main theoretical conclusions. Due to both modifications the run-time of computations is reduced essentially. First, the hybrid reduced dimension models can be used to simulate heat conduction models for a broad set of domains and CPU time of the serial algorithm is reduced up to factor 3. Second, the developed parallel version of the ADI algorithm enables to achieve effective acceleration of computations and it scales well even for discrete problems of moderate sizes.

References

Amar, H., Givoli, D. (2018). Mixed-dimensional modeling of time-dependent wave problems using the Panasenko construction. *Journal of Theoretical and Computational Acoustics*, 26(3), 1850034. https://doi.org/10.1142/S2591728518500342.

Amosov, A., Panasenko, G. (2018). Partial dimension reduction for the heat equation in a domain containing thin tubes. *Mathematical Methods in the Applied Sciences*, 41(18), 9529–9545. https://doi.org/10.1002/mma.5311.

Čiegis, R., Katauskis, P., Skakauskas, V. (2018). Numerical methods for fractional diffusion. *Nonlinear Analysis: Modelling and Control*, 23(4), 234–250. https://doi.org/10.15388/NA.2018.2.6.

Čiegis, R., Suboč, O., Bugajev, A. (2014). Parallel numerical algorithms for three-dimensional parabolic and pseudoparabolic problems with different boundary conditions. *Nonlinear Analysis: Modelling and Control*, 19(3), 382–395. https://doi.org/10.15388/NA.2014.3.5.

Čiegis, R., Panasenko, G., Pileckas, K., Šumskas, V. (2020). ADI scheme for partially dimension reduced heat conduction models. *Computers & Mathematics with Applications*, 80(5), 1275–1286. https://doi.org/10.1016/j.camwa.2020.06.012.

Eymard, R., Gallouët, T., Herbin, R. (2000). Finite volume methods. *Handbook of Numerical Analysis*, 7, 713–1018. https://doi.org/10.15388/NA.2018.2.6.

Faille, I. (1992). A control volume method to solve an elliptic equation on a two-dimensional irregular mesh. *Computer Methods in Applied Mechanics and Engineering*, 100(2), 275–290. https://doi.org/10.1016/0045-7825(92)90186-N.

Guo, G., Lu, S. (2016). Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional fourth-order diffusion equation. *Computers & Mathematics with Applications*, 71, 1944–1959.

Hundsdorfer, W., Verwer, J. (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics, Vol. 33. Springer, Berlin, Heidelberg, New York, Tokyo. https://doi.org/10.1007/978-3-662-09017-6.

Imankulov, T., Daribayev, B., Mukhambetzhanov, S. (2021). Comparative analysis of parallel algorithms for solving oil recovery problem using CUDA and OpenCL. *International Journal of Nonlinear Analysis and Applications*, 12(1), 351–364.

Kumar, V., Grama, A., Gupta, A., Karypis, G. (1994). *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Benjamin/Cummings, Redwood City, CA.

Otero, B., Rojas, O., Moya, F., Castillo, J.E. (2020). Alternating direction implicit time integrations for finite difference acoustic wave propagation: parallelization and convergence. *Computers and Fluids*, 205, 104584.

Panasenko, G. (1998). Method of asymptotic partial decomposition of domain. *Mathematical Models and Methods in Applied Sciences*, 8(1), 139–156. https://doi.org/10.1142/S021820259800007X.

Panasenko, G. (2005). *Multi-scale Modeling for Structures and Composites*. Springer Netherlands, Dordrecht. https://doi.org/10.1007/1-4020-2982-9.

Panasenko, G., Viallon, M.C. (2015). Finite volume implementation of the method of asymptotic partial domain decomposition for the heat equation on a thin structure. *Russian Journal of Mathematical Physics*, 22(2), 237–263. https://doi.org/10.1134/S1061920815020107.

Samarskii, A.A. (2001). *The Theory of Difference Schemes*. Marcel Dekker, New York. https://doi.org/10.1007/1-4020-2982-9.

Skakauskas, V., Katauskis, P., Čiegis, R. (2018). Modelling of the NO + CO reaction over inhomogeneous surfaces. *Journal of Mathematical Chemistry*, 56(9), 2626–2642. https://doi.org/10.1007/s10910-018-0908-3.

Viallon, M.C. (2013). Error estimate for a 1D–2D finite volume scheme. Comparison with a standard scheme on a 2D non-admissible mesh. *Comptes Rendus Mathematique*, 351(1), 47–51. https://doi.org/10.1016/j.crma.2013.01.011.

Xue, G., Feng, H. (2018). A new parallel algorithm for solving parabolic equations. *Advances in Difference Equations*. 174. https://doi.org/10.1186/s13662-018-1617-8.

R. Čiegis has graduated from Vilnius University Faculty of Mathematics, in 1982. He received the PhD degree from the Institute of Mathematics of Byelorussian Academy of Science in 1985 and the degree of habil. doctor of mathematics from the Institute of Mathematics and Informatics, Vilnius, in 1993. He is a professor and the head of Mathematical Modeling department of Vilnius Gediminas Technical University. His research interests include numerical methods for solving nonlinear PDE, parallel numerical methods, mathematical modelling in nonlinear optics, porous media flows, technology, image processing, biotechnology.

R. Čiegis has graduated from Kaunas Polytechnical Institute Faculty of Chemical Engineering in 1982 and Vilnius University Kaunas Faculty, in 1989. He received the PhD degree in economics from the Vilnius University in 1995 and the degree of habil. doctor of management from the Kaunas Vytautas Magnus University in 2002. He is a professor of Vilnius University. His research interests include environmental economics, sustainable economic development, macroeconomics, regional economics.

O. Suboč has graduated from Vilnius University Faculty of Mathematics, in 1998. She received the PhD degree from the Institute of Mathematics and Informatics and Vytautas Magnus University, in 2002. She is an associated professor in Vilnius Gediminas Technical University. Her research interests include numerical methods for solving nonlinear PDE, mathematical modelling, technology, image processing, non-local conditions, biotechnology.