BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES,
DOI: 10.24425/bpasts.2025.154276

Kronecker product-based solvers for higher-order finite element method Navier-Stokes simulations

TOMASZ SŁUŻALEC[1]*, MARCIN ŁOŚ[1], MATEUSZ DOBIJA[2]***, MACIEJ PASZYŃSKI[1]******

[1]AGH University of Krakow, al. Mickiewicza 30, 30-059 Krakow, Poland
[2]Jagiellonian University, Łojasiewicza 11, 30-348 Krakow, Poland

Abstract. Transient time-dependent problems solved with higher-order finite element methods and time integration schemes sometimes encounter instabilities in time steps due to varying model parameters. This problem is commonly illustrated on a transient cavity flow modeled with Navier-Stokes equations, where for large Reynolds number, finite element discretizations $\mathbf{B}u = f$ become unstable. The instability comes from the discrete inf-sup condition not fulfilled by the Galerkin method. To stabilize time steps, we employ a Petrov-Galerkin method $\mathbf{B}^T\mathbf{W}x = \mathbf{W}^T f$ with optimal test functions. However, this method commonly has two disadvantages. First, having a larger test space fixed, we must compute the matrix of coefficients of the optimal test functions $\mathbf{W}$ on the fly, which requires solving a system of linear equations $\mathbf{GW} = \mathbf{B}$ with proper Gram matrix $\mathbf{G}$ each time step for varying model parameters. Second, the matrix of coefficients of optimal test functions is dense,
and thus, the cost of multiplying it to other matrices $\mathbf{B}^T\mathbf{W}$ (which is needed) is high. To overcome these problems, we explore the Kronecker product structure of the matrix of coefficients of the optimal test functions $\mathbf{G}$ as well as of the matrices $\mathbf{B}$ resulting from the variational splitting of the time-integration scheme. Our solver can be successfully applied to the high Reynolds number Navier-Stokes equations.

Key words: partial differential equations, finite element method, Navier-Stokes equations, variational splitting, Petrov-Galerkin method, Kronecker product solver

1. INTRODUCTION

We focus on non-stable time-dependent simulations using the example of the Navier-Stokes problem. The problem of stabilization and development of an efficient solver for Navier-Stokes equations is an important research topic [1, 2, 3, 4, 5, 6]. We propose a novel combination of the Petrov-Galerkin method, optimal test functions, and direction splitting solver to stabilize the simulation for large Reynolds numbers with a linear computational cost. The direction splitting solvers has multiple applications, from propagation of EM waves [7], modeling of flow and transport with advection-diffusion models [8], elastic wave propagation [9]. The following are the novelties of this paper. First, we employ the Petrov-Galerkin method with optimal test functions [10, 11, 12, 13] to stabilize each non-stationary Navier-Stokes simulation time step. Second, we combine the stabilization with the Petrov-Galerkin method using the optimal test functions with the direction splitting solver [14, 15, 16, 17], resulting in a Kronecker product structure of the problem matrices in every time step. Third, we also show that the matrix of the coefficients of the optimal test functions has a Kronecker product form [18, 19, 20]. For the reader convenience, in the Appenix A in Table 4 we provide a list of the most important symbols.

Let us focus on the Eriksson-Johnson model problem [21]. Given the unit square domain $\Omega = (0,1)^2$ and the convection vector $\beta = (\beta_x, \beta_y) = (1,0)$ consider the PDE:

$$\beta_x \frac{\partial u}{\partial x} + \beta_y \frac{\partial u}{\partial x} - \varepsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0. \tag{1}$$

The problem is driven by the inflow Dirichlet boundary condition

$$u(x,y) = 0, \text{ at } x,y \in \{0,1\}, \tag{2}$$

$$u(x,y) = g(x,y) = \sin(\pi y), \text{ at } x = 0. \tag{3}$$

It develops a boundary layer of width ε at the outflow $x = 1$.

The weak form reads: Find $w \in H_0^1(\Omega)$

$$\begin{aligned} b(w,v) =& l(v), \forall v \in H_0^1(\Omega), \\ b(w,v) =& \left(\frac{\partial w}{\partial x}, v \right)_\Omega + \varepsilon \left(\frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} \right)_\Omega + \varepsilon \left(\frac{\partial w}{\partial y}, \frac{\partial v}{\partial y} \right)_\Omega, \\ l(v) =& -(\sin(\pi y), v)_\Omega - \varepsilon \left(\sin(\pi y), \frac{\partial v}{\partial x} \right)_\Omega \\ & + \varepsilon \left((1-x)\pi \cos(\pi y), \frac{\partial v}{\partial y} \right)_\Omega. \end{aligned} \tag{4}$$

The right-hand side is a result of the shift of the Dirichlet boundary conditions, and the solution is $u(x,y) = w(x,y) + (1-x)sin(\pi y)$. The Galerkin method generally does not provide a correct numerical solution for the Eriksson-Johnson problem. For example, let us consider the discretization with B-spline basis functions with knot vectors and points.

```
knot_x = [0 0 0 1 2 3 4 5 6 7 8 9 10 10 10];
points_x = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
0.9 1];
knot_y =[0 0 0 1 2 3 4 4 4];
```

*e-mail: sluzalec@agh.edu.pl
**e-mail: los@agh.edu.pl
***e-mail: mateusz.dobija@doctoral.uj.edu.pl
****e-mail: maciej.paszynski@agh.edu.pl

```
points_y=[0 0.25 0.5 0.75 1];
```

They correspond to the basis functions illustrated in Figure 1. The basis functions along x axis are obtained by introducing knot points $\xi_i = points\_x[knot_x[i]+1]$ into the recursive formula (5),

$$B_{i,0}(\xi) = 1 \text{ if } \xi_i \le \xi \le \xi_{i+1}, \text{ otherwise } 0,$$
$$B_{i,p}(\xi) = \frac{\xi-\xi_i}{\xi_{i+p}-\xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_i} B_{i+1,p-1}(\xi) \quad (5)$$

for the order p defined as the number of repetitions of the first $knot\_x[1]$ minus one, assuming that the subsequent knots inserted into the denominator must be different. If they are not distinct, the given term is zero.

Similarly, the basis functions along y axis are obtained by introducing knot points $\xi_i = points\_y[knot_y[i]+1]$ into the recursive formula (5). Tensor products of the one-dimensional basis result in the two-dimensional basis functions.

$$\{B^{xy}_{ij,p}(x,y) = B^x_{i,p}(x) B^y_{j,p}(y)\}_{i=1,\ldots,N_x; j=1,\ldots,N_y} \quad (6)$$

In the Galerkin method, these basis functions define the trial and test spaces. The problem matrices are defined as follows

$$\mathbf{B}_{n,m} = \left(\frac{\partial B^x_{i,p}}{\partial x} B^y_{j,p}, B^{xy}_{kl,p}\right)_\Omega + \varepsilon \left(\frac{\partial B^x_{i,p}}{\partial x} B^y_{j,p}, \frac{\partial B^x_{k,p}}{\partial x} B^y_{l,p}\right)_\Omega + \varepsilon \left(B^x_{i,p} \frac{\partial B^y_{j,p}}{\partial y}, B^x_{k,p} \frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega, \quad (7)$$

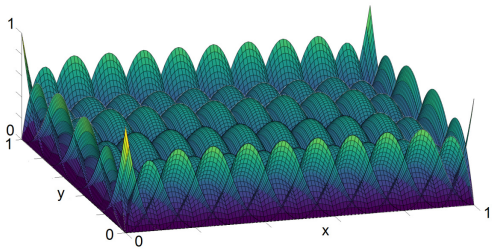
where $n = i+(j-1)N_y$, $m = k+(l-1)N_y$, and the right-hand side vector is

$$f_m = -\left((\sin(\pi y), B^{xy}_{kl,p}\right)_\Omega - \varepsilon \left(\sin(\pi y), \frac{\partial B^x_{k,p}}{\partial x} B^y_{l,p}\right)_\Omega + \varepsilon \left((1-x)\pi\cos(\pi y), B^x_{k,p} \frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega. \quad (8)$$

In the Galerkin method, we solve $\mathbf{B}x = f$, and both trial and test spaces are equal; see Figure 1. As Figure 2 illustrates, the Galerkin method generates unwanted oscillations. The reason why the Galerkin method does not work results from the fact that the coercivity constant α goes to zero as ε goes to zero. Consequently, the error bounds provided by the Céa Lemma are very high, and a good quality solution is not guaranteed.

In conclusion, we need a good enough trial space U_h and carefully selected basis functions from the test space V_h, so they realize supremum for α. To find the correct solution, we will employ the Petrov-Galerkin method. The Petrov-Galerkin method allows using different trial and test spaces. For example, we can employ the same trial space defined by the same knot vectors and points.

```
knot_x = [0 0 0 1 2 3 4 5 6 7 8 9 10 10 10];
points_x = [0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
0.9 1];
knot_y =[0 0 0 1 2 3 4 4 4];
points_y=[0 0.25 0.5 0.75 1];
```



$$= U_h = V_h$$

Fig. 1. Basis functions for trial and test spaces. **Galerkin method** (trial space) $U_h = V_h$ (test space). We need to test in a test space larger than the trial space.

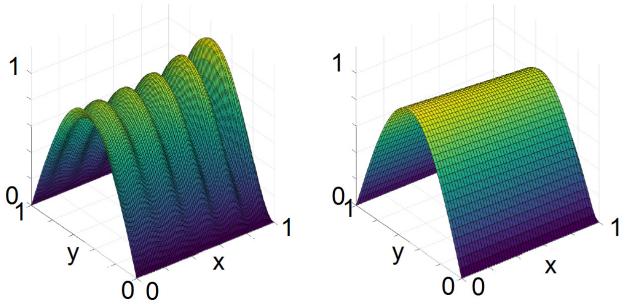


Fig. 2. Uniform mesh $\varepsilon = 0.01$, **Left panel:** Galerkin solution, **Right panel:** exact solution.

We can seek better quality test functions in the larger test space, defined by a larger number of basis functions, by introducing the following knot and point vectors

```
knot_test_x = [0 0 0 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 20 20];
points_test_x = [0 0.05 0.1 0.15 0.2 0.25 0.3
0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8
0.85 0.9 0.95 1];
knot_test_y =[0 0 0 1 2 3 4 4 4];
points_test_y=[0 0.25 0.5 0.75 1];
```

The basis of the test and trial spaces for the Petrov-Galerkin formulation are illustrated in Figure 3.

In our version of the Petrov-Galerkin method, we compute

$$\mathbf{B}^T \mathbf{W} x = \mathbf{W}^T f \quad (9)$$

where $\mathbf{W} = [w^1, \cdots, w^n]$ is the matrix of optimal test functions coefficients, and it can be computed from

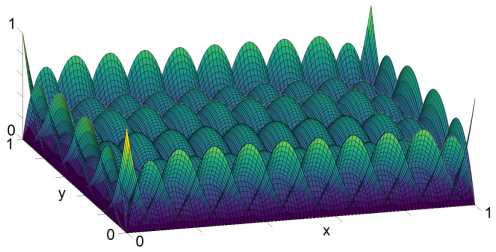
$$\mathbf{GW} = \mathbf{B}$$

where $\mathbf{G}$ is the Gram matrix of selected inner product, namely

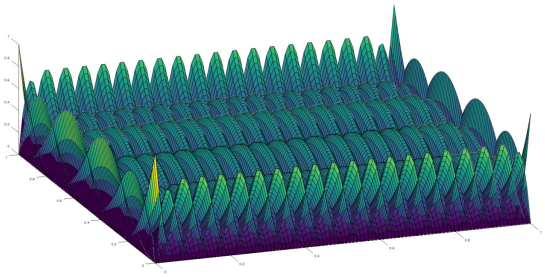
$$\mathbf{G}_{n,m} = \left(B^{xy}_{ij,p}, B^{xy}_{kl,p}\right)_\Omega + \varepsilon \left(\frac{\partial B^x_{i,p}}{\partial x} B^y_{j,p}, \frac{\partial B^x_{k,p}}{\partial x} B^y_{l,p}\right)_\Omega + \varepsilon \left(B^x_{i,p} \frac{\partial B^y_{j,p}}{\partial y}, B^x_{k,p} \frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega, \quad (10)$$

The optimal test space $V_h^{opt} = span\{w^1, \cdots, w^n\} \subset V_h$, $\#V_h = \#U_h$.

The optimal test functions of the Petrov-Galerkin method are obtained by projecting from the larger test space. The



$= U_h,$



$= V_h$

Fig. 3. Basis functions for trial and test spaces. **Petrov-Galerkin** method (trial space) $U_h \neq V_h$ (test space).

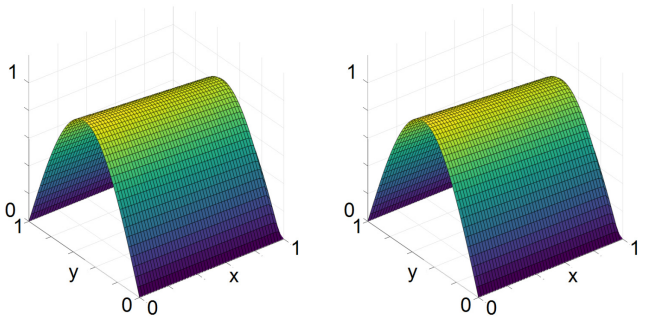


Fig. 4. Uniform mesh $\varepsilon = 0.01$, **Left panel:** solution of the Petrov-Galerkin problem, **Right panel:** exact solution.

Petrov-Galerkin method with optimal test functions can solve the Eriksson-Johnson problem correctly; see Figure 4.

2. GENERALIZATION TO TIME-DEPENDENT PROBLEMS DRIVEN BY THE NAVIER-STOKES PROBLEM

In this section, we consider the stabilization of time-dependent problems using the example of Navier-Stokes equations.

- We will employ an unconditionally stable second-order time discretization scheme developed by Petar Minev and Jean-Luc Guermound [14].
- For spatial discretization, we will apply higher order B-spline basis functions from the isogeometric analysis [22].
- We will apply the Petrov-Galerkin method with optimal test functions in every time step of the Navier-Stokes computations. This involves all the sub-steps, including the pressure and velocity updates. We will stabilize the problem by solving $\mathbf{B}^T\mathbf{W}x = \mathbf{W}^T f$ instead of $\mathbf{B}x = f$.
- The disadvantage of using the Petrov-Galerkin method with optimal test functions is that the matrix of optimal test functions coefficient $\mathbf{W}$ may change with each time step. Thus, $\mathbf{W}$ has to be computed in every time step, which brings the additional cost of solving $\mathbf{GW} = \mathbf{B}$. We speed up this process by exploring the Kronecker product structure of the matrix $\mathbf{W} = W^x \otimes W^y$.
- Another disadvantage of using the Petrov-Galerkin method with optimal test functions is that the matrix of coefficients of optimal test functions $\mathbf{W}$ is dense. As a remedy for this problem, we will explore the Kronecker product structure of the problem matrices $\mathbf{B}^T$ resulting from the application of the time integration skim, allowing for the variational splitting. Then, the multiplication $(B_q^x \otimes B_q^y)^T (W_q^x \otimes W_q^y) = ((B_q^x)^T W_q^x) \otimes ((B_q^y)^T W_q^y)$.

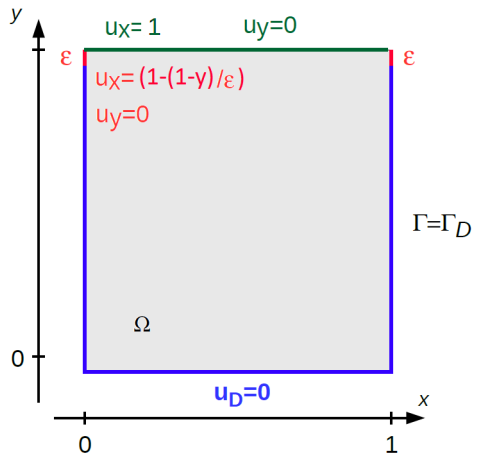


Fig. 5. Non-stationary cavity flow problem.

Let us focus on the non-stationary cavity flow problem described with the Navier-Stokes equation for the incompressible fluid; see Figure 5. The Dirichlet boundary condition drives the cavity flow for the velocity $u_x = 1$, $u_y = 0$ on the top boundary. On the remaining parts of the boundary, the velocity is equal to 0, and the ε thick transition zone in the left and right top corners ensures the possibility of a weak formulation. This problem exhibits pressure singularities at the two top corners.

Let $\Omega = (0,1)^2$ be the open boundary and $I = [0,T] \subset \mathbb{R}$ be the time interval. In this problem's time-dependent version, we introduce the variable Reynolds number $Re(t) = 1 + 1999t$ for $t \in [0,2]$. The problem reads: Find velocity u and pressure field p such that:

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \dfrac{1}{Re(t)}\Delta\mathbf{u} + \nabla p = 0 & \text{in } \Omega\times I, \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \Omega\times I, \\ \mathbf{u} = h & \text{in } \Gamma\times I, \\ \mathbf{u}(0) = 0 & \text{in } \Omega, \end{cases} \tag{11}$$

where

$$h(x,y) = \begin{cases} 0 & x \in (0,1),\ y = 0 \\ 0 & x \in \{0,1\},\ y \in (0,1-\varepsilon) \\ 1 & x \in (0,1),\ y = 1 \\ \left(1 - \dfrac{(1-y)}{\varepsilon}\right) & x \in \{0,1\},\ y \in (1-\varepsilon,1) \end{cases} \tag{12}$$

The Γ denotes the boundary of the spatial domain Ω.

To solve the cavity flow problem, we will use a fast time integration scheme proposed by [14]. We consider the alternating-directions implicit (ADI) method with the Peaceman-Reachford scheme applied to the velocity update. The time interval is uniformly partitioned $I = [0,T]$ such that

$$0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T,$$

and we denote the time step $\tau := t_{n+1} - t_n$ $n = 0,\ldots,N-1$. We calculate predicted value of the pressure in order to estimate the value of the velocity. After we compute the velocity we can correct the pressure.

- *Pressure predictor*
  We define $\tilde{p}^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n-\frac{1}{2}}$, $\forall n = 0,\ldots,N-1$ and let $p^{-\frac{1}{2}} = p_0$ and $\phi^{-\frac{1}{2}} = 0$.
- *Velocity update*

$$\begin{aligned} u^{n+\frac{1}{2}} - \frac{\tau}{2}\frac{1}{Re(t)}\partial_{xx}u^{n+\frac{1}{2}} &= u^n + \frac{\tau}{2}\frac{1}{Re(t)}\partial_{yy}\mathbf{u}^n - \frac{\tau}{2}\nabla\tilde{p}^{n+\frac{1}{2}} \\ &+ \frac{\tau}{2}\mathbf{f}^{n+\frac{1}{2}} + (\mathbf{u}^n\cdot\nabla)\,\mathbf{u}^n, \\ u^{n+\frac{1}{2}} &= 0 \text{ in } \Gamma_x, \\ u^{n+1} - \frac{\tau}{2}\frac{1}{Re(t)}\partial_{yy}u^{n+1} &= u^{n+\frac{1}{2}} + \frac{\tau}{2}\frac{1}{Re(t)}\partial_{xx}\mathbf{u}^{n+\frac{1}{2}} \\ &- \frac{\tau}{2}\nabla\tilde{p}^{n+\frac{1}{2}} + \frac{\tau}{2}\mathbf{f}^{n+\frac{1}{2}}, \\ u^{n+1} &= 0 \text{ in } \Gamma_y, \end{aligned}$$

- *Penalty step*

$$\begin{aligned} \psi - \frac{1}{Re(t)}\partial_{xx}\psi &= -\frac{1}{\tau}\nabla\cdot\mathbf{u}^{n+1}, \quad \partial_x\psi = 0 \quad \text{in } \Gamma_x, \\ \phi^{n+\frac{1}{2}} - \frac{1}{Re(t)}\partial_{yy}\phi^{n+\frac{1}{2}} &= \psi, \quad \partial_y\phi^{n+\frac{1}{2}} = 0 \quad \text{in } \Gamma_y, \end{aligned}$$

- *Pressure update* $p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+\frac{1}{2}} - \chi\nabla\cdot\left(\frac{1}{2}(u^{n+1}+u^n)\right)$.

To solve the problem numerically, we introduce the weak form

$$\begin{aligned} (u^{n+\frac{1}{2}},\mathbf{v}) + \frac{\tau}{2}\frac{1}{Re(t)}(\partial_x u^{n+\frac{1}{2}},\partial_x\mathbf{v}) \\ = (u^n,\mathbf{v}) - \frac{\tau}{2}\frac{1}{Re(t)}(\partial_y\mathbf{u}^n,\partial_y\mathbf{v}) \\ - \frac{\tau}{2}(\nabla\tilde{p}^{n+\frac{1}{2}},\mathbf{v}) + \frac{\tau}{2}(\mathbf{u}^n\cdot\nabla)\,\mathbf{u}^n + \frac{\tau}{2}(\mathbf{f}^{n+\frac{1}{2}},\mathbf{v}), \\ \forall v \in V \end{aligned}$$

$$\begin{aligned} (u^{n+1},\mathbf{v}) + \frac{\tau}{2}\frac{1}{Re(t)}(\partial_y u^{n+1},\partial_y\mathbf{v}) \\ = (u^{n+\frac{1}{2}},\mathbf{v}) - \frac{\tau}{2}\frac{1}{Re(t)}(\partial_x\mathbf{u}^{n+\frac{1}{2}},\partial_x\mathbf{v}) \\ - \frac{\tau}{2}(\nabla\tilde{p}^{n+\frac{1}{2}},\mathbf{v}) + \frac{\tau}{2}(\mathbf{f}^{n+\frac{1}{2}},\mathbf{v}), \\ \forall v \in V \end{aligned}$$

The weak formulations for the penalty/update formulas read

$$\begin{aligned} (\psi,w) + (\partial_x\psi,\partial_x w) &= -\frac{1}{\tau}(\nabla\cdot\mathbf{u}^{n+1},w), \quad \forall w \in V \\ (\phi^{n+\frac{1}{2}},w) + (\partial_y\phi^{n+\frac{1}{2}},\partial_y w) &= (\psi,w) \quad \forall w \in V. \end{aligned}$$

Now, we reformulate this fast time integration scheme for Navier-Stokes equations in the matrix form. To achieve this, we introduce discretization with trial B-spline basis functions, and we test with optimal test functions computed for $1/Re(t)$. We introduce the following matrices

$$\begin{aligned} M^x_{i,k} &= \left(B^x_{i,p},B^x_{k,p}\right)_\Omega, \quad K^x_{i,k} = \left(\frac{\partial B^x_{i,p}}{\partial x},\frac{\partial B^x_{k,p}}{\partial x}\right)_\Omega, \\ R^x_{i,k} &= \left(\frac{\partial B^x_{i,p}}{\partial x},B^x_{k,p}\right)_\Omega, \quad M^y_{j,l} = \left(B^y_{j,p},B^y_{l,p}\right)_\Omega, \\ K^y_{j,l} &= \left(\frac{\partial B^y_{j,p}}{\partial y},\frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega, \quad R^y_{j,l} = \left(\frac{\partial B^y_{j,p}}{\partial y},B^y_{l,p}\right)_\Omega. \end{aligned} \tag{13}$$

We summarize the matrix formulation for the velocity

$$\begin{aligned} &\underbrace{\begin{bmatrix} (M^x + \frac{\tau}{2}\frac{1}{Re(t)}K^x)\otimes M^y & 0 \\ 0 & (M^x + \frac{\tau}{2}\frac{1}{Re(t)}K^x)\otimes M^y \end{bmatrix}}_{\mathbf{B}^{n+\frac{1}{2}}} \underbrace{\begin{bmatrix} u_1^{n+\frac{1}{2}} \\ u_2^{n+\frac{1}{2}} \end{bmatrix}}_{u^{n+\frac{1}{2}}} \\ = &\underbrace{\begin{bmatrix} \left(M^x\otimes(M^y - \frac{\tau}{2}\frac{1}{Re(t)}K^y) - \frac{\tau}{2}(u_1^nK^x + u_2^nK^y)\right)u_1^n \\ \left(M^x\otimes(M^y - \frac{\tau}{2}\frac{1}{Re(t)}K^y) - \frac{\tau}{2}(u_1^nK^x + u_2^nK^y)\right)u_2^n \end{bmatrix}}_{} \\ &\underbrace{- \frac{\tau}{2}\begin{bmatrix} R^x\otimes M^y & 0 \\ 0 & M^x\otimes R^y \end{bmatrix}\begin{bmatrix} \tilde{p}^{n+\frac{1}{2}} \\ \tilde{p}^{n+\frac{1}{2}} \end{bmatrix} + \frac{\tau}{2}\begin{bmatrix} F_1^{n+\frac{1}{2}} \\ F_2^{n+\frac{1}{2}} \end{bmatrix}}_{f^{n+\frac{1}{2}}}, \end{aligned}$$

$$\begin{aligned} &\underbrace{\begin{bmatrix} M^x\otimes(M^y + \frac{\tau}{2}\frac{1}{Re(t)}K^y) & 0 \\ 0 & M^x\otimes(M^y + \frac{\tau}{2}\frac{1}{Re(t)}K^y) \end{bmatrix}}_{\mathbf{B}^{n+1}} \underbrace{\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \end{bmatrix}}_{u^{n+1}} \\ = &\underbrace{\begin{bmatrix} \left((M^x - \frac{\tau}{2}\frac{1}{Re(t)}K^x)\otimes M^y - \frac{\tau}{2}(u_1^{n+\frac{1}{2}}K^x + u_2^nK^y)\right)u_1^{n+\frac{1}{2}} \\ \left((M^x - \frac{\tau}{2}\frac{1}{Re(t)}K^y)\otimes M^y - \frac{\tau}{2}(u_1^{n+\frac{1}{2}}K^x + u_2^nK^y)\right)u_2^{n+\frac{1}{2}} \end{bmatrix}}_{} \\ &\underbrace{- \frac{\tau}{2}\begin{bmatrix} M^x\otimes R^y & 0 \\ 0 & R^x\otimes M^y \end{bmatrix}\begin{bmatrix} \tilde{p}^{n+\frac{1}{2}} \\ \tilde{p}^{n+\frac{1}{2}} \end{bmatrix} + \frac{\tau}{2}\begin{bmatrix} F_1^{n+\frac{1}{2}} \\ F_2^{n+\frac{1}{2}} \end{bmatrix}}_{f^{n+1}}, \end{aligned}$$

Each of these sub-steps (14)-(14) results in two sub-problems.

We also introduce the penalty steps

$$(M^x+K^x)\otimes M^y\psi = -\frac{1}{\tau}(R^x\otimes M^y u_1^{n+\frac{1}{2}} + M^x\otimes R^y u_2^{n+\frac{1}{2}}),$$
$$M^x\otimes(M^y+K^y)p^{n+\frac{1}{2}} = M^x\otimes M^y\psi.$$

$$M^x\otimes(M^y+K^y)\psi = -\frac{1}{\tau}(M^x\otimes R^y u_1^{n+1} + R^x\otimes M^y u_2^{n+1}),$$
$$(M^x+K^x)\otimes M^y p^{n+1} = M^x\otimes M^y\psi.$$

Each of the systems

$$\mathbf{B}^{n+\frac{1}{2}}u^{n+\frac{1}{2}} = f^{n+\frac{1}{2}}, \quad \mathbf{B}^{n+1}u^{n+1} = f^{n+1}, \tag{14}$$

solved for a big Reynolds number Re results in unstable behavior. It is similar to ε in the advection-dominated diffusion problem. At high Reynolds numbers, the coefficient $\frac{1}{Re}$ becomes very small, which can negatively impact the numerical stability. During the simulations with the time-dependent Navier-Stokes equations, the Re number can change from one step to another; thus, it requires stabilization. It can be obtained by solving stabilized formulations

$$\begin{aligned}(\mathbf{B}^{n+\frac{1}{2}})^T\mathbf{W}^{n+\frac{1}{2}}u^{n+\frac{1}{2}} &= (\mathbf{W}^{n+\frac{1}{2}})^T f^{n+\frac{1}{2}},\\ (\mathbf{B}^{n+1})^T\mathbf{W}^{n+1}u^{n+1} &= (\mathbf{W}^{n+1})^T f^{n+1}\end{aligned} \tag{15}$$

The matrices of coefficients of the optimal test functions $\mathbf{W}^{n+\frac{1}{2}}$, $\mathbf{W}^{n+1}$ can be obtained by solving

$$\mathbf{G}^{n+\frac{1}{2}}\mathbf{W}^{n+\frac{1}{2}} = \mathbf{B}^{n+\frac{1}{2}}, \quad \mathbf{G}^{n+1}\mathbf{W}^{n+1} = (\mathbf{B}^{n+1}) \tag{16}$$

where

$$\underbrace{\begin{bmatrix}(M^x+K^x)\otimes M^y & 0\\ 0 & (M^x+K^x)\otimes M^y\end{bmatrix}}_{\mathbf{G}^{n+\frac{1}{2}}}\underbrace{\begin{bmatrix}\mathbf{W}_1^{n+\frac{1}{2}}\\ \mathbf{W}_2^{n+\frac{1}{2}}\end{bmatrix}}_{W^{n+\frac{1}{2}}} \tag{17}$$
$$= \underbrace{\begin{bmatrix}(M^x+\frac{\tau}{2}\frac{1}{Re(t)}K^x)\otimes M^y & 0\\ 0 & (M^x+\frac{\tau}{2}\frac{1}{Re(t)}K^x)\otimes M^y\end{bmatrix}}_{\mathbf{B}^{n+\frac{1}{2}}},$$

and

$$\underbrace{\begin{bmatrix}M^x\otimes(M^y+K^y) & 0\\ 0 & M^x\otimes(M^y+K^y)\end{bmatrix}}_{\mathbf{G}^{n+1}}\underbrace{\begin{bmatrix}\mathbf{W}_1^{n+1}\\ \mathbf{W}_2^{n+1}\end{bmatrix}}_{W^{n+\frac{1}{2}}} \tag{18}$$
$$= \underbrace{\begin{bmatrix}M^x\otimes(M^y+\frac{\tau}{2}\frac{1}{Re(t)}K^y) & 0\\ 0 & M^x\otimes(M^y+\frac{\tau}{2}\frac{1}{Re(t)}K^y)\end{bmatrix}}_{\mathbf{B}^{n+1}} \tag{19}$$

Each of these sub-steps (17)-(18) results in two sub-problems. There is a significant difference in the numerical results for the non-stationary cavity flow problem modeled by the Navier-Stokes equations, depending on whether we use the stabilization using the Petrov-Galerking formulation with the optimal test functions. The results obtained by the classical Galerkin method remind us of chaotic behavior compared to the Petrov-Galerkin method with the optimal test functions; see Figure 6. This is related to the fact that high Reynolds results in very small values of the coefficient $\frac{1}{Re}$, which negatively impacts the numerical stability.

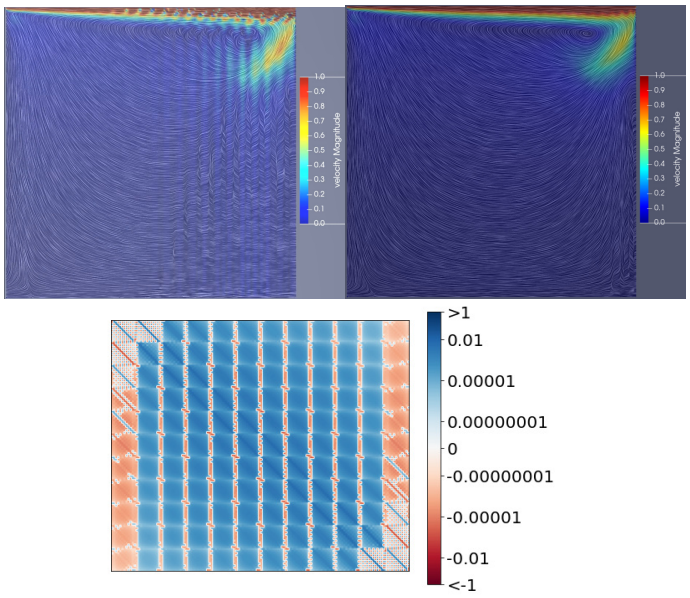


Fig. 6. Left panel: The unstable behavior of the cavity flow problem is solved with the Galerkin formulation for $Re = 2000$. The instability results in some unphysical oscillations visible in the velocity profile. **Right panel**: The stabilized behavior of the cavity flow problem was obtained using the Petrov-Galerkin formulation with the optimal test functions computed for $Re = 2000$. The stabilization removes the oscillations from the velocity profile. **Bottom panel**: The matrix of coefficients of the optimal test functions computed for $Re = 2000$ This matrix has been decomposed into the Kronecker product structure and employed to stabilize the solution.

3. SOLVER ALGORITHM USING KRONECKER PRODUCT STRUCTURE

Each of the systems (15) have the following structure

$$(B_q^x\otimes B_q^y)^T\mathbf{W_q}u_q = \mathbf{W_q}^T f_q \quad , \tag{20}$$

where

$$(G_q^x\otimes G_q^y)\mathbf{W}_q = B_q^x\otimes B_q^y \quad . \tag{21}$$

Here we introduce the Gram matrix for each sub-problem having a Kronecker product structure as well. From the properties of the Kronecker product we have

$$\begin{aligned}\mathbf{W}_q &= (G_q^x\otimes G_q^y)^{-1}B_q^x\otimes B_q^y\\ &= (G_q^x)^{-1}\otimes(G_q^y)^{-1}(B_q^x\otimes B_q^y),\\ &= (G_q^x)^{-1}B_q^x\otimes(G_q^y)^{-1}B_q^y\end{aligned} \tag{22}$$

thus $\mathbf{W}_q$ has the Kronecker product structure $\mathbf{W}_q = (W_q^x \otimes W_q^y)$, and our original system

$$\left((B_q^x)^T W_q^x \otimes (B_q^y)^T W_q^y\right) u_q = \left((W_q^x)^T \otimes (W_q^y)^T\right) f_q \tag{23}$$

The system (23) can be solved in two steps

$$\left((B_q^x)^T W_q^x\right) Z_q = \left((W_q^x)^T \otimes (W_q^y)^T\right) f_q \tag{24}$$

$$\left((B_q^y)^T W_q^y\right) U_q = Z_q^T \tag{25}$$

The matrices $B_q^x \in \mathscr{M}^{\hat{N}_x \times N_x}$, $W_q^x \in \mathscr{M}^{\hat{N}_x \times N_x}$, $Z_q \in \mathscr{M}^{N_x \times N_y}$, and $B_q^y \in \mathscr{M}^{\hat{N}_y \times N_y}$, $W_q^y \in \mathscr{M}^{\hat{N}_y \times N_y}$, and U_q is the solution vector u_q projected onto $\mathscr{M}^{N_y \times N_x}$. Here $N_x \times N_y$ is the dimension of the trial space, and $\hat{N}_x \times \hat{N}_y$ is the dimension of the test space. In general, the matrices B_q^x, B_q^y constructed from B-splines of order p are $p+2$ diagonal, but the matrices W_q^x and W_q^y are dense.

From the properties of the Kronecker product, the right-hand side in (24) can be computed as $\left((W_q^x)^T \otimes (W_q^y)^T\right) f_q = (W_q^y)^T F_q (W_q^x)$, where F_q is the projection of f_q vector into $\mathscr{M}^{\hat{N}_y \times \hat{N}_x}$. The cost of formulation of the right-hand side is then $\mathscr{O}(N_x \hat{N}_x^2 + N_y \hat{N}_y^2)$. The cost of solving system (24) is $\mathscr{O}(N_x^2 \hat{N}_x)$ and the cost of solving system (25) is $\mathscr{O}(N_y^2 \hat{N}_y)$.

The Kronecker product components of $\mathbf{W}_x$ are computed from $W_q^x = (G_q^x)^{-1} B_q^x$ and $W_q^y = (G_q^y)^{-1} B_q^y$, thus, they can be obtained from factorizations of

$$\begin{aligned} G_q^x W_q^x &= B_q^x \\ G_q^y W_q^y &= B_q^y \end{aligned} \tag{26}$$

Since B_q^x, B_q^y, G_q^x, and G_q^y constructed from B-splines have a multi-diagonal structure, these systems can be solved in linear computational costs $\mathscr{O}(N_x)$ or $\mathscr{O}(N_y)$.

We summarize the computational costs in Table 1. Recall $N_x \times N_y$ is the dimension of the trial space, and $\hat{N}_x \times \hat{N}_y$ is the direction of the test space. The total computational cost of one-time step of the simulation is then $\mathscr{O}(N_x \hat{N}_x^2 + N_y \hat{N}_y^2 + N_x^2 \hat{N}_x + N_y^2 \hat{N}_y) = \mathscr{O}(N_x \hat{N}_x^2 + N_y \hat{N}_y^2)$ since $N_x < \hat{N}_x$ and $N_y < \hat{N}_y$.

Alternative methods include the multi-frontal solver [23] or an iterative solver [24]. The mutli-frontal solver executed for the system (20) has the computational cost of $\mathscr{O}((N_x N_y + \hat{N}_x \hat{N}_y)^{\frac{3}{2}})$. An iterative solver would have a computational cost of $\mathscr{O}(\psi((N_x N_y) + (\hat{N}_x \hat{N}_y))$ where ψ is the number of the iterative solver steps, which depends on the problem setup and the type of iterative solver. However, the construction of an iterative solver for high Reynolds number Navier-Stokes equations is challenging; it usually requires the application of special preconditioners, and it provides an approximate solution. Our solver does not require a preconditioner, and it is a direct solver that provides the solution with accuracy up to the numerical representation and factorization errors.

Step	Computational cost
Generation of (24) RHS	$\mathscr{O}(N_x \hat{N}_x^2 + N_y \hat{N}_y^2)$
Solution of (24)	$\mathscr{O}(N_x^2 \hat{N}_x)$
Solution of (25)	$\mathscr{O}(N_y^2 \hat{N}_y)$

Table 1. Computational costs of the Kronecker product based solver.

Solver	# non-zero entries	time [s]
velocity 1:	1770294	2.867
velocity 2:	1770294	2.936
pressure 1:	931731	4.778
pressure 2:	931731	5.014

Table 2. Problem dimensions and execution times for particular velocity and pressure solvers for the first time-step of the simulation.

4. PROBLEM WITH MANUFACTURED SOLUTION

We consider the non-stationary Stokes equation over a 2D spatial domain $\Omega = (0,1)^2$ and $I = (0,2]$ with no-slip boundary conditions: Find $\mathbf{v} = (v_1, v_2)$ and p such that

$$\begin{cases} \partial_t \mathbf{v} - \dfrac{1}{Re(t)} \Delta \mathbf{v} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{v} = 0, \\ \mathbf{v}|_{\partial\Omega} = 0, \\ \mathbf{v}(0) = \mathbf{v}_0, \end{cases} \tag{27}$$

with $\mathbf{f}$ and $\mathbf{v}_0$ defined in such a way that the manufactured solution is $\mathbf{v}(x,y,t) = (sin(x)sin(y+t), cos(x)cos(y+t))$ and $p(x,y) = cos(x)sin(y+t)$, for $(x,y) \in \Omega = [0,1]^2$ and $t \in [0,2]$, where $Re(t) = 1 + 1999 * t$. We consider time interval $t \in [0,2]$. We compute 128 time steps uniformly distributed over the interval $[0,2]$. We employ the same matrix of the optimal test functions as in the cavity flow example. The mesh size is 40×40; we employ cubic B-splines, $C2$, for the trial and quartic B-splines, $C2$, for the test. We use identical B-splines to approximate the velocity and pressure fields. The dimension of the trial space is $dimtrial = 5547$, and the dimension of the test space is $dimtest = 20667$. The problem dimensions and costs for the first and last iterations are summarized in Tables 2-3. We run the simulation on Intel(R) Core(TM) i7-9750H processor with 2.60GHz clock and with 32 GB of RAM. The snapshots from the simulation are presented in Figure 7. The sub-problem dimensions, execution times, and FLOPS for particular velocity and pressure solvers are for the simulation's first and last time-step. We can observe a slight increase in the execution time, which results from the increasing temperature of the laptop's processor, which slows down the clock so as not to overheat the processor. For this problem with manufactured solution, we can measure the relative error in L^2 norm, and it is less than 1 percent, as presented in Figure 9, except for the initial state and the final state, where it goes up to around 2.5 percent.

5. CONCLUSIONS

This paper combined the following unique features to obtain an efficient solver for transient Navier-Stokes equations. First,

Solver	# non-zero entries	time [s]
velocity 1:	1770294	3.872
velocity 2:	1770294	4.043
pressure 1:	931731	6.516
pressure 2:	931731	6.508

Table 3. Problem dimensions, and execution times for particular velocity and pressure solvers for the last time-step of the simulation.

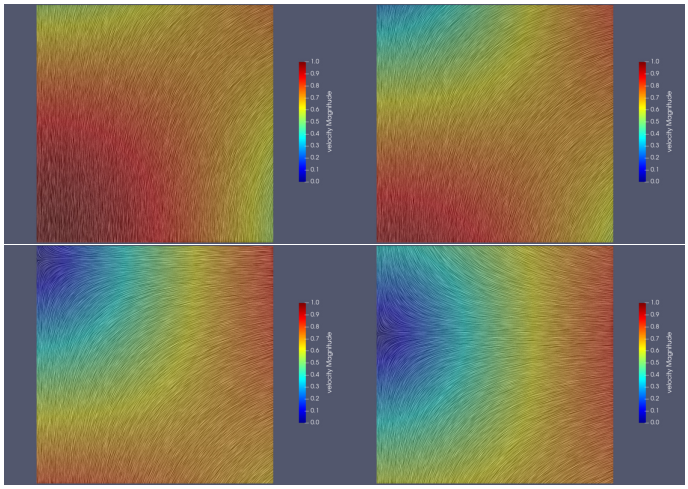


Fig. 7. Snapshots from the simulations from time steps 0, 20, 40, 64. Solution of the problem with manufactured solution employed for testing of the stabilized solver. For this problem, we know the exact solution so we can compute the numerical errors, see Figure 9.

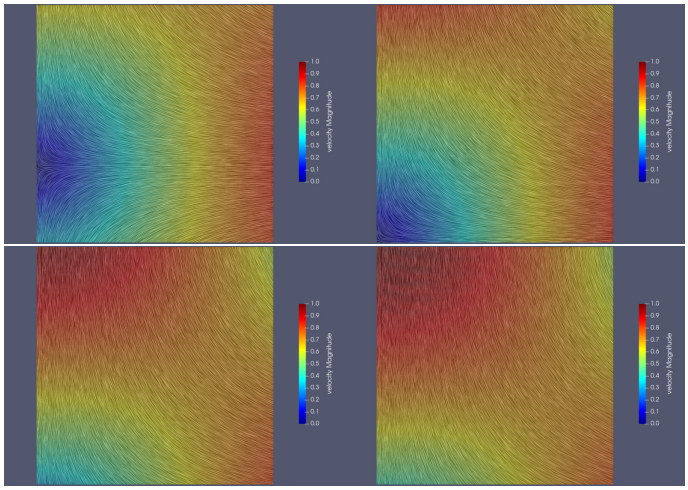


Fig. 8. Further snapshots from the numerical results of the problem with the manufactured solution were employed to test the numerical accuracy of the solver. Snapshots from the simulations from time steps 80, 100, 120, and 128. These 128 times steps span interval $[0,2]$.

we considered a direction-splitting solver for transient Navier-Stokes equations. Second, for high Reynolds numbers, the solver was stabilized using the Petrov-Galerkin formulation with the matrix of coefficients of the optimal test functions. Third, we discovered its Kronecker product structure for the matrix of coefficients of the optimal test functions (called the

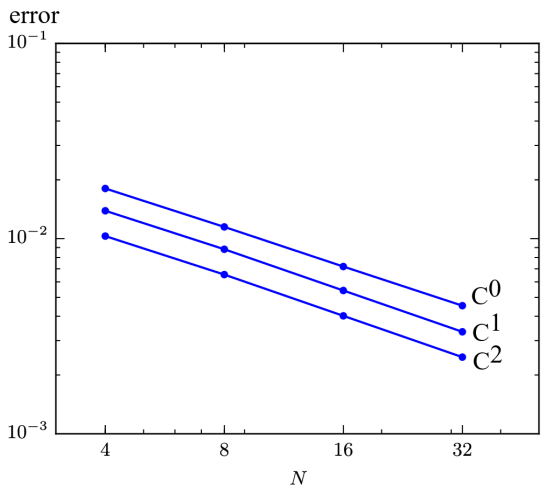


Fig. 9. The relative L^2 norm error for the entire simulation for $t \in [0,2]$, increasing mesh size from 4×4, 8×8, 16×16, up to 32×32 elements, for the linear B-splines for trial and quadratic B-splines for the test (C^0 plot), quadratic B-splines for trial and cubic B-splines for test (C^1 plot) and cubic B-splines for trial and quartic B-splines for test (C^2 plot).

stabilization matrix). We employed the Kronecker product structures of the problem matrices, resulting from the time integration scheme suitable for the variational splitting. These components allowed us to develop a fast, stabilized solver of transient Navier-Stokes equations for sequential execution on a laptop. We tested our methodology on two model problems: the cavity flow problem and the manufactured solution problem. The future work may involve extension to the refined isogeometric analysis [25] incorporating C^0 separators into mesh.

6. ACKNOWLEDGEMENET

Research project supported by the program “Excellence initiative - research university" for the AGH University of Krakow.

A. COLLECTION OF SYMBOLS

The Appendix collects all the symbols in Table 4.

REFERENCES

[1] X. Zhang, M. Chen, C. Chen, Z. Li, "Localized method of approximate particular solutions for solving unsteady Navier–Stokes problem," *Appl. Math. Model.*, vol. 40, no. 3, pp. 2265-2273, 2016.

[2] R. An, "Comparisons of Stokes/Oseen/Newton iteration methods for Navier–Stokes equations with friction boundary conditions," *Appl. Math. Model.*, vol. 38, no. 23, pp. 5535–5544, 2014.

[3] P. Huang, X. Feng, Y. He, "Two-level defect-correction oseen iterative stabilized finite element methods for the stationary Navier–Stokes equations," *Appl. Math. Model.*, vol. 37, no. 3, pp. 728–741, 2013.

[4] A. Dastbelaraki, M. Yaghoubi, M. M. Tavakol, A. Rahmatmand, "Numerical analysis of convection heat transfer from an array of perforated fins using the reynolds averaged Navier–Stokes equations and large-eddy simula-

Symbol	Definition
ξ_i	$points\_x[knot_x[i]+1]$
$B^x_{i,0}(x)$	1 if $\xi_i \le x \le \xi_{i+1}$, otherwise 0
$B^x_{i,p}(x)$	$\frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}B_{i,p-1}(x)+\frac{\xi_{i+p+1}-x}{\xi_{i+p+1}-\xi_i}B_{i+1,p-1}(x)$
η_j	$points\_y[knot_y[i]+1]$
$B^y_{j,0}(y)$	1 if $\eta_j \le y \le \eta_{j+1}$, otherwise 0
$B^y_{j,p}(y)$	$\frac{\eta-\eta_j}{\eta_{j+p}-\eta_j}B_{j,p-1}(y)+\frac{\eta_{j+p+1}-y}{\eta_{j+p+1}-\eta_j}B_{j+1,p-1}(y)$
n	$i+(j-1)N_y$
m	$k+(l-1)N_y$
$\mathbf{B}_{n,m}$	$\left(\frac{\partial B^x_{i,p}}{\partial x}B^y_{j,p},B^{xy}_{kl,p}\right)_\Omega+\varepsilon\left(\frac{\partial B^x_{i,p}}{\partial x}B^y_{j,p},\frac{\partial B^x_{k,p}}{\partial x}B^y_{l,p}\right)_\Omega$ $+\varepsilon\left(B^x_{i,p}\frac{\partial B^y_{j,p}}{\partial y},B^x_{k,p}\frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega$
f_m	$-\left((\sin(\pi y),B^{xy}_{kl,p}\right)_\Omega-\varepsilon\left(\sin(\pi y),\frac{\partial B^x_{k,p}}{\partial x}B^y_{l,p}\right)_\Omega$ $+\varepsilon\left((1-x)\pi\cos(\pi y),B^x_{k,p}\frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega$
$\mathbf{G}_{n,m}$	$\left(B^{xy}_{ij,p},B^{xy}_{kl,p}\right)_\Omega+\varepsilon\left(\frac{\partial B^x_{i,p}}{\partial x}B^y_{j,p},\frac{\partial B^x_{k,p}}{\partial x}B^y_{l,p}\right)_\Omega$ $+\varepsilon\left(B^x_{i,p}\frac{\partial B^y_{j,p}}{\partial y},B^x_{k,p}\frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega$
$\mathbf{W}$	$[w^1,\cdots,w^n]$
$M^x_{i,k}$, $K^x_{i,k}$	$\left(B^x_{i,p},B^x_{k,p}\right)_\Omega$, $\left(\frac{\partial B^x_{i,p}}{\partial x},\frac{\partial B^x_{k,p}}{\partial x}\right)_\Omega$
$R^x_{i,k}$, $M^y_{j,l}$	$\left(\frac{\partial B^x_{i,p}}{\partial x},B^x_{k,p}\right)_\Omega$, $\left(B^y_{j,p},B^y_{l,p}\right)_\Omega$
$K^y_{j,l}$, $R^y_{j,l}$	$\left(\frac{\partial B^y_{j,p}}{\partial y},\frac{\partial B^y_{l,p}}{\partial y}\right)_\Omega$, $\left(\frac{\partial B^y_{j,p}}{\partial y},B^y_{l,p}\right)_\Omega$

Table 4. Collection of symbols

tion method," *Appl. Math. Model.*, vol. 63, pp. 660–687, 2018.

[5] P. Huang, X. Feng, D. Liu, "A stabilized finite element method for the time-dependent Stokes equations based on Crank–Nicolson scheme," *Appl. Math. Model.*, vol. 37, no. (4, pp. 1910–1919, 2013.

[6] X. Feng, Y. He, D. Liu, "Convergence analysis of an implicit fractional-step method for the incompressible Navier–Stokes equations," *Appl. Math. Model.*, vol. 35, no. 12, pp. 5856–5871, 2011.

[7] M. Łoś, M. Woźniak, M. Paszyński, "Varying coefficients in parallel shared-memory variational splitting solvers for non-stationary Maxwell equations," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 72, no. 3, p. e149179, 2024.

[8] M. Łoś, M. Woźniak, I. Muga, M. Paszyński, "Three-dimensional simulations of the airborne COVID-19 pathogens using the advection-diffusion model and alternating-directions implicit solver," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 69, no. 4, p. e137125, 2021.

[9] M. Łoś, P. Behnoudfar, M. Dobija, M. Paszyński "ADI-based, conditionally stable schemes for seismic P-wave and elastic wave propagation problems," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 70, no. 5, p. e141985, 2022.

[10] L. Demkowicz, J. Gopalakrishnan, "A class of discontinuous Petrov–Galerkin methods. Part I: The transport equation," *Comput. Methods. Appl. Mech. Eng.*, vol. 199, no. 23, pp. 1558–1572, 2010.

[11] N. V. Roberts, T. Bui-Thanh, L. Demkowicz, "The DPG method for the Stokes problem," *Comput. Math. Appl.*, vol. 67, no. 4, pp. 966–995, 2014.

[12] A. H. Niemi, J. A. Bramwell, L. F. Demkowicz, "Discontinuous Petrov–Galerkin method with optimal test functions for thin-body problems in solid mechanics," *Comput. Methods. Appl. Mech. Eng.*, vol. 200, no. 9, pp. 1291-1300, 2011.

[13] V. M. Calo, N. O. Collier, A. H. Niemi, "Analysis of the discontinuous petrov–galerkin method with optimal test functions for the Reissner–Mindlin plate bending model," *Comput. Math. Appl.*, vol. 66, no. 12, pp. 2570–2586, 2014.

[14] J.-L. Guermond, P. Minev, "A new class of massively parallel direction splitting for the incompressible Navier–Stokes equations," *Comput. Methods. Appl. Mech. Eng.*, vol. 200, np. 23-24, pp. 2083–2093, 2011.

[15] M. Łoś, I. Muga, J. Munoz-Matute, M. Paszyński, "Isogeometric residual minimization (iGRM) for non-stationary Stokes and Navier-Stokes problems," *Comput. Math. Appl.*, vol. 95, pp. 200–214, 2021.

[16] M. Łoś, J. Munoz-Matute, I. Muga, M. Paszyński, "Isogeometric residual minimization method (iGRM) with direction splitting for non-stationary advection–diffusion problems," *Comput. Math. Appl.*, vol. 79, no. 2, pp. 213–229, 2020.

[17] M. Woźniak, M. Łoś, M. Paszyński, L. Dalcin, V. Calo, "Parallel fast isogeometric solvers for explicit dynamics," *Comput. Inform.*, vol. 36, no. 2, pp. 423-448, 2017.

[18] J. Liesen, V. Mehrmann, *The Kronecker Product and Linear Matrix Equations*, Springer International Publishing, Cham, pp. 303-310, 2015.

[19] T. Lyche, *The Kronecker Product*, Springer International Publishing, Cham, pp. 225–236, 2020.

[20] Y. S. Yun, C. J. Kang, "Some results on kronecker products and commutation matrices," *East Asian J Applied Math.*, vol. 29, pp. 259–268, 2013.

[21] K. Eriksson, C. Johnson, "Adaptive finite element methods for parabolic problems I: A linear model problem," *SIAM J. Num. Anal.*, vol. 28, no. 1, pp. 43-77, 1991.

[22] T. Hughes, J. Cottrell, Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Comput. Methods. Appl. Mech. Eng.*, vol. 194, no. 39, pp. 4135–4195, 2005.

[23] M. Woźniak, M. Paszyński, D. Pardo, L. Dalcin, V. Calo, "Computational cost of isogeometric multi-frontal solvers on parallel distributed memory machines," *Comput. Methods. Appl. Mech. Eng.*, vol. 284, pp. 971-987, 2015.

[24] D. Pardo, J. Alvarez-Aramberri, M. Paszyński, L. Dalcin, V. Calo, "Impact of element-level static condensation on iterative solver performance," *Comput. Math. Appl.*, vol. 70, no. 10, pp. 2331-2341, 2015.

[25] A. Paszyńska, K. Jopek, M. Woźniak, M. Paszyński, "Heuristic algorithm to predict the location of C0 separators for efficient isogeometric analysis simulations with direct solvers," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 66, no. 6, pp. 907-917, 2022.