

Parallel Shared-Memory Isogeometric Residual Minimization (iGRM) for Three-Dimensional Advection-Diffusion Problems

Marcin Łoś¹, Judit Munoz-Matute²,
Krzysztof Podsiadło¹, Maciej Paszyński¹, and Keshav Pingali³

¹ AGH University of Science and Technology, Kraków, Poland

² The University of the Basque Country, Bilbao, Spain

³ The University of Texas at Austin, USA

maciej.paszynski@agh.edu.pl

Abstract. In this paper, we present a residual minimization method for three-dimensional isogeometric analysis simulations of advection-diffusion equations. First, we apply the implicit time integration scheme for the three-dimensional advection-diffusion equation. Namely, we utilize the Douglas-Gunn time integration scheme. Second, in every time step, we apply the residual minimization method for stabilization of the numerical solution. Third, we use isogeometric analysis with B-spline basis functions for the numerical discretization. We perform alternating directions splitting of the resulting system of linear equations, so the computational cost of the sequential LU factorization is linear $\mathcal{O}(N)$. We test our method on the three-dimensional simulation of the advection-diffusion problem. We parallelize the solver for shared-memory machine using the GALOIS framework.

Keywords: isogeometric analysis · implicit dynamics · advection-diffusion problems · linear computational cost · direct solvers · GALOIS framework

1 Introduction

The alternating direction implicit method (ADI) is a popular method for performing finite difference simulations on regular grids. The first papers concerning the ADI method were published in 1960 [1, 3, 5, 19]. This method is still popular for fast solutions of different classes of problems with finite difference method [8, 9]. In its basic version, the method introduces intermediate time steps, and the differential operator splits into the x , y (and z in 3D) components. As a result of this operation, on the left-hand side, we only deal with derivatives in one direction, while the rest of the operator is on the right-hand side. The resulting system of linear equations has a multi-diagonal form, so the factorization of this system is possible with a linear $\mathcal{O}(N)$ computational cost. It is a common misunderstanding that the direction splitting solvers are limited to simple geometries. They can be also applied to discretizations in extremely complicated geometries, as described in [10].

In this paper, we generalize this method for three-dimensional simulations of the time-dependent advection-diffusion problem with the residual minimization method. We use the basic version of the direction splitting algorithm, working on a regular computational cube, since this approach is straightforward and it is enough to prove our claims that the residual minimization stabilizes the advection-diffusion simulations. In particular, we apply the residual minimization method with isogeometric finite element method simulations over a three-dimensional cube shape computational

grids with tensor product B-spline basis functions. The resulting system of linear equations can be factorized in a linear $\mathcal{O}(N)$ computational cost when executed in sequential mode.

We use the finite element method discretizations with B-spline basis functions. This setup, as opposed to the traditional finite difference discretization, allows us to apply the residual minimization method to stabilize our simulations.

The isogeometric analysis (IGA) [4] is a modern method for performing finite element method (FEM) simulations with B-splines and NURBS. It enables higher order and continuity B-spline based approximations of the modeled phenomena. The direction splitting method has been rediscovered to solve the isogeometric L^2 projection problem over regular grids with tensor product B-spline basis functions [6, 7]. The direction splitting, in this case, is performed with respect to space, and the splitting is possible by exploiting the Kronecker product structure of the Gram matrix with tensor product structure of the B-spline basis functions. The L^2 projections with IGA-FEM were applied for performing fast and smooth simulations of explicit dynamics [11–14, 16, 20]. This is because the explicit dynamics with isogeometric discretization is equivalent to the solution of a sequence of isogeometric L^2 projections.

In this paper, we focus on the advection-diffusion equation used for simulation of the propagation of a pollutant from a chimney. We introduce implicit time integration scheme, that allows for the alternating direction splitting of the advection-diffusion equation. We discover that the numerical simulations are unstable, and deliver some unexpected oscillations and reflections. Next, we utilize the residual minimization method in a way that it preserves the Kronecker product structure of the matrix and enables stabilized linear computational cost solutions.

The actual mathematical theory concerning the stability of the numerical method for weak formulations is based on the famous “Babuška-Brezzi condition” (BBC) developed in years 1971–1974 at the same time by Ivo Babuška, and Franco Brezzi [25–27]. The condition states that a weak problem is stable when

$$\sup_{v \in V} \frac{|b(u, v)|}{\|v\|_V} \geq \gamma \|u\|_U, \forall u \in U. \quad (1)$$

However, the inf-sup condition in the above form concerns the abstract formulation where we consider all the test functions from $v \in V$ and look for solution at $u \in U$ (e.g. $U = V$). The above condition is satisfied also if we restrict to the space of trial functions $u_h \in U_h \subset U$

$$\sup_{v \in V} \frac{|b(u_h, v)|}{\|v\|_V} \geq \gamma \|u_h\|_{U_h}. \quad (2)$$

However, if we use test functions from the finite dimensional test space $V_h = \text{span}\{v_h\} \subset V$

$$\sup_{v_h \in V_h} \frac{|b(u_h, v_h)|}{\|v_h\|_{V_h}} \geq \gamma_h \|u_h\|_{U_h}, \quad (3)$$

we do not have a guarantee that the supremum (3) will be equal to the original supremum (1), since we have restricted V to V_h . The optimality of the method depends on the quality of the polynomial test functions defining the space $V_h = \text{span}\{v_h\}$ and how far are they from the supremum defined in (1). There are many methods for stabilization of different PDEs [28–31]. In 2010, the Discontinuous Petrov Galerkin (DPG) method was proposed, with the modern summary of the method described in [32].

The DPG method utilizes the residual minimization with broken test spaces. In other words, it first generates a system of linear equations

$$\begin{bmatrix} G & -B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix} = \begin{bmatrix} l \\ 0 \end{bmatrix}. \quad (4)$$

This system of linear equations has the inner product block G over the test space, the two blocks with the actual weak form B and B^T , and the zero block 0. The test space is larger than the trial space, and the inner product and the weak form blocks are rather sparse matrices. Therefore, the dimension of the system of linear equations is at least two times larger than the original system of equations arising from standard Galerkin method. In the DPG method, the test space is broken in order to obtain a block-diagonal matrix G and the Schur complements can be locally computed over each finite element. The price to pay is the presence of the additional fluxes on the element interfaces, resulting from breaking the test spaces, so the system over each finite element looks like

$$\begin{bmatrix} G & -B_1 & -B_2 \\ B_1^T & 0 & 0 \\ B_2^T & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ u \\ t \end{bmatrix} = \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}. \quad (5)$$

We do not know any other reason of breaking the test spaces in the DPG method other than reduction of the computational cost of the solver.

In this paper, we want to avoid dealing with fluxes and broken spaces since it is technically very complicated. Thus, we stay with the unbroken global system (4) and then we have to face one of the two possible methods. The first one would be to apply adaptive finite element method, but then the cost of factorization in 3D would be up to four times slower than in the standard finite element method and broken DPG (without the static condensation). This is because depending on the structure of the refined mesh, we will have a computational cost of the multi-frontal solver varying between $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ [33], and our N is two times bigger than in the original weak problem, and $2^2 = 4$. This could be an option that we will discuss in a future paper.

Another method that we exploit in this paper is to keep a tensor product structure of the computational patch of elements with tensor product B-spline basis functions, decompose the system matrix into a Kronecker product structure, and utilize a linear computational cost alternating directions solver. Even for the system (4) resulting from the residual minimization we successfully perform direction splitting to obtain a Kronecker product structure of the matrix to maintain the linear computational cost of the alternating directions method.

In order to stabilize the time-dependent advection-diffusion simulations, we perform the following steps. First, we apply the time integration scheme. We use the Douglas-Gunn second order time integration scheme [2]. Second, we stabilize a system from every time step by employing the residual minimization method [34–36]. Finally, we perform numerical discretization with isogeometric analysis [4], using tensor product B-spline basis functions over a three-dimensional cube shape patch of elements.

The novelties of this paper with regard to our previous work are the following. In [11], we described parallel object-oriented JAVA based implementation of the explicit dynamics version of the alternating directions solver, without any residual minimization stabilization, and for two-dimensional problems only. In [12], we described sequential Fortran based implementation of the explicit dynamics solver, with applications of the elastic wave propagation, without implicit time integration schemes and any residual minimization stabilization. In [16], we described the parallel distributed memory implementation of the explicit dynamics solver, again without implicit time

integration scheme and residual minimization method. In [14], we described the parallel shared-memory implementation of the explicit dynamics solver, with the same restrictions as before. In [13, 17] we applied the explicit dynamics solver for two and three-dimensional tumor growth simulations. In all of these papers, we did not use implicit time integration schemes, and we did not perform operator splitting on top of the residual minimization method. In [20], we investigate different time integration schemes for two-dimensional residual minimization method for advection-diffusion problems. We do not go for three-dimensional computations, and we do not apply parallel computations there.

In this paper, we apply the residual minimization with direction splitting for the first time in three-dimensions. We also investigate the parallel scalability of our solver, using the GALOIS framework for parallelization. For more details on the GALOIS framework itself, we refer to [37–40].

The structure of this paper is the following. We start in Section 2 with the derivation of the isogeometric alternating direction implicit method for the advection-diffusion problem. The following Section 3 derives the residual minimization method formulation of the advection-diffusion problem in three-dimensions. Next, in Section 4, we present the linear computational cost numerical results. We summarize the paper with conclusions in Section 5

2 Model problem of three-dimensional advection-diffusion

Let $\Omega = \Omega_x \times \Omega_y \times \Omega_z \subset \mathbb{R}^3$ an open bounded domain and $I = (0, T] \subset \mathbb{R}$, we consider the three-dimensional *linear advection-diffusion equation*

$$\begin{cases} u_t - \nabla \cdot (\alpha \nabla u) + \beta \cdot \nabla u = f & \text{in } \Omega \times I, \\ u = 0 & \text{on } \Gamma \times I, \\ u(0) = u_0 & \text{in } \Omega, \end{cases} \quad (6)$$

where Ω_x, Ω_y and Ω_z are intervals in \mathbb{R} . Here, $u_t := \partial u / \partial t$, $\Gamma = \partial \Omega$ denotes the boundary of the spatial domain Ω , $f : \Omega \times I \rightarrow \mathbb{R}$ is a given source and $u_0 : \Omega \rightarrow \mathbb{R}$ is a given initial condition. We consider constant diffusivity α and constant velocity field $\beta = [\beta_x \ \beta_y \ \beta_z]$.

We split the advection-diffusion operator $\mathcal{L}u = -\nabla \cdot (\alpha \nabla u) + \beta \cdot \nabla u$ as $\mathcal{L}u = \mathcal{L}_1 u + \mathcal{L}_2 u + \mathcal{L}_3 u$ where

$$\mathcal{L}_1 u := -\alpha \frac{\partial u}{\partial x^2} + \beta_x \frac{\partial u}{\partial x}, \quad \mathcal{L}_2 u := -\alpha \frac{\partial u}{\partial y^2} + \beta_y \frac{\partial u}{\partial y}, \quad \mathcal{L}_3 u := -\alpha \frac{\partial u}{\partial z^2} + \beta_z \frac{\partial u}{\partial z}.$$

Based on this operator splitting, we consider different Alternating Direction Implicit (ADI) schemes to discretize problem (6).

First, we perform a uniform partition of the time interval $\bar{I} = [0, T]$ as

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

and denote $\tau := t_{n+1} - t_n$, $\forall n = 0, \dots, N-1$.

In the Douglas-Gunn scheme, we integrate the solution from time step t_n to t_{n+1} in three substeps as follows:

$$\begin{cases} (1 + \frac{\tau}{2} \mathcal{L}_1) u^{n+1/3} = \tau f^{n+1/2} + (1 - \frac{\tau}{2} \mathcal{L}_1 - \tau \mathcal{L}_2 - \tau \mathcal{L}_3) u^n, \\ (1 + \frac{\tau}{2} \mathcal{L}_2) u^{n+2/3} = u^{n+1/3} + \frac{\tau}{2} \mathcal{L}_2 u^n, \\ (1 + \frac{\tau}{2} \mathcal{L}_3) u^{n+1} = u^{n+2/3} + \frac{\tau}{2} \mathcal{L}_3 u^n. \end{cases} \quad (7)$$

The variational formulation of scheme (7) is

$$\left\{ \begin{array}{l} (u^{n+1/3}, v) + \frac{\tau}{2} \left(\alpha \frac{\partial u^{n+1/3}}{\partial x}, \frac{\partial v}{\partial x} \right) + \frac{\tau}{2} \left(\beta_x \frac{\partial u^{n+1/3}}{\partial x}, v \right) = (u^n, v) - \frac{\tau}{2} \left(\alpha \frac{\partial u^n}{\partial x}, \frac{\partial v}{\partial x} \right) - \frac{\tau}{2} \left(\beta_x \frac{\partial u^n}{\partial x}, v \right) \\ \quad - \tau \left(\alpha \frac{\partial u^n}{\partial y}, \frac{\partial v}{\partial y} \right) - \tau \left(\beta_y \frac{\partial u^n}{\partial y}, v \right) - \tau \left(\alpha \frac{\partial u^n}{\partial z}, \frac{\partial v}{\partial z} \right) - \tau \left(\beta_z \frac{\partial u^n}{\partial z}, v \right) + \tau (f^{n+1/2}, v), \\ (u^{n+2/3}, v) + \frac{\tau}{2} \left(\alpha \frac{\partial u^{n+2/3}}{\partial y}, \frac{\partial v}{\partial y} \right) + \frac{\tau}{2} \left(\beta_y \frac{\partial u^{n+2/3}}{\partial y}, v \right) = (u^{n+1/3}, v) + \frac{\tau}{2} \left(\alpha \frac{\partial u^n}{\partial y}, \frac{\partial v}{\partial y} \right) + \frac{\tau}{2} \left(\beta_y \frac{\partial u^n}{\partial y}, v \right), \\ (u^{n+1}, v) + \frac{\tau}{2} \left(\alpha \frac{\partial u^{n+1}}{\partial z}, \frac{\partial v}{\partial z} \right) + \frac{\tau}{2} \left(\beta_z \frac{\partial u^{n+1}}{\partial z}, v \right) = (u^{n+2/3}, v) + \frac{\tau}{2} \left(\alpha \frac{\partial u^n}{\partial z}, \frac{\partial v}{\partial z} \right) + \frac{\tau}{2} \left(\beta_z \frac{\partial u^n}{\partial z}, v \right), \end{array} \right. \quad (8)$$

where (\cdot, \cdot) denotes the inner product of $L^2(\Omega)$. Finally, expressing problem (8) in matrix form we have

$$\left\{ \begin{array}{l} \left[M^x + \frac{\tau}{2}(K^x + G^x) \right] \otimes M^y \otimes M^z u^{n+1/3} \\ = \left[M^x - \frac{\tau}{2}(K^x + G^x) \right] \otimes M^y \otimes M^z u^n \\ \quad - \tau M^x \otimes (K^y + G^y) \otimes M^z u^n - \tau M^x \otimes M^y \otimes (K^z + G^z) u^n + \tau F^{n+1/2} \\ M^x \otimes \left[M^y + \frac{\tau}{2}(K^y + G^y) \right] \otimes M^z u^{n+2/3} \\ = M^x \otimes M^y \otimes M^z u^{n+1/3} + M^x \otimes \frac{\tau}{2}(K^y + G^y) \otimes M^z u^n, \\ M^x \otimes M^y \otimes \left[M^z + \frac{\tau}{2}(K^z + G^z) \right] u^{n+1} \\ = M^x \otimes M^y \otimes M^z u^{n+2/3} + M^x \otimes M^y \otimes \frac{\tau}{2}(K^z + G^z) u^n, \end{array} \right. \quad (9)$$

where $M^{x,y,z}$, $K^{x,y,z}$ and $G^{x,y,z}$ are the 1D mass, stiffness and advection matrices, respectively.

3 Isogeometric residual minimization method

In our method, in every time step we solve the problem with identical left-hand-side: Find $u \in U$ such that

$$b(u, v) = l(v) \quad \forall v \in V, \quad (10)$$

$$b(u, v) = (u, v) + \tau/2 \left(\left(\beta_{\underline{i}} \frac{\partial u}{\partial x_{\underline{i}}}, v \right) + \alpha_{\underline{i}} \left(\frac{\partial u}{\partial x_{\underline{i}}}, \frac{\partial v}{\partial x_{\underline{i}}} \right) \right), \quad (11)$$

Here $i \in \{1, 2, 3\}$, so we have denoted here $(x_1, x_2, x_3) = (x, y, z)$, and \underline{i} means that we are not using the Einstein summation here. The right-hand-side depends on the sub-step and the time integration scheme used. In the Douglas-Gunn time integration scheme, in the first, second and third sub-step

the right-hand side is defined as:

$$\left\{ \begin{array}{l} l(w, v) = (w, v) - \frac{\tau}{2} \left(\alpha \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} \right) - \frac{\tau}{2} \left(\beta_x \frac{\partial w}{\partial x}, v \right) - \tau \left(\alpha \frac{\partial w}{\partial y}, \frac{\partial v}{\partial y} \right) - \tau \left(\beta_y \frac{\partial w}{\partial y}, v \right) \\ \quad - \tau \left(\alpha \frac{\partial w}{\partial z}, \frac{\partial v}{\partial z} \right) - \tau \left(\beta_z \frac{\partial w}{\partial z}, v \right) + \tau(f^{n+1/2}, v), \\ l(w, v) = (w, v) + \frac{\tau}{2} \left(\alpha \frac{\partial w}{\partial y}, \frac{\partial v}{\partial y} \right) + \frac{\tau}{2} \left(\beta_y \frac{\partial w}{\partial y}, v \right), \\ l(w, v) = (w, v) + \frac{\tau}{2} \left(\alpha \frac{\partial w}{\partial z}, \frac{\partial v}{\partial z} \right) + \frac{\tau}{2} \left(\beta_z \frac{\partial w}{\partial z}, v \right). \end{array} \right. \quad (12)$$

In our advection-diffusion problem we seek the solution in space

$$U = V = \left\{ v : \int_{\Omega} \left(v^2 + \left(\frac{\partial v}{\partial x_i} \right)^2 \right) < \infty \right\}. \quad (13)$$

where $i = 1, 2, 3$ denotes the spatial directions. The inner product in V is defined as

$$(u, v)_V = (u, v)_{L_2} + \left(\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i} \right)_{L_2}, \quad (14)$$

where $i = 1, 2, 3$ depending on the sub-step index in the alternating directions method, and we do not use here the Einstein convention. For a weak problem, we define the operator

$$B : U \rightarrow V', \quad (15)$$

such that

$$\langle Bu, v \rangle_{V' \times V} = b(u, v), \quad (16)$$

so we can reformulate the problem as

$$Bu - l = 0. \quad (17)$$

We wish to minimize the residual

$$u_h = \operatorname{argmin}_{w_h \in U_h} \frac{1}{2} \|Bw_h - l\|_{V'}^2. \quad (18)$$

We introduce the Riesz operator being the isometric isomorphism

$$R_V : V \ni v \rightarrow (v, \cdot) \in V'. \quad (19)$$

We can project the problem back to V

$$u_h = \operatorname{argmin}_{w_h \in U_h} \frac{1}{2} \|R_V^{-1}(Bw_h - l)\|_V^2. \quad (20)$$

The minimum is attained at u_h when the Gâteaux derivative is equal to 0 in all directions:

$$\langle R_V^{-1}(Bu_h - l), R_V^{-1}(Bw_h) \rangle_V = 0, \quad \forall w_h \in U_h. \quad (21)$$

We define the error representation function $r = R_V^{-1}(Bu_h - l)$ and our problem is reduced to

$$\langle r, R_V^{-1}(B w_h) \rangle = 0, \quad \forall w_h \in U_h, \quad (22)$$

which is equivalent to

$$\langle B w_h, r \rangle = 0, \quad \forall w_h \in U_h. \quad (23)$$

From the definition of the residual we have

$$(r, v)_V = \langle Bu_h - l, v \rangle, \quad \forall v \in V. \quad (24)$$

Our problem reduces to the following semi-infinite problem: Find $(r, u_h)_{V \times U_h}$ such as

$$\begin{aligned} (r, v)_V - \langle Bu_h, v \rangle &= \langle l, v \rangle, \quad \forall v \in V, \\ \langle B w_h, r \rangle &= 0 \quad \forall w_h \in U_h. \end{aligned} \quad (25)$$

We discretize the test space $V_m \in V$ to get the discrete problem: Find $(r_m, u_h)_{V_m \times U_h}$ such as

$$\begin{aligned} (r_m, v_m)_{V_m} - \langle Bu_h, v_m \rangle &= \langle l, v \rangle \quad \forall v \in V_m \\ \langle B w_h, r_m \rangle &= 0 \quad \forall w_h \in U_h. \end{aligned} \quad (26)$$

Note that the residual minimization method is a Petrov-Galerkin method (test and trial spaces are different). We stabilize the problem by increasing the dimension of the test space. Notice that the residual minimization system here is of the following form

$$\begin{bmatrix} G & -B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix} = \begin{bmatrix} l \\ 0 \end{bmatrix}, \quad (27)$$

where the right-top and left-bottom matrices B and B^T can be split according to (9), and the inner product (14) part G can be split in the following way:

$$G = \begin{cases} [\tilde{M}^x + \tilde{K}^x] \otimes \tilde{M}^y \otimes \tilde{M}^z, \\ \tilde{M}^x \otimes [\tilde{M}^y + \tilde{K}^y] \otimes \tilde{M}^z, \\ \tilde{M}^x \otimes \tilde{M}^y \otimes [\tilde{M}^z + \tilde{K}^z], \end{cases} \quad (28)$$

where we consider three different splittings for three sub-steps, and $\tilde{M}^{x,y,z}$, and $\tilde{K}^{x,y,z}$ are the 1D mass and stiffness matrices over the test space in direction x, y , or z , respectively.

Now, in the first sub-step, we approximate the solution with tensor product of one dimensional B-splines basis functions of order p , $u_h = \sum_{i,j,k} u_{i,j,k} B_{i;p}^x(x) B_{j;p}^y(y) B_{k;p}^z(z)$. We test with tensor product of one dimensional B-splines basis functions, where we enrich the order in the direction of the x axis from p to $o \geq p$, and we enrich the test space only in the direction of the alternating splitting $v_m \leftarrow B_{i;o}^x(x) B_{j;p}^y(y) B_{k;p}^z(z)$. We approximate the residual with tensor product of one dimensional B-splines basis functions of order p , $r_m = \sum_{s,t,q} r_{s,t,q} B_{s;t}^x(x) B_{t;p}^y(y) B_{q;p}^z(z)$, and we test with tensor product of 1D B-spline basis functions of order o and p , in the corresponding directions $w_h \leftarrow B_{k;o}^x(x) B_{l;p}^y(y) B_{m;p}^z(z)$.

Notice that we stabilize the problem by enriching the test space with respect to the trial space in the alternating direction manner. Now, in the first sub-step we have $M^y = \tilde{M}^y$, $M^z = \tilde{M}^z$ and $M^x \neq \tilde{M}^x$, and $M^{yT} = M^y$, $M^{zT} = M^z$. Now, in the first sub-step we have

$$\begin{pmatrix} G & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} [\tilde{M}^x + \tilde{K}^x] & [M^x + \frac{\tau}{2}(K^x + G^x)] \\ [M^x + \frac{\tau}{2}(K^x + G^x)]^T & 0 \end{pmatrix} \otimes M^y \otimes M^z \quad (29)$$

in the second sub-step

$$\begin{pmatrix} G & B \\ B^T & 0 \end{pmatrix} = M^x \otimes \begin{pmatrix} [\tilde{M}^y + \tilde{K}^y] & [M^y + \frac{\tau}{2}(K^y + G^y)] \\ [M^y + \frac{\tau}{2}(K^y + G^y)]^T & 0 \end{pmatrix} \otimes M^z \quad (30)$$

and in the third sub-step

$$\begin{pmatrix} G & B \\ B^T & 0 \end{pmatrix} = M^x \otimes M^y \otimes \begin{pmatrix} [\tilde{M}^z + \tilde{K}^z] & [M^z + \frac{\tau}{2}(K^z + G^z)] \\ [M^z + \frac{\tau}{2}(K^z + G^z)]^T & 0 \end{pmatrix} \quad (31)$$

All these matrices are the Kronecker products of three multi-diagonal sub-matrices, and they can be factorized in a linear $\mathcal{O}(N)$ computational cost.

4 Numerical results

4.1 Manufactured solution problem

In order to verify the order and accuracy of the Douglas-Gunn time-integration schemes with IGA-FEM discretization and the direction splitting solver, we construct a time-dependent advection-diffusion problem with manufactured solution.

$$\frac{du}{dt} - \nabla \cdot (\alpha \nabla u) + \beta \cdot \nabla u = f,$$

with $\alpha = 10^{-2}$, $\beta = (1, 0, 0)$, with zero Dirichlet boundary conditions solved on a square $[0, 1]^3$ domain. We setup the forcing function $f(x, y, z; t)$ in such a way that it delivers the manufactured solution of the form $u_{exact}(x, y, z; t) = \sin(\pi x) \sin(\pi y) \sin(\pi z) \sin(\pi t)$ on a time interval $[0, 2]$.

We solve the problem with residual minimization method on $32 \times 32 \times 32$ mesh with different time steps, as presented in Figure 1, using the Douglas-Gunn time integration scheme and the direction splitting solver using the Kronecker product structure of the matrices.

We compute the error between the exact solution u_{exact} and the numerical solution u_h . We present the comparisons with different time step size τ . We compute relative error $\|u_{exact}(t) - u_h(t)\|_{L^2} / \|u_{exact}(t)\|_{L^2} \cdot 100\%$ and plot it in Figure 1. The horizontal lines represent the time step size selected for the entire simulation, and the vertical lines present the numerical error with respect to the known exact solution.

The Douglas-Gunn scheme is of the second order accurate, down to the accuracy of 10^{-5} .

4.2 Pollution propagation simulations

In this section, we describe the numerical simulation of three-dimensional model advection-diffusion problem over a 3D cube shape domain with dimensions $5000 \times 5000 \times 5000$ meters.

$$\frac{du}{dt} - \nabla \cdot (\alpha \nabla u) - \beta \cdot \nabla u = f, \quad (32)$$

In our equation we have the diffusion coefficients $\alpha = (50, 50, 0.5)$. We utilize tensor products of 1D B-splines along the x , y , and z . We apply the alternating direction implicit solver with three intermediate time steps. The velocity field is $\beta = (\beta^x(t), \beta^y(t), \beta^z(t)) = (\cos a(t), \sin a(t), v(t))$

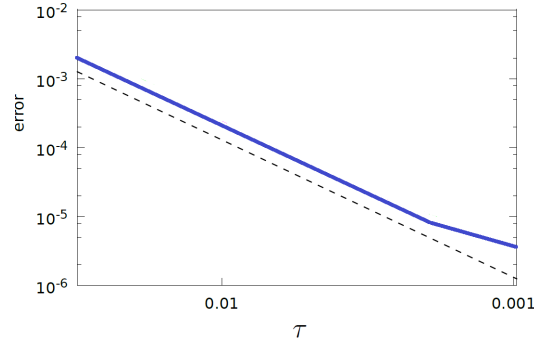


Fig. 1. Numerical error for Douglas-Gunn time integration scheme on $32 \times 32 \times 32$ mesh for different time steps.

where $a(t) = \frac{\pi}{3}(\sin(s) + \frac{1}{2}\sin(2.3s)) + \frac{3}{8}\pi$, $v(t) = \frac{1}{3}\sin(s)$, $s = \frac{t}{150}$. The source is given by $f(p) = (r-1)^2(r+1)^{-2}$, where $r = \min(1, (|p-p_0|/25)^2)$, p represents the distance from the source, and p_0 is the location of the source $p_0 = (3, 3, 2)$. The initial state is defined as the constant concentration of the order of 10^{-6} in the entire domain (numerical zero).

The physical meaning of this setup is the following. We model the propagation of the pollutant generated by a single source modeled by the f function, distributed by the wind blowing with changing directions, modeled by β function, and the diffusion phenomena modeled by the coefficients α . The computational domain unit is meter $[m]$, the wind velocity β is given in meters per second $[\frac{m}{s}]$, and the diffusion coefficient α is given in square meters per second $[\frac{m^2}{s}]$. The units for the solution are then kilograms per cube meter $[\frac{kg}{m^3}]$. We expect from the numerical results to observe the propagation of the pollutant as distributed by the wind and the diffusion process.

Our first numerical results concern the computational mesh with a size of $50 \times 50 \times 50$ elements with quadratic B-splines. We employ standard Galerkin formulation here with direction splitting, without the residual minimization method. We perform 300 time steps of the numerical simulation. The snapshots presented in Figure 2 represent time steps 100, 200 and 300. We observe unexpected “oscillations” and “reflections”. Since the simulation is supposed to model the propagation of the pollutant from a chimney by means of the advection (wind) and diffusion phenomena, the oscillations and reflections on the boundary are not expected there. Both these phenomena appear and disappear during the entire simulation; they do not cause a blowup of the entire simulations, just unexpected local behavior.

To improve the spatial stability of the simulation, we add now the residual minimization method on top of the Galerkin setup. Thus, the second simulation was performed again on the mesh size with $50 \times 50 \times 50$ elements, with quadratic B-splines for trial and cubic B-Splines for test. The snapshots from the numerical results are presented in Figure 2. We perform 300 time steps, and we present the snapshots in time steps 100, 200, and 300.

We use the implicit extension of the parallel code [14] for shared memory Linux cluster nodes. The total simulation time was 100 minutes on a laptop with i7 6700Q processor 2.6GHz (8 cores with HT) and 16B or RAM. We emphasize that ADI is not an iterative solver. It is just a linear $\mathcal{O}(N)$ computational cost solver that performs Gaussian elimination for matrices having Kronecker product structure. Thus, the solution obtained by the solver is exact (up to the round-off errors). In

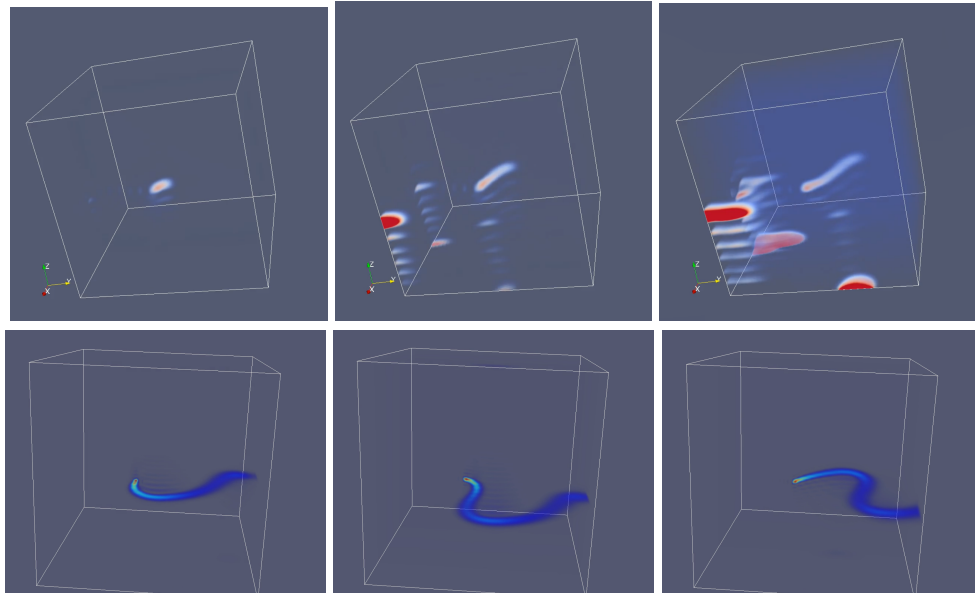


Fig. 2. Top panel: Simulation with Galerkin method. Snapshots from the time steps 10, 20, and 30 of with quadratic C1 B-splines over $50 \times 50 \times 50$ mesh, without stabilization. **Bottom panel:** Simulation with residual minimization method. Snapshots from the time steps 100, 200 and 300 of the first problem simulation with quadratic C1 B-splines for trial and cubic C2 B-splines for test over $50 \times 50 \times 50$ mesh.

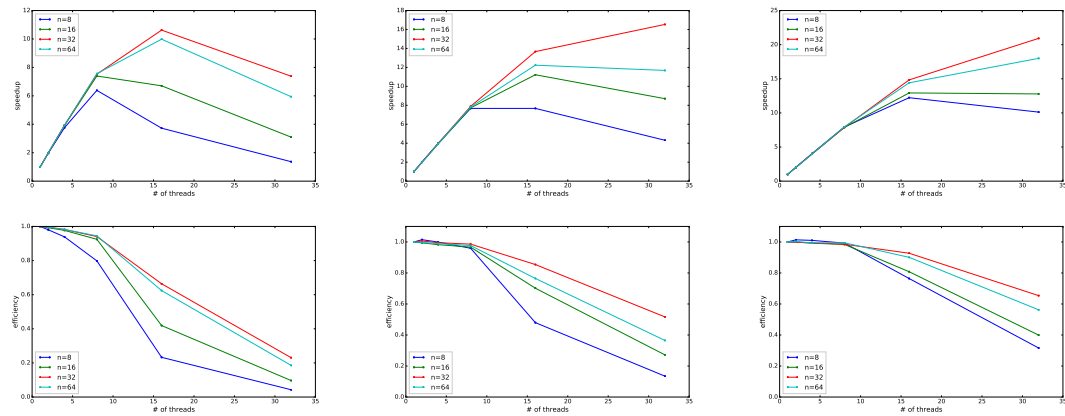


Fig. 3. Speedup (top line) and efficiency (bottom line) for $p = 1$ for trial, $p = 2$ for testing (first column), $p = 2$ for trial, $p = 3$ for testing (second column), and $p = 3$ for trial, $p = 4$ for testing (third column).

this sense, we do not present the iterations or convergence of the ADI solver, since it is executed once per each time step. In other words, we can perform 300 Gaussian elimination, each with 1,000,000 unknowns, with the high accuracy resulting from the ADI direct solver (only round-off errors are involved), on a laptop with eight cores, with the implicit method, within 1.5 hours.

5 Parallel scalability

Implementation of the ideas described in the preceding sections has been created in C++ and parallelized using our code for IGA-FEM simulations with ADI solver [14], extended to the implicit method. We use the GALOIS framework for parallelization [37–40].

The parallelization concerns mainly the algorithm of the integration of the right-hand-side vector. The cost of generation of the one-dimensional matrices and the factorization with multiple-right-hand sides is negligible in comparison to the integration of the higher-order B-splines over the three-dimensional mesh. The parallel implementation in GALOIS involves the usage of *Galois::for_each*, *Galois::Runtime::LL::SimpleLock*.

```

1. for each element  $E = [\xi_{l_x}, \xi_{l_x+1}] \times [\xi_{l_y}, \xi_{l_y+1}] \times [\xi_{l_z}, \xi_{l_z+1}]$  in parallel do
2.    $U^{loc} \leftarrow 0$ 
3.   for each quadrature point  $\xi = (X_{k_x}, X_{k_y}, X_{k_z})$  do
4.      $\mathbf{x} \leftarrow \Psi_E(\xi)$ ,  $W \leftarrow w_{k_x} w_{k_y} w_{k_z}$ ,  $u, Du \leftarrow 0$ 
5.     for  $I \in \mathcal{I}(E)$  do
6.        $u \leftarrow u + U_I^{(t)} \mathcal{B}_I(\xi)$ ,  $Du \leftarrow Du + U_I^{(t)} \nabla \mathcal{B}_I(\xi)$ 
7.     endfor
8.     for  $I \in \mathcal{I}(E)$  do
9.        $v \leftarrow \mathcal{B}_I(\xi)$ ,  $Dv \leftarrow \nabla \mathcal{B}_I(\xi)$ ,  $U_I^{loc} \leftarrow U_I^{loc} + W |E| (uv + \Delta t F(u, Du, v, Dv))$ 
10.    endfor
11.  endfor
12. endfor
13. synchronized
14.  for  $I \in \mathcal{I}(E)$  do
15.     $U_I^{(t+1)} \leftarrow U_I^{(t+1)} + U_I^{loc}$ 
16.  endfor
17. end

```

The speedup and efficiency of the code are presented in Figure 3. We can draw the following conclusions from the presented plots:

- For linear B-splines for trial and quadratic B-splines for testing and large grids $32 \times 32 \times 32$ and $64 \times 64 \times 64$ the speedup grows up to 16 cores. It is around 10-11 for 16 cores. The corresponding efficiency for 16 cores is around 0.7. Then, for 32 cores the speedup went down since for more than 20 cores used the hyperthreading is utilized.
- For quadratic B-splines for trial and cubic B-splines and large grids $32 \times 32 \times 32$ and $64 \times 64 \times 64$ the speedup grows up to 16 cores. It is around 12-14 for 16 cores. The corresponding efficiency for 16 cores is around 0.8-0.9. Then, for 32 cores and $32 \times 32 \times 32$ mesh the speedup grows up to 17, and for $64 \times 64 \times 64$ mesh is decreases slightly since for more than 20 cores the hyperthreading is used.

- For cubic B-splines for trial and quartic B-splines for testing and large grids $32 \times 32 \times 32$ and $64 \times 64 \times 64$ the speedup grows up to 32 cores. It is around 15 for 16 cores (near perfect speedup) and around 20 for 32 cores, where we use the hyperthreading (more than 20 cores). The corresponding efficiency for 16 cores is around 0.9-1.0. Then, for 32 cores the efficiency decreases slightly down to 0.6-0.7.
- Increasing the mesh size increases the parallel scalability up to $32 \times 32 \times 32$ mesh. Larger mesh, $64 \times 64 \times 64$ performs slightly worse than $32 \times 32 \times 32$ mesh.
- The most interesting observation is that while increasing the B-splines order we observe the improvement of the parallel scalability. This is important from the point of view of the stabilization with the residual minimization method. The order of B-splines in the test space is increased to enforce the stabilization, and when we increase the order to obtain the stabilization, we also improve the parallel scalability.

6 Conclusions

We introduced an isogeometric finite element method for an implicit simulations of the advection-diffusion problem with Douglas-Gunn time-integration scheme that results in a Kronecker product structure of the matrix in every time step. The application of B-spline basis functions for the approximation of the numerical solutions results in a smooth, higher order approximation of the solution. It also enables for the residual minimization stabilization with a linear computational cost $\mathcal{O}(N)$ of the direct solver. The method has been verified on a three-dimensional advection-diffusion problem. Our future work will involve the extension of the model to more complicated equations and geometries. In particular, we plan to use the isogeometric alternating direction implicit solver for tumor growth simulations in two- and three-dimensions [13, 17]. Our equations can also be extended to model a pollution problem, with different chemical components, propagating and reacting together through space in time, as described in [18].

Acknowledgments

The work of Maciej Paszyński, Marcin Łoś and visit of Judit Muñoz-Matute at AGH have been supported by National Science Centre, Poland grant no. 2017/26/M/ST1/00281. The J. Tinsley Oden Faculty Fellowship Research Program at the Institute for Computational Engineering and Sciences (ICES) of the University of Texas at Austin has supported the visit of Maciej Paszyński to ICES.

References

1. D.W. Peaceman, H.H. Rachford Jr., *The numerical solution of parabolic and elliptic differential equations*, Journal of Society of Industrial and Applied Mathematics 3 (1955) 28–41.
2. J. Douglas, J. E. Gunn, A general formulation of alternating direction methods. Numerische Mathematik, 6(1) (1964) 428-453.
3. G. Birkhoff, R.S. Varga, D. Young, *Alternating direction implicit methods*, Advanced Computing 3 (1962) 189–273.
4. J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, *Isogeometric analysis: towards unification of CAD and FEA* John Wiley and Sons, (2009).

5. J. Douglas, H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of American Mathematical Society 82 (1956) 421–439.
6. L. Gao, V.M. Calo, *Fast isogeometric solvers for explicit dynamics*, Computer Methods in Applied Mechanics and Engineering, 274 (1) (2014) 19-41.
7. L. Gao, V.M. Calo, *Preconditioners based on the alternating-direction-implicit algorithm for the 2D steady-state diffusion equation with orthotropic heterogeneous coefficients*, 273 (1) (2015) 274-295.
8. J. L. Guermond, P. Mineev, *A new class of fractional step techniques for the incompressible Navier-Stokes equations using direction splitting*, Comptes Rendus Mathematique 348(9-10) (2010) 581–585.
9. J. L. Guermond, P. Mineev, J. Shen, *An overview of projection methods for incompressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006) 6011–6054.
10. J. Keating, P. Mineev, *A fast algorithm for direct simulation of particulate flows using conforming grids*. Journal of Computational Physics 255 (2013) 486–501.
11. G. Gurgul, M. Woźniak, M. Łoś, D. Szeliga, M. Paszyński, *Open source JAVA implementation of the parallel multi-thread alternating direction isogeometric L2 projections solver for material science simulations*, Computer Methods in Material Science, 17 (2017) 1-11.
12. M. Łoś, M. Woźniak, M. Paszyński, L. Dalcin, V.M. Calo, *Dynamics with matrices possessing Kronecker product structure*, Procedia Computer Science 51 (2015) 286-295.
13. M. Łoś, M. Paszyński, A. Khusek, W. Dzwinel, *Application of fast isogeometric L2 projection solver for tumor growth simulations*, Computer Methods in Applied Mechanics and Engineering, 316 (2017) 1257-1269.
14. M. Łoś, M. Woźniak, M. Paszyński, A. Lenharth, K. Pingali, *IGA-ADS : isogeometric analysis FEM using ADS solver*, Computer & Physics Communications, 217 (2017) 99-116.
15. M. Łoś, M. Paszyński, *Applications of Alternating Direction Solver for simulations of time-dependent problems*, Computer Science 18(2) (2017) 117-128.
16. M. Woźniak, M. Łoś, M. Paszyński, L. Dalcin, V. M. Calo, *Parallel fast isogeometric solvers for explicit dynamics*, *Computing and Informatics* 36 (2), (2017) 423-448.
17. M. Łoś, A. Khusek, M. Amber Hassam, K. Pingali, W. Dzwinel, M. Paszyński, *Parallel fast isogeometric L2 projection solver with GALOIS system for 3D tumor growth simulations*, Computer Methods in Applied Mechanics and Engineering, 343 (2019) 1-22.
18. A. Oliver, G. Montero, R. Montenegro, E. Rodriguez, J.M. Escobar, A. Pérez-Foguet, *Adaptive finite element simulation of stack pollutant emissions over complex terrain*, Energy, 49 (2013) 47-60.
19. E.L. Wachspress, G. Habetler, *An alternating-direction-implicit iteration technique*, Journal of Society of Industrial and Applied Mathematics 8 (1960) 403–423.
20. M. Łoś, J. Muñoz-Matute, I. Muga, M. Paszyński, *Isogeometric residual minimization method (iGRM) with direction splitting for non-stationary advection-diffusion problems*, Computers & Mathematics with Applications (2019) in press.
21. K. Pingali, D. Nguyen, M. Kulkarni, M. Burtscher, M. A. Hassaan, R. Kaleem, T.-H. Lee, A. Lenharth, R. Manevich, M. Mendez-Lojo, D. Prountzos, X. Sui, *The tao of parallelism in algorithms*, SIGPLAN Not. 46 (6) (2011) 12–25. doi:10.1145/1993316.1993501. URL <http://doi.acm.org/10.1145/1993316.1993501>
22. M. A. Hassaan, M. Burtscher, K. Pingali, *Ordered vs. Unordered: A Comparison of Parallelism and Work-efficiency in Irregular Algorithms*, *Proceedings of the 16th ACM Symposium on Principles and Practice of Parallel Programming*, PPOPP '11 (2011)
23. A. Andrew, D. Nguyen, K. Pingali, *Priority queues are not good concurrent priority schedulers*, *European Conference on Parallel Processing*, Springer, Berlin, Heidelberg (2015) 209-221
24. M. Kulkarni, K. Pingali, B. Walter, G. Ramanarayanan, K. Bala, L. P. Kavita Chew, *Optimistic parallelism requires abstractions*, *ACM SIGPLAN Notices* 42(6) (2007) 211-222
25. L. Demkowicz, *Babuška \Leftrightarrow Brezzi*, ICES-Report 0608, 2006, The University of Texas at Austin, USA, <https://www.ices.utexas.edu/media/reports/2006/0608.pdf>
26. I. Babuška, *Error bounds for finite element method*, Numerische Mathematik, 16, (1971) 322-333.

27. F. Brezzi, *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multiplier*, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique 8.R2 (1974): 129-151
28. T. J. R. Hughes, G. Scovazzi, T. E. Tezduyar, *Stabilized methods for compressible flows*, Journal of Scientific Computing, 43 (3) (2010) 343-368
29. L. P. Franca, S. L. Frey, T. J. R. Hughes, *Stabilized finite element methods: I. Application to the advective-diffusive model*, Computer Methods in Applied Mechanics and Engineering, 95(2) (1992) 253-276
30. L. P. Franca, S. L. Frey, *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 99(2-3) (1992) 209-233
31. F. Brezzi, M.-O. Bristeau, L. P. Franca, M. Mallet, G. Rogé, *A relationship between stabilized finite element methods and the Galerkin method with bubble functions*, Computer Methods in Applied Mechanics and Engineering, 96(1) (1992) 117-129
32. L. Demkowicz, J. Gopalakrishnan, *Recent developments in discontinuous Galerkin finite element methods for partial differential equations* (eds. X. Feng, O. Karakashian, Y. Xing). In: vol. 157. IMA Volumes in Mathematics and its Applications, (2014). An Overview of the DPG Method, 149-180
33. M. Paszyński, D. Pardo, V. M. Calo, *Direct solvers performance on h-adapted grids*, Computers & Mathematics with Applications 70 (3), 282-295
34. J. Chan, J. A. Evans, *A Minimum-residual finite element method for the convection-diffusion equations*, ICES-Report 13-12 (2013)
35. D. Broersen, W. Dahmen, R.P. Stevenson, *On the stability of DPG formulations of transport equations*, Mathematics of Computation, 87 (2018) 1051-1082.
36. D. Broersen, R. Stevenson, *A robust Petrov-Galerkin discretisation of convection-diffusion equations*, Computers and Mathematics with Applications, 68(11) (2014) 1605-1618.
37. K. Pingali, D. Nguyen, M. Kulkarni, M. Burtscher, M. A. Hassaan, R. Kaleem, T.-H. Lee, A. Lenharth, R. Manevich, M. Mendez-Lojo, D. Proutzos, X. Sui, *The tao of parallelism in algorithms*, SIGPLAN Not. 46 (6) (2011) 12-25. doi:10.1145/1993316.1993501. URL <http://doi.acm.org/10.1145/1993316.1993501>
38. M. A. Hassaan, M. Burtscher, K. Pingali, *Ordered vs. Unordered: A Comparison of Parallelism and Work-efficiency in Irregular Algorithms*, *Proceedings of the 16th ACM Symposium on Principles and Practice of Parallel Programming*, PPOPP '11 (2011)
39. A. Andrew, D. Nguyen, K. Pingali, *Priority queues are not good concurrent priority schedulers*, *European Conference on Parallel Processing*, Springer, Berlin, Heidelberg (2015) 209-221
40. M. Kulkarni, K. Pingali, B. Walter, G. Ramanarayanan, K. Bala, L. P. Kavita Chew, *Optimistic parallelism requires abstractions*, *ACM SIGPLAN Notices* 42(6) (2007) 211-222