International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 5, Issue 2, Feb - Mar 2016, 99-108

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

ALTERNATING DIRECTIONS SOLVER FOR ISOGEOMETRIC SIMULATIONS OF NON-LINEAR PROBLEMS

MARCIN ŁOŚ, MACIEJ WOŹNIAK & MACIEJ PASZYŃSKI

AGH University of Science and Technology al. Mickiewicza, Krakow, Poland

ABSTRACT

In this paper we present an application of Alternating Direction Solver (ADS) for solution of non-stationary PDE-s with isogeometric finite element method. We illustrate this approach by solving examplary non-stationary three-dimensional problemusing explicit Euler scheme. In particular we focus on the difficult problem of non-linear flow in heterogenous media.

KEYWORDS: Isogeometric Finite Element Method, Non-Stationary Problems, Alternating Direction Solver, Linear Computational Cost, Non-Linear Flow In Heterogenous Media

INTRODUCTION

The Alternating Directions Implicit (ADI) method has been originally introduced in [1-4] to solve parabolic, hiperbolic and elliptic PDE-s. The method has been applied as a solver to two-dimensional non-stationary problem [5], as well as preconditioner for iterative solvers in case of complex geometries [6]. The ADS solver can be applied for solution of non-stationary problems, in particular for the solution of non-linear flow in heterogenous media. The ADS solver transforms the 2D or 3D problem into two or three 1D problems with multiple right-hand-sides. In this paper we present the application of the sequential solver to non-linear flow in heterogenous media [7]. The other applications may include heat transfer problems or linear elasticty problems, solved e.g. with Newmark scheme [8].

Algorithm

1. B-Spline Basis Functions

Classical higher order finite element methods with hierarchical basis functions deliver C0 continuity on the interfaces between elements and C^p continuity inside of elements. The isogeometric finite element method delivers global C^{p-1} continuity inside the elements. Isogeometric basis functions can be represented by recursive Cox-de-Boor formula presented in equation (1).

$$N_{i,0}(\xi) = \begin{cases} 1 \text{ if} \xi_i \le \xi \le \xi_{i+1} \\ 0 \text{ otherwise} \end{cases} \quad N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{1}$$

2. Alternating Direction Solver for Time Dependent Problems

In this section we transform time dependent problem to be used with ADS. First we apply the L^2 isogeometric projection for time the problem.

$$\frac{\partial u}{\partial t} - L(u) = f(x,t) \text{ in } \Omega \times (0,T) \tag{2}$$

$$u(x,0) - u_0(x) \text{ in } \Omega \tag{3}$$

Next we transform time dependent problem into weak form

$$\left(\mathrm{v}, \frac{\partial \mathrm{u}}{\partial \mathrm{t}}\right)_{\Omega} + \mathrm{b(v,u)=(v,f)}_{\Omega} \forall v \in V \tag{4}$$

Where

$$b(u,v) = (v, L(u))_{\Omega} \text{and } (f_1, f_2)\Omega = \int_{\Omega} f_1 f_2 \, dx \tag{5}$$

Finally we can utilize Forward Euler scheme

$$\frac{\partial u}{\partial t} \approx \frac{u_{t+1} - u_t}{\Delta t} \tag{6}$$

$$(v, \frac{u_{t+1} - u_t}{\Delta t} + (v, L(u_t))_{\Omega} = (v, f_t)_{\Omega} \qquad \forall v \in V \tag{7}$$

$$(v, u_{t+1})_{\Omega} = (v, u_t + \Delta t[r_t - L(u_t)])_{\Omega} \forall v \in V \tag{8}$$

3. Alternating Directions Isogeometric L^2 Projection Solver

Following [5, 6] we desribe the the L^2 alternating direction solver that reduces the 2D or 3D L^2 projection problem into 2 or 3 1D problems with multiple right-hand sides. The projection problem can be summarized as $\min \left\| \sum_{i=1} b_i B_i - f \right\|_{L^2}$ which is equivalent to

$$\left(\begin{matrix}
\begin{bmatrix} \int_\Omega (B_1^y B_1^x)(B_1^y B_1^x) & \cdots & \int_\Omega (B_1^y B_1^x)(B_1^y B_{N_x}^x) \\ \vdots & \ddots & \vdots \\ \int_\Omega (B_1^y B_{N_x}^x)(B_1^y B_{1_x}^x) & \cdots & \int_\Omega (B_1^y B_{N_x}^x)(B_1^y B_{N_x}^x) \end{bmatrix} & \cdots & \begin{bmatrix} \int_\Omega (B_1^y B_1^x)(B_{N_y}^y B_1^x) & \cdots & \int_\Omega (B_1^y B_1^x)(B_{N_y}^y B_{N_x}^x) \\ \vdots & \ddots & \vdots \\ \int_\Omega (B_1^y B_{N_x}^x)(B_{N_y}^y B_{1_x}^x) & \cdots & \int_\Omega (B_1^y B_{N_x}^x)(B_{N_y}^y B_{N_x}^x) \end{bmatrix} \\
\vdots & \ddots & \vdots \\
\begin{bmatrix} \int_\Omega (B_{N_y}^y B_1^x)(B_{N_y}^y B_1^x) & \cdots & \int_\Omega (B_{N_y}^y B_1^x)(B_1^y B_{N_x}^x) \\ \vdots & \ddots & \vdots \\ \int_\Omega (B_{N_y}^y B_{N_x}^x)(B_{N_y}^y B_{1_x}^x) & \cdots & \int_\Omega (B_{N_y}^y B_{N_x}^x)(B_1^y B_{N_x}^x) \end{bmatrix} & \cdots & \begin{bmatrix} \int_\Omega (B_{N_y}^y B_1^x)(B_{N_y}^y B_1^x) & \cdots & \int_\Omega (B_{N_y}^y B_1^x)(B_{N_y}^y B_{N_x}^x) \\ \vdots & \ddots & \vdots \\ \int_\Omega (B_{N_y}^y B_{N_x}^x)(B_{N_y}^y B_{1_x}^x) & \cdots & \int_\Omega (B_{N_y}^y B_{N_x}^x)(B_{N_y}^y B_{N_x}^x) \end{bmatrix}
\end{matrix}\right)$$

$$\left(\begin{matrix} \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,N_x} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} b_{N_y,1} \\ \vdots \\ b_{N_y 1,N_x} \end{bmatrix} \end{matrix}\right) = \left(\begin{matrix} \begin{bmatrix} \int_\Omega (B_1^y B_1^x) f \\ \vdots \\ \int_\Omega (B_1^y B_{N_x}^x) f \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \int_\Omega (B_{N_y}^y B_1^x) f \\ \vdots \\ \int_\Omega (B_{N_y}^y B_{N_x}^x) f \end{bmatrix} \end{matrix}\right) \tag{9}$$

when the two dimensional B-spline basis are built from tensor products of two one dimensional basis $B^x = \{B_1, \cdots, B_{N_x}\}$ and $By = \{B_1, \cdots, B_{N_y}\}$.

We notice that

$$\int_\Omega g_1(x) g_2(y) = \int_x \int_y g_1(x) g_2(y) = \int_x g_1(x) \int_y g_2(y) \tag{10}$$

which implies

$$\left(\begin{matrix}
\begin{bmatrix}
\int_y B_1^y B_1^y \int_x B_1^x B_1^x & \cdots & \int_y B_1^y B_1^y \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_y B_1^y B_1^y \int_x B_{N_x}^x B_1^x & \cdots & \int_y B_1^y B_1^y \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix} & \cdots &
\begin{bmatrix}
\int_y B_1^y B_{N_y}^y \int_x B_1^x B_1^x & \cdots & \int_y B_1^y B_{N_y}^y \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_y B_1^y B_{N_y}^y \int_x B_{N_x}^x B_1^x & \cdots & \int_y B_1^y B_{N_y}^y \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix} \\
\vdots & \ddots & \vdots \\
\begin{bmatrix}
\int_y B_{N_y}^y B_1^y \int_x B_1^x B_1^x & \cdots & \int_y B_{N_y}^y B_1^y \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_y B_{N_y}^y B_1^y \int_x B_{N_x}^x B_1^x & \cdots & \int_y B_{N_y}^y B_1^y \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix} & \cdots &
\begin{bmatrix}
\int_y B_{N_y}^y B_{N_y}^y \int_x B_1^x B_1^x & \cdots & \int_y B_{N_y}^y B_{N_y}^y \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_y B_{N_y}^y B_{N_y}^y \int_x B_{N_x}^x B_1^x & \cdots & \int_y B_{N_y}^y B_{N_y}^y \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix}
\end{matrix}\right)
\left(\begin{matrix}
\begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,N_x} \end{bmatrix} \\ \vdots \\
\begin{bmatrix} b_{N_y,1} \\ \vdots \\ b_{N_y,1,N_x} \end{bmatrix}
\end{matrix}\right)
=
\left(\begin{matrix}
\begin{bmatrix} \int_\Omega (B_1^y B_1^x) f \\ \vdots \\ \int_\Omega (B_1^y B_{N_x}^x) f \end{bmatrix} \\ \vdots \\
\begin{bmatrix} \int_\Omega (B_{N_y}^y B_1^x) f \\ \vdots \\ \int_\Omega (B_{N_y}^y B_{N_x}^x) f \end{bmatrix}
\end{matrix}\right) \tag{11}$$

Notice that

$$\left(\begin{matrix}
\begin{bmatrix}
\int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix} & & \\
& \ddots & \\
& & \begin{bmatrix}
\int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\
\vdots & \ddots & \vdots \\
\int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x
\end{bmatrix}
\end{matrix}\right)$$

$$* \left(\begin{matrix}
\begin{bmatrix}
\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_1^y \\
\vdots & \ddots & \vdots \\
\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_1^y
\end{bmatrix} & \cdots &
\begin{bmatrix}
\int_y B_1^y B_{N_y}^y & \cdots & \int_y B_1^y B_{N_y}^y \\
\vdots & \ddots & \vdots \\
\int_y B_1^y B_{N_y}^y & \cdots & \int_y B_1^y B_{N_y}^y
\end{bmatrix} \\
\vdots & \ddots & \vdots \\
\begin{bmatrix}
\int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_1^y \\
\vdots & \ddots & \vdots \\
\int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_1^y
\end{bmatrix} & \cdots &
\begin{bmatrix}
\int_y B_{N_y}^y B_{N_y}^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y \\
\vdots & \ddots & \vdots \\
\int_y B_{N_y}^y B_{N_y}^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y
\end{bmatrix}
\end{matrix}\right)
\left(\begin{matrix}
\begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,N_x} \end{bmatrix} \\ \vdots \\
\begin{bmatrix} b_{N_y,1} \\ \vdots \\ b_{N_y 1,N_x} \end{bmatrix}
\end{matrix}\right)
=
\left(\begin{matrix}
\begin{bmatrix} \int_\Omega (B_1^y B_1^x) f \\ \vdots \\ \int_\Omega (B_1^y B_{N_x}^x) f \end{bmatrix} \\ \vdots \\
\begin{bmatrix} \int_\Omega (B_{N_y}^y B_1^x) f \\ \vdots \\ \int_\Omega (B_{N_y}^y B_{N_x}^x) f \end{bmatrix}
\end{matrix}\right) \tag{12}$$

Also, please notice that

$$\begin{pmatrix} A & & \\ & \ddots & \\ & & A \end{pmatrix}\begin{pmatrix} A^{-1} & & \\ & \ddots & \\ & & A^{-1} \end{pmatrix}=\begin{pmatrix} I & & \\ & \ddots & \\ & & I \end{pmatrix} \tag{13}$$

All sub-matrices are invertible, so we end up with

$$\begin{pmatrix}
\begin{bmatrix} \int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_1^y \\ \vdots & \ddots & \vdots \\ \int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_1^y \end{bmatrix} & \cdots & \begin{bmatrix} \int_y B_1^y B_{N_y}^y & \cdots & \int_y B_1^y B_{N_y}^y \\ \vdots & \ddots & \vdots \\ \int_y B_1^y B_{N_y}^y & \cdots & \int_y B_1^y B_{N_y}^y \end{bmatrix} \\
\vdots & \ddots & \vdots \\
\begin{bmatrix} \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_1^y \\ \vdots & \ddots & \vdots \\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_1^y \end{bmatrix} & \cdots & \begin{bmatrix} \int_y B_{N_y}^y B_{N_y}^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y \\ \vdots & \ddots & \vdots \\ \int_y B_{N_y}^y B_{N_y}^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y \end{bmatrix}
\end{pmatrix} *$$

$$* \begin{pmatrix} \begin{bmatrix} b_{1,1} \\ \vdots \\ b_{1,N_x} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} b_{N_y,1} \\ \vdots \\ b_{N_y 1,N_x} \end{bmatrix} \end{pmatrix} = \begin{pmatrix}
\begin{bmatrix} \int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\ \vdots & \ddots & \vdots \\ \int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x \end{bmatrix}^{-1} & & \\
& \ddots & \\
& & \begin{bmatrix} \int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\ \vdots & \ddots & \vdots \\ \int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x \end{bmatrix}^{-1}
\end{pmatrix}
\begin{pmatrix} \begin{bmatrix} \int_\Omega \left(B_1^y B_1^x\right) f \\ \vdots \\ \int_\Omega \left(B_1^y B_{N_x}^x\right) f \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \int_\Omega \left(B_{N_y}^y B_1^x\right) f \\ \vdots \\ \int_\Omega \left(B_{N_y}^y B_{N_x}^x\right) f \end{bmatrix} \end{pmatrix} =$$

$$= \begin{pmatrix}
\begin{bmatrix} \int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\ \vdots & \ddots & \vdots \\ \int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x \end{bmatrix}^{-1} \begin{bmatrix} \int_\Omega \left(B_1^y B_1^x\right) f \\ \vdots \\ \int_\Omega \left(B_1^y B_{N_x}^x\right) f \end{bmatrix} \\
\vdots \\
\begin{bmatrix} \int_x B_1^x B_1^x & \cdots & \int_x B_1^x B_{N_x}^x \\ \vdots & \ddots & \vdots \\ \int_x B_{N_x}^x B_1^x & \cdots & \int_x B_{N_x}^x B_{N_x}^x \end{bmatrix}^{-1} \begin{bmatrix} \int_\Omega \left(B_{N_y}^y B_1^x\right) f \\ \vdots \\ \int_\Omega \left(B_{N_y}^y B_{N_x}^x\right) f \end{bmatrix}
\end{pmatrix} = \begin{pmatrix} \begin{bmatrix} t_{1,1} \\ \vdots \\ t_{1,N_x} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} t_{N_y,1} \\ \vdots \\ t_{N_y,N_x} \end{bmatrix} \end{pmatrix} \tag{14}$$

Now, we perform the following re-ordering in the block system

$$\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{1,N_x}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{N_y,1}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}\rightarrow\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{1.N_x}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}\qquad\begin{pmatrix}\begin{bmatrix}t_{1,1}\\ \vdots\\ t_{1,N_x}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}t_{N_y,1}\\ \vdots\\ t_{N_y,N_x}\end{bmatrix}\end{pmatrix}\rightarrow\begin{pmatrix}\begin{bmatrix}t_{1,1}\\ \vdots\\ t_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}t_{1.N_x}\\ \vdots\\ t_{N_y,N_x}\end{bmatrix}\end{pmatrix}\tag{15}$$

to get

$$\begin{pmatrix}\begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix}\end{pmatrix}\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{1,N_x}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}=\begin{pmatrix}\begin{bmatrix}t_{1,1}\\ \vdots\\ t_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}t_{1,N_x}\\ \vdots\\ t_{N_y,N_x}\end{bmatrix}\end{pmatrix}\tag{16}$$

which implies

$$\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{1,N_x}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}=\begin{pmatrix}\begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix}^{-1}\begin{bmatrix}t_{1,1}\\ \vdots\\ t_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix}^{-1}\begin{bmatrix}t_{1,N_x}\\ \vdots\\ t_{N_y,N_x}\end{bmatrix}\end{pmatrix}$$

We denote

$$\begin{pmatrix}\begin{bmatrix}d_{1,1}\\ \vdots\\ d_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}d_{1,N_x}\\ \vdots\\ d_{N_y,N_x}\end{bmatrix}\end{pmatrix}=\begin{pmatrix}\begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix}^{-1}\begin{bmatrix}t_{1,1}\\ \vdots\\ t_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}\int_y B_1^y B_1^y & \cdots & \int_y B_1^y B_{N_y}^y\\ \vdots & \ddots & \vdots\\ \int_y B_{N_y}^y B_1^y & \cdots & \int_y B_{N_y}^y B_{N_y}^y\end{bmatrix}^{-1}\begin{bmatrix}t_{1,N_x}\\ \vdots\\ t_{N_y,N_x}\end{bmatrix}\end{pmatrix}$$

and reorder $\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{1.N_x}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}$ back to $\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{1,N_x}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{N_y,1}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}$ to get finally $\begin{pmatrix}\begin{bmatrix}b_{1,1}\\ \vdots\\ b_{1,N_x}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}b_{N_y,1}\\ \vdots\\ b_{N_y,N_x}\end{bmatrix}\end{pmatrix}=\begin{pmatrix}\begin{bmatrix}d_{1,1}\\ \vdots\\ d_{N_y,1}\end{bmatrix}\\ \vdots\\ \begin{bmatrix}d_{1.N_x}\\ \vdots\\ d_{N_y,N_x}\end{bmatrix}\end{pmatrix}$

This algorithm for two dimensional L^2 projection with N unknowns involves solutions of two one dimensional systems with $N^{1/2}$ unknowns and $N^{1/2}$ right hand sides.

This algorithm for three dimensional L^2 projection with N unknowns involves solutions of two one dimensional systems with $N^{1/3}$ unknowns and $N^{1/3}$ right hand sides.

4. Non-Linear Flow in Heterogenous Media

In this section we present the application of the isogeometric L^2 projection solver for the simulation of the problem of nonlinear flows in highly-heterogeneous porous media [12]. The time dependent problem is given by:

$$\frac{\partial u}{\partial t} - L(u) = f(x,t) \text{ in } \Omega \times (0,T) \tag{17}$$

$$u(x,0) = u_0(x) \text{ in } \Omega \tag{18}$$

$$\left(v, \frac{\partial u}{\partial t}\right)_\Omega + b(v,u) = (v,f)_\Omega \forall v \in V \tag{19}$$

We transform the time dependent problem into weak form where:

$$b(u,v) = (v, L(u))_\Omega \tag{20}$$

and

$$(f_1, f_2)\Omega = \int_\Omega f_1 f_2 \, dx \tag{21}$$

We can utilize Forward Euler scheme

$$\frac{\partial u}{\partial t} \approx \frac{u_{t+1} - u_t}{\Delta t} \tag{22}$$

$$(v, \frac{u_{t+1} - u_t}{\Delta t})_\Omega + b(v,u) = (v,f)_\Omega \forall v E v \tag{23}$$

$$(v, ut+1)_\Omega = (v, ut)_\Omega + \Delta t[(v,f)_\Omega - b(v,u)] \forall v \in V \tag{24}$$

Non-linear flow in heterogenous media

$$\frac{\partial u(x)}{\partial t} - \nabla \cdot (K(x) \nabla u) = h(x) \tag{25}$$

Where u-pressure, K-permeability, h-forcing, domain $D = [0,1]^3$

$$h(x) = 1 + sin(2\pi x_1)\, sin(2\pi x_2)\, sin(2\pi x_3) \tag{26}$$

$$K(x,u,\mu) = K_q(x) e^{10\mu} \tag{27}$$

Where $K_q$is the formation map, see Figure 1.

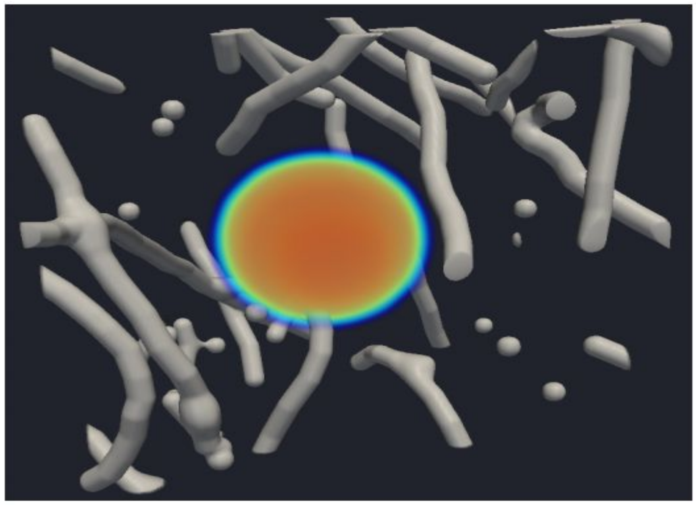


Figure 1: Initial State and the Formation Map (K_q=1000 in the Gray Areas, 0 in Other Places)

We use the time step $\Delta t = 10^{-5}$ due to Currant-Fourrier-Levy (CFL) condition restriction.

We solve the problem over the cube $\Omega = [0,1]^3$ domain. We utilize the isogeometric L^2 projection solver to execute the Euler scheme for the above problem. The time step size has been selected as 10^{-5}. The initial value is a ball with radius 0.05 and maximum value 0.02. The snapshots from the numerical simulation are presented in Figures 2, 3 and 4.

We would like to emphesize that transformation of the 3D problem into three 1D problems with multiple right hand sides results in reduction of the computational cost from quadratic to linear cost, for every iteration of the ADS solver for non-stationary problem.

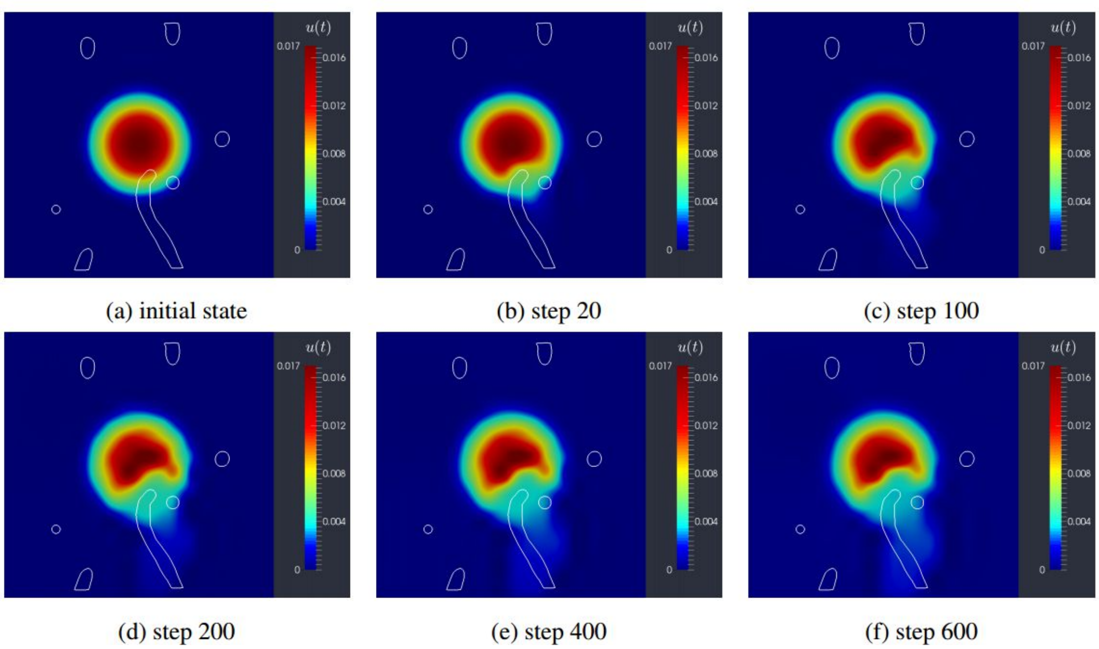
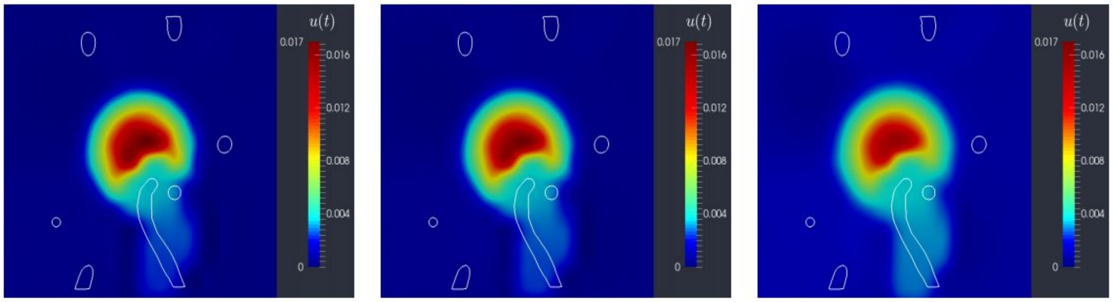
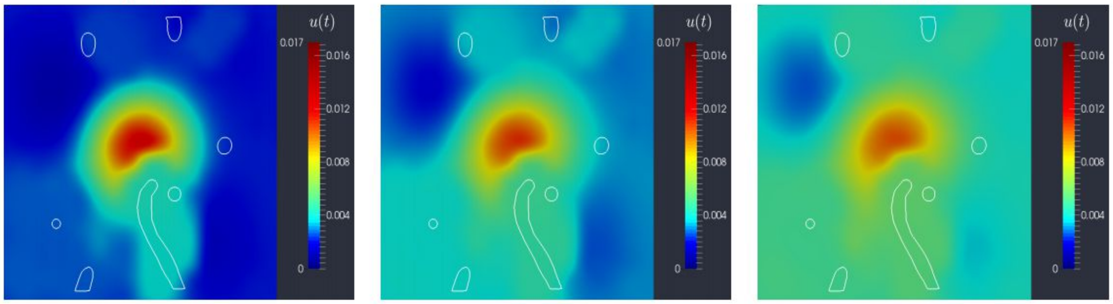


Figure 2: Snapshots from the Numerical Simulation, Time Steps 0, 20, 100, 200, 400 and 600

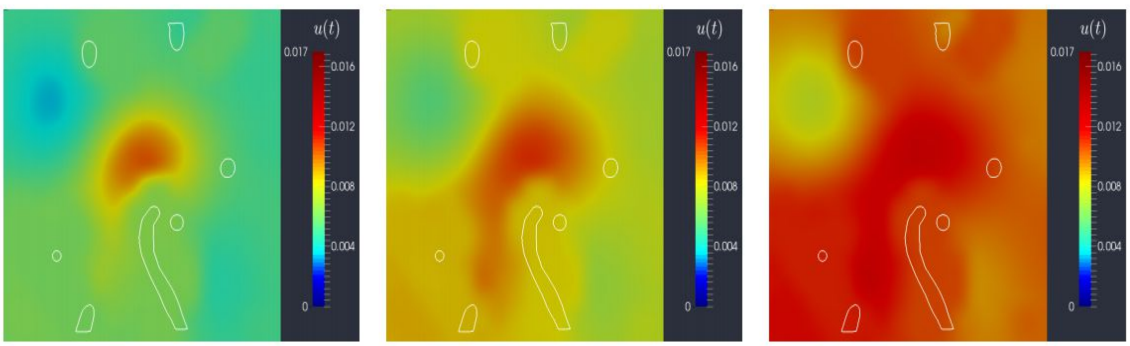


(g) step 800 (h) step 1000 (i) step 2000



(j) step 4000 (k) step 6000 (l) step 8000

Figure 3: Snapshots from the Numerical Simulation, Time Steps 800, 1000, 2000, 4000, 6000 and 8000



(m) step 10000 (n) step 15000 (o) step 20000

Figure 4: Snapshots from the Numerical Simulation, Time Steps 10000, 15000 and 20000

CONCLUSIONS

In this paper we presented the application of the Alternating Direction Solver for solution of non-stationary problems. We have tested the solver on the problem of non-linear flow in heterogenous media [7]. The important extension of this method is the parallelization of the ADS solver, which has been described in our related paper [9]. Future work will include the application of the GPU solver for solution of 1D problems with multiple right-hand sides [10].

ACKNOWLEDGEMENTS

The work presented in this paper has been supported by Polish National Science Center grant no. DEC-2014/15/N/ST6/04662. We would like to express our gratitude to prof. Victor Calo for introducing us into the Alternating Direction Solver algorithms.

REFERENCES

1. Peaceman D W, Rachford Jr. H H. The numerical solution of parabolic and elliptic differential equations. *Journal of Society of Industrial and Applied Mathematics,*1955, 3(1):28-41

2. Douglas J, RachfordH. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of American Mathematical Society,*1956, 82:421-439

3. Wachspress E L, Habetler G. An alternating-direction-implicit iteration technique. *Journal of Society of Industrial and Applied Mathematics,* 1960,8(2): 403-423

4. Birkhoff G, Varga RS, Young D. Alternating direction implicit methods. *Advanced Computing,* 1962, 3: 189-273

5. Gao L, Calo V M. Fast Isogeometric Solvers for Explicit Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 2014,274(1): 19-41

6. Gao L, Calo V M. Preconditioners based on the alternating-direction-implicit algorithm for the 2D steady-state diffusion equation with orthotropic heterogenous coefficients. *Journal of Computational and Applied Mathematics*, 2015, 273 (1): 274-295

7. Alotaibi M, Calo V M, Efendiev Y, Galvis J, Ghommem M. Global-Local Nonlinear Model Reduction for Flows in Heterogeneous Porous Media. *arXiv:1407.0782[math.NA]*

8. Łoś M, Woźniak M, Paszyński M, Dalcin L, Calo V M. Dynamics with Matrices Possessing Kronecker Product Structure.*Procedia Computer Science*, 51 (2015) 286-295.

9. Wozniak M, Łoś M, Paszynski M, Dalcin L, Calo V M. Parallel fast isogeometric solvers for explicit dynamics.accepted to *Computing and Informationcs,* 2015

10. Kuźnik K, Paszyński M, Calo V M.Grammar-Based Multi-Frontal Solver for One Dimensional Isogeometric Analysis with Multiple Right-Hand-Sides.*Procedia Computer Science*, 2013, 18: 1574-1583