

Article

# About Revisiting Domain Decomposition Methods for Poroelasticity

Horacio Florez 

Harold Vance Department of Petroleum Engineering, Texas A & M University, 3116 TAMU, College Station, TX 77843, USA; [florezg@gmail.com](mailto:florezg@gmail.com) or [hflorez@tamu.edu](mailto:hflorez@tamu.edu)

Received: 20 August 2018; Accepted: 27 September 2018; Published: 2 October 2018



**Abstract:** In this paper, we revisit well-established domain decomposition (DD) schemes to perform realistic simulations of coupled flow and poroelasticity problems on parallel computers. We define distinct solution schemes to take into account different transmission conditions among subdomain boundaries. Indeed, we examine two different approaches, i.e., Dirichlet-Neumann (DN) and the mortar finite element method (MFEM), and we recognize their advantages and disadvantages. The MFEM significantly lessens the computational cost of reservoir compaction and subsidence calculations by dodging the conforming Cartesian grids that arise from the pay-zone onto its vicinity. There is a manifest necessity of producing non-matching interfaces between the reservoir and its neighborhood. We thus employ MFEM over nonuniform rational B-splines (NURBS) surfaces to stick these non-conforming subdomain parts. We then decouple the mortar saddle-point problem (SPP) using the Dirichlet-Neumann domain decomposition (DNDD) scheme. We confirm that this procedure is proper for calculations at the field level. We also carry comprehensive comparisons between the conventional and non-matching solutions to prove the method's accuracy. Examples encompass linking finite element codes for slightly compressible single-phase and poroelasticity. We have used this program to a category of problems ranking from near-borehole applications to whole field subsidence estimations.

**Keywords:** domain decomposition; Dirichlet-Neumann; mortar finite elements

## 1. Introduction

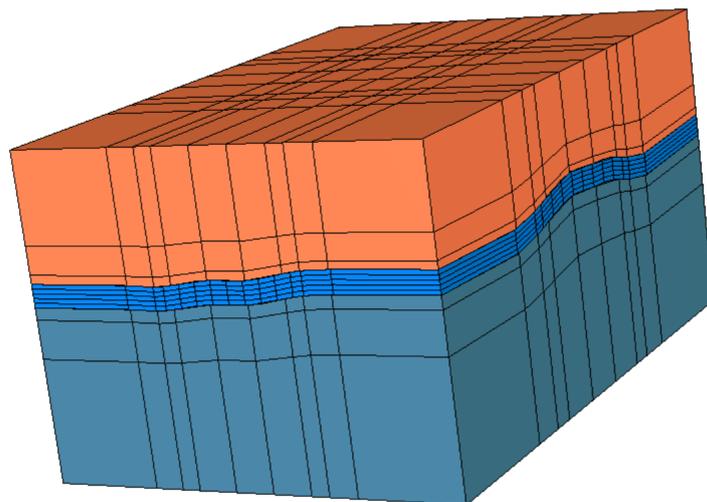
Oil production or fluids injection induce stress variations within the reservoir, driving compaction and surface's subsidence with severe effects in well-casings, cap-rock, fault reactivation, as well as adverse environmental impact. We then require reliable computations to estimate these variations and their influence. Towards that end, we must pair flow simulations with poroelasticity that substantially rises memory consumption and processing time, e.g., the mechanics component can take as much as 90% of the total runtime [1–8]. Domain decomposition procedures have been intensively analyzed to overwhelm the central processing unit (CPU) load related to these predictions, and they involve splitting the whole domain into smaller subdomains in such a fashion that each part can be tackled individually to use parallel computing.

Poroelasticity is the underlying theory to estimate compaction and related risks, i.e., surface subsidence and near-wellbore issues induced by hydrocarbons production [7,9–11]. Rice and Cleary [12] reformulated Biot's theory [13] to express the equations regarding coefficients that are more suitable for practical applications [14,15]. The finite element method (FEM) is the most popular tool to solve the poroelasticity equations. Indeed, Lewis and Schrefler [16] used such techniques to 1- and 2-D consolidation problems. Phillips and Wheeler [17,18] showed the theoretical convergence of 2-D models that pair both continuous (CG) and discontinuous Galerkin schemes for the displacements with mixed finite element (FE) spaces for flow. Sanhu and Wilson [19] published an

early FE application to 3-D consolidation that was reviewed by Yokoo [20]. Ghaboussi and Wilson [21] considered fluid's compressibility as well as the stability of CG schemes [22]. Zienkiewicz et al. [23] incorporated solid grains' compressibility while Booker and Small [24] performed the stability analysis in time. Zienkiewicz and Shiomi [25] tackled various CG approaches for subsurface consolidation. Mixed finite elements [17,18], reduced quadrature rules [26], and penalty factors [27], which were analyzed and applied for tackling incompressibility in elasticity, was purposed for compaction computations coupled with the incompressible fluid flow. Many CG schemes have been used for various practical poromechanics applications. The reader can find particular models in petroleum engineering in [6,28–30].

MFEM is a reliable means to enforce a weaker continuity requirement at the boundary of subdomains in which we use various grids, e.g., non-matching. The technique permits pairing elasticity and poroelasticity on multiple domains, which is a need for flow combined with geomechanics. The reader can find a brief intro to MFEM, its evolution, and traditional references in [1–3,31]. Bernardi et al. proposed MFEM for Poisson's equation to express a relaxed continuity requirement across subdomain boundaries that we use either non-conforming grids or distinct variational forms [32]. We summarized historical and recent developments regarding MFEM in Florez [1]. For instance, we refer the reader to papers in [33–37] for applications for elasticity with curved surfaces in solid mechanics, and dual spaces [38,39]. Also, Florez [1,40] introduced a mortar scheme able to glue subdomains on curved interfaces geometrically modeled using NURBS entities, i.e., curves and surfaces. His paper included computer implementation details.

We introduced in [41] a reservoir geometry rebuilding algorithm that generates matching hexahedral grids that we can employ for the application herein. The simplicity of the earlier procedure advises remodeling the reservoir in an equal way but letting non-matching grids in the various reservoir levels as we depict in Figure 1. We thus reviewed our method in [40,42] to enhance its execution and to enable distinctive grids in the overburden and underburden compared with the pay-zone level. We present herein an MFEM that sticks these three non-conforming subdomains ensuring continuous displacements over the given boundaries that we geometrically represent via NURBS entities. We determine the SPP to require continuity among subdomains, and we further decouple the SPP with the Dirichlet-Neumann domain decomposition method (DNDDM). We briefly introduce MFEM, and we also discuss on calculating the mortar projector including all algorithms in pseudo-code. The preliminary numerical results confirm that MFEM can undertake problems whose size is of practical interest yet at the whole reservoir field.



**Figure 1.** Nonconforming discretization by levels.

The remaining sections are: Section 2 defines the mathematical model and finite element formulation for single-phase flow coupled with poroelasticity. Section 3 comprises MFEM and emphasizes on computing the projection matrix, and thus Section 4 includes uncoupling procedures applying domain decomposition approaches. Section 5 involves numerical examples while the three last sections encompass conclusions, path-forward, and acknowledgments sequentially.

## 2. Governing Equations and Finite Element Formulation

We examine the equations that govern homogeneous linear isotropic poroelasticity and introduce their FE formulation. We exclude all particulars herein for the sake of conciseness. The reader can find a more comprehensive development in [2–4,6,16,22]. We consider a bordered domain  $\Omega \subset \mathbb{R}^3$  and its outskirt  $\Gamma = \partial\Omega$ , and  $\mathcal{T}_h$  is a non-deteriorated, almost even conforming triangulation of  $\Omega$  based on hexahedra. We can show that [43], for flexible porous media, the single-phase flow model develops from the incompressible equation of continuity and Darcy’s law, this is:

$$\frac{\partial\phi^*}{\partial t} + \nabla \cdot \left( -\frac{1}{\mu} \underline{K} (\nabla p - \rho g \nabla z) \right) = q \tag{1}$$

where  $\phi^*$  is a particular algorithm porosity,  $\underline{K}$  means the absolute permeability, fluid properties are the dynamic viscosity,  $\mu$ , and  $\rho$ , the density. Gravity acceleration constant is  $g$ ,  $p$  refers to fluid pressure while  $q$  expresses sources and sinks, e.g., injector and producer wells. The resulting model porosity,  $\phi^*$  appears as:

$$\phi^* = \phi^0 + \alpha \cdot (\nabla \cdot \underline{u} - \varepsilon_v^0) + \frac{1}{M} (p - p^0) \tag{2}$$

above  $\alpha$  means Biot’s constant,  $\underline{u}$  refers the displacement vector while  $\varepsilon_v^0$  comprises the original volumetric strain,  $M$  means Biot’s modulus [15],  $\phi^0$  and  $p^0$  refer to initialization quantities. The common boundary conditions (BCS) for  $p$  involve no-flow or homogenous Neumann, i.e.:

$$\nabla p \cdot \underline{\hat{n}} = 0 \text{ on } \Gamma, \tag{3}$$

we also require the original pressure in the volume, and  $\underline{\hat{n}}$  corresponds to the outer unitary normal vector. For mechanics, we consider the quasi-steady equilibrium equation, i.e., we neglect velocity variations:

$$\begin{aligned} -\nabla \cdot \underline{\underline{\sigma}} &= \underline{f} \text{ in } \Omega ; \Gamma = \Gamma_D^u \cup \Gamma_N^u \\ \underline{u} &= \underline{0} \text{ on } \Gamma_D^u \\ \underline{t} &= \underline{\underline{\sigma}} \cdot \underline{\hat{n}} \text{ on } \Gamma_N^u \end{aligned} \tag{4}$$

where the stress tensor is  $\underline{\underline{\sigma}}$ ,  $\underline{f}$  means the body forces vector, e.g., gravity. BCS often concern given tractions on the part of the border. We can decompose boundary conditions in categories such as Dirichlet, i.e.,  $\Gamma_D^u$ , and Neumann,  $\Gamma_N^u$ , i.e., where we prescribe the remote tractions. Biot’s poroelasticity theory and Hooke’s law determine  $\underline{\underline{\sigma}}$  by:

$$\begin{aligned} \underline{\underline{\sigma}} &= \underline{\underline{C}} : \underline{\underline{\varepsilon}} - \alpha (p - p^0) \underline{\underline{\delta}} \\ \underline{\underline{C}} &= \lambda \underline{\underline{\delta}} \otimes \underline{\underline{\delta}} + 2G \underline{\underline{I}} \end{aligned} \tag{5}$$

where the elastic moduli for linear isotropic elasticity is  $\underline{\underline{C}}$ ,  $\underline{\underline{\delta}}$  means the Kronecker delta whereas  $\lambda$ ,  $G$ , are the Lamé constants, and  $\underline{\underline{I}}$  denotes the fourth order identity tensor. The symmetrical gradient or strain tensor,  $\underline{\underline{\varepsilon}}$  gives:

$$\underline{\underline{\varepsilon}} = \nabla^s \underline{u} = \frac{1}{2} [\nabla \underline{u} + (\nabla \underline{u})^T] \tag{6}$$

We develop a weak form by replacing Equation (2) into Equation (1), and we multiply by a test-function  $v \in H_0^1(\Omega)$ . We then integrate across the domain and employ the theorem of divergence:

$$\int_{\Omega} \left( \frac{1}{M} \frac{\partial p}{\partial t} v + \alpha v \nabla \cdot \underline{u} + \frac{1}{\mu} \underline{K} \cdot \nabla p (\nabla v)^T \right) \cdot dx = \int_{\Omega} q \cdot v dx + \int_{\Omega} \left( \frac{\rho g}{\mu} \underline{K} \cdot \nabla z (\nabla v)^T \right) dx + \int_{\partial \Omega_N^p} v \frac{1}{\mu} \underline{K} (\nabla p - \rho g \nabla z) \cdot \underline{\hat{n}}^T ds \tag{7}$$

We obtain Equation (4) weak form similarly by testing against the virtual displacement,  $\underline{\chi}$ , which yields:

$$\int_{\Omega} (\nabla \underline{\chi})^T : \underline{\sigma} d\Omega = \int_{\partial \Omega_N^u} \underline{\chi}^T \cdot \underline{t} ds + \int_{\Omega} \underline{\chi}^T \cdot \underline{f} d\Omega \tag{8}$$

where  $\underline{t} = \underline{\sigma} \cdot \underline{\hat{n}}$  means tractions exerted in the boundary. The above equation resembles the virtual work theorem. We often take the continuous FE space as a finite-dimensional subset of the Sobolev spaces [44], thus:

$$\mathcal{C}_k(\mathcal{T}_h) = \left\{ v \in L^2(\Omega) : \forall e \in \mathcal{T}_h, v|_e \in \mathbb{P}_k(e) \right\} \tag{9}$$

where  $\mathbb{P}_k(e)$  refers to the space of polynomials of total degree less than or equal to  $k$ ,  $\mathcal{C}_k(\mathcal{T}_h)$  denotes test functions that are continuous across element boundaries. We expand primary variables within the element  $e$ , i.e.,  $p$  and  $\underline{u}$ , as nodal quantities multiplied by shape functions [45,46]:

$$p_e^h(\underline{x}) = (\underline{\Pi}^e)^T \cdot \underline{p}^e ; \underline{u}_e^h(\underline{x}) = \underline{\Psi}^e \cdot \underline{u}^e \tag{10}$$

where  $\underline{\Pi}^e$  and  $\underline{\Psi}^e$  are shape function matrices:

$$\begin{aligned} \Pi_i^e &= \psi_i^e(\underline{x}) \\ \Psi_{ij}^e &= \begin{cases} \psi_k^e(\underline{x}) & \text{if } j = \mathbf{j} \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{j} &= \text{nDOF} \cdot (k - 1) + i ; k = 1, \dots, nn \end{aligned} \tag{11}$$

above  $nn$  means given element's number of nodes while nDOF denotes the number of degrees-of-freedom that ties the space dimension,  $n$ . The engineering strain  $\underline{\hat{\epsilon}}$  corresponds to:

$$\underline{\hat{\epsilon}} = \underline{B} \cdot \underline{u}^e ; \underline{B} = \underline{D} \cdot \underline{\Psi}^e \tag{12}$$

where the linear operator  $\underline{D}$  becomes:

$$\underline{D}^T = \begin{bmatrix} \partial_x & 0 & 0 & \partial_y & \partial_z & 0 \\ 0 & \partial_y & 0 & \partial_x & 0 & \partial_z \\ 0 & 0 & \partial_z & 0 & \partial_x & \partial_y \end{bmatrix} . \tag{13}$$

We finalize replacing Equation (5) into (8) and using (7), which renders the FE formulation, i.e.:

$$\left[ \begin{array}{cc} \underline{0} & \underline{0} \\ \underline{Q}^T & \underline{S} \end{array} \right] \frac{d}{dt} \left\{ \begin{array}{c} \underline{u} \\ \underline{p} \end{array} \right\} + \left[ \begin{array}{cc} \underline{K} & -\underline{Q} \\ \underline{0} & \underline{H} \end{array} \right] \left\{ \begin{array}{c} \underline{u} \\ \underline{p} \end{array} \right\} = \left\{ \begin{array}{c} \underline{f}_u \\ \underline{f}_p \end{array} \right\} \tag{14}$$

the above matrices expressions are:

$$\begin{aligned}
 \underline{\underline{S}} &= \int_{\Omega} \frac{1}{M} \underline{\underline{\Pi}} \cdot \underline{\underline{\Pi}}^T dx; \underline{\underline{Q}} = \int_{\Omega} \underline{\underline{B}}^T \alpha \underline{\underline{\omega}} \cdot \underline{\underline{\Pi}} dx \\
 \underline{\underline{K}} &= \int_{\Omega} \underline{\underline{B}}^T \underline{\underline{C}} \underline{\underline{B}} dx; \underline{\underline{f}}_u = \int_{\partial\Omega_N^u} \underline{\underline{t}} \cdot \underline{\underline{\Psi}}^T ds + \int_{\Omega} \underline{\underline{\Psi}}^T \underline{\underline{f}} \cdot dx \\
 \underline{\underline{H}} &= \int_{\Omega} \frac{1}{\mu} \underline{\underline{K}} \nabla \underline{\underline{\Pi}} \cdot (\nabla \underline{\underline{\Pi}})^T dx; \underline{\underline{\omega}} = (1, 1, 1, 0, 0, 0)^T \\
 \underline{\underline{f}}_p &= \int_{\partial\Omega_N^p} \left( \frac{1}{\mu} \underline{\underline{K}} \nabla p \cdot \underline{\underline{n}} \right) \cdot \underline{\underline{\Pi}} ds + \int_{\Omega} \underline{\underline{\Pi}}^T q \cdot dx + \int_{\Omega} \left( \frac{\rho g}{\mu} \underline{\underline{K}} \cdot \nabla \underline{\underline{\Pi}} (\nabla z)^T \right) dx
 \end{aligned}
 \tag{15}$$

We can arrive at the loosely coupled method in various ways. We present an instance in Equation (16) whereas we first solve displacements by considering pressures from the prior time step. Next, we update them using the latest displacements:

$$\begin{aligned}
 \underline{\underline{K}} \cdot \underline{\underline{u}}^{k+1} &= \underline{\underline{f}}_u + \underline{\underline{Q}} (p^k - p^0) \\
 \underline{\underline{S}}' \cdot p^{k+1} &= \underline{\underline{S}}'' \cdot p^k + \underline{\underline{f}}_p \cdot \Delta t - \underline{\underline{Q}}^T (\underline{\underline{u}}^{k+1} - \underline{\underline{u}}^k)
 \end{aligned}
 \tag{16}$$

and the new matrixes are:

$$\begin{aligned}
 \underline{\underline{S}}' &= \underline{\underline{S}} + \theta \cdot \Delta t \cdot \underline{\underline{H}} \\
 \underline{\underline{S}}'' &= \underline{\underline{S}} - (1 - \theta) \cdot \Delta t \cdot \underline{\underline{H}}
 \end{aligned}
 \tag{17}$$

where the factor of implicitness is  $\theta \in [0, 1]$  while  $\Delta t$  represents the timestep size. We remark that an iterative coupling scheme relies on the loosely coupled by internally iterating to refresh lagged quantities [6,22]. We realized in [1,2] that flow simulations, in particular, multi-phase, demand mass-conservation. Indeed, besides we discretize herein the single-phase flow equation with a Continuous Galerkin (CG) formulation is possible to post-process to produce mass-conservative fluxes locally as indicated in [47]. We believe that CG produces meaningful pressure fields that we can employ for coupling with the mechanics.

### 3. The Mortar Approach

The fundamental goal here is to advance MFEM to stick rounded interfaces such as the one that Figure 1 displays. We can describe MFEM, i.e., for linear isotropic elasticity via the following bilinear forms,  $a$  and  $\beta$ , that become [36,37]:

$$\begin{aligned}
 a(\underline{\underline{u}}, \underline{\underline{v}}) &= \int_{\Omega} \underline{\underline{\varepsilon}}(\underline{\underline{v}})^T \cdot \underline{\underline{C}} \cdot \underline{\underline{\varepsilon}}(\underline{\underline{u}}) dx \\
 l(\underline{\underline{v}}) &= \int_{\partial\Omega^N} \underline{\underline{t}}^T \cdot \underline{\underline{v}} ds + \int_{\Omega} \underline{\underline{f}}^T \cdot \underline{\underline{v}} dx \\
 \beta(\underline{\underline{u}}, \underline{\underline{\Phi}}) &= \int_{\Gamma} [\underline{\underline{u}}]^T \cdot \underline{\underline{\Phi}} ds; [\underline{\underline{u}}] = (\underline{\underline{u}}^{(1)} - \underline{\underline{u}}^{(2)})
 \end{aligned}
 \tag{18}$$

where  $\beta$  enforces for the pasting condition across subdomain boundaries, whereas we require the displacement jump, i.e.,  $[\underline{\underline{u}}]$ , to be zero in a “weak” or integral sense:

$$\begin{cases} a(\underline{\underline{u}}_h, \underline{\underline{v}}_h) + \beta(\underline{\underline{v}}_h, \underline{\underline{\Delta}}_h) = l(\underline{\underline{v}}_h) \\ \beta(\underline{\underline{u}}_h, \underline{\underline{\Phi}}_h) = 0 \end{cases}
 \tag{19}$$

in Equation (19)  $\Phi_h$  is the mortar space while  $v_h$  and  $\Delta_h$  are the weighting and Lagrange multiplier spaces, respectively. Let  $\underline{\mathcal{S}} = \underline{\mathcal{S}}(\xi, \eta) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  be a two-parameter mapping that describes a given mortar geometry, i.e., NURBS surface, where the parameters  $(\xi, \eta)$ , correspond to orthogonal directions related with the surface in the reference or computational plane,  $\bar{\Omega}$ , a rectangle whose image becomes the surface in 3-D. Let  $\bar{\mathcal{T}}_h^M$  be a conforming triangulation of the reference plane,  $\bar{\Omega}$ , whose mapping or image depicts the mortar as a discrete geometrical object, e.g., a quadrilateral surface. We also take the mortar space as a finite-dimensional subset of the continuous set of Sobolev spaces:

$$C_k(\bar{\mathcal{T}}_h^M) = \left\{ \Phi \in L^2(\bar{\Omega}) : \forall e^M \in \bar{\mathcal{T}}_h^M, \Phi|_{e^M} \in \mathbb{P}_k(e^M) \right\} \tag{20}$$

where  $\mathbb{P}_k(e^M)$  means the set of polynomials whose total degree is less than or equal to  $k$ ,  $C_k(\bar{\mathcal{T}}_h^M)$  denotes continuous test functions across  $e^M$  edges. We can write (19) in matrix form as:

$$\begin{bmatrix} \underline{K}^{(1)} & \underline{0} & \underline{\beta}^{(1),T} \\ \underline{0} & \underline{K}^{(2)} & -\underline{\beta}^{(2),T} \\ \underline{\beta}^{(1)} & -\underline{\beta}^{(2)} & \underline{0} \end{bmatrix} \cdot \begin{bmatrix} \underline{u}^{(1)} \\ \underline{u}^{(2)} \\ \underline{\Delta} \end{bmatrix} = \begin{bmatrix} \underline{l}^{(1)} \\ \underline{l}^{(2)} \\ \underline{0} \end{bmatrix} \tag{21}$$

The above equation is the so-called saddle-point problem (SPP). Notice that subdomains are only glued by the Lagrange multiplier  $\underline{\Delta}$ , if these latter were known (i.e., for elasticity, they are the uncomputed tractions on the subdomain interfaces), then we can unlink the system (21). For further details, please refer to the following references [1,48,49].

We denote the rectangular matrixes  $\underline{\beta}^{(i)}, i = 1, \dots, 2$ , “projectors” because they permit us to map onto and from the mortar space. For 3-D problems, they are defined by:

$$\beta_{ij}^{(k)} = \iint_{\bar{\Omega}} \varphi_j^{(k)}(\underline{\xi}) \Phi_i(\underline{\xi}) \cdot \left\| \partial_{\xi} \underline{\mathcal{S}}(\underline{\xi}) \times \partial_{\eta} \underline{\mathcal{S}}(\underline{\xi}) \right\| d\xi d\eta \tag{22}$$

where  $\left\| \partial_{\xi} \underline{\mathcal{S}} \times \partial_{\eta} \underline{\mathcal{S}} \right\|$  represents the surface’s normal vector metric and  $\underline{\xi} = (\xi, \eta)^T$  is the vector of parameters. We already presented implementation details in Florez [1]. Herein we extend the previous treatment, and we keep some of the algorithms already reported for the sake of completeness. MFEM requires a pre-processing step to ensuring that most computations are explicit [1], which implies possessing certain data structures. Such a step encompasses the geometrical modeling of the mortar interfaces, and surface and volume mesh generation. Algorithm 1 depicts high-level steps that preprocessor executes, where  $nMOR$  and  $nSD$  are the numbers of mortar entities and subdomains, and the mortar and non-mortar quadrilateral grids, in the physical plane, are represented as  $\mathcal{T}_h^M$  and  $\mathcal{T}_h^{NM}$  separately. The function  $dim_h(\cdot)$  yields the argument’s number of elements. We denote non-mortar and mortar sides as NM- and M-side respectively. For instance, Figure 2 depicts the topology that we are interested in, i.e., three non-conforming levels. We anticipate the level in the middle, i.e., the pay-zone, to be the finer one. The figure also exhibits a sample mortar surface that is an interpolation NURBS bicubic surface. The color map resembles the  $u \equiv \xi$  surface’s parameter.

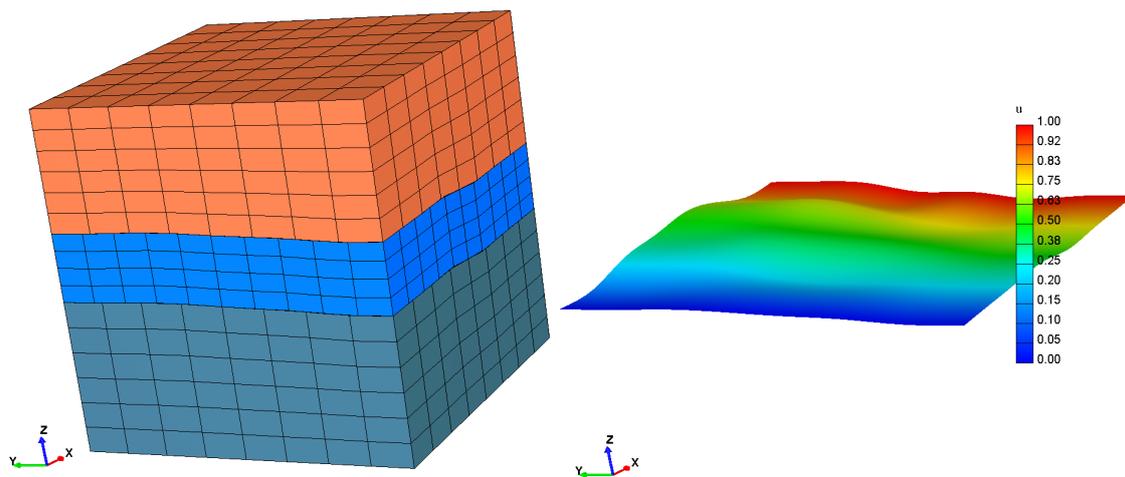


Figure 2. Nonconforming discretization subdomains and one mortar surface.

In Algorithm 1 mesh generation begins with surface discretization. Over mortar interfaces,  $\underline{\mathcal{S}}_j$ , though, we have to save both physical  $\underline{x}_k$  and parametric  $(\zeta_k, \eta_k)$  coordinates. Having the former, i.e.,  $(\zeta_k, \eta_k)$ , bypasses determining them [1]. Before producing a volume mesh on the subdomains, we enforce preservation of the NM-side’s surface grid points. We then map the surface-to-volume numeration. We will require such mapping table while we are assembling the global SPP or passing messages for the decoupled case. The table designates the global degrees-of-freedom (DOF) that we have to glue (notice Figure 3).

---

**Algorithm 1:** Preprocessing high-level steps.

---

**Data:** Geometrical representation of  $\Omega$  and mortar interfaces  $\underline{\mathcal{S}}_j$   
**Result:** Input data structures for MFEM computation

```

1 for  $j = 1; j \leq nMOR; j = j + 1$  do
2   for  $i = 1; i \leq 2; i = i + 1$  do
3      $\mathcal{T}_h^{NM} =$  Generate mesh over  $\underline{\mathcal{S}}_j$  in the  $ji$ -th NM-side;
4     Store nodal points  $\underline{x}_k$  and corresponding  $(\xi_k, \eta_k)$  pairs;
5 for  $i = 1; i < nSD; i = i + 1$  do
6   Generate volume mesh in  $\Omega_i$  honoring NM-side surface meshes;
7   for  $j = 1; j < nMOR; j = j + 1$  do
8     if  $\partial\Omega_i \cap \underline{\mathcal{S}}_j$  then
9       Store surface-to-volume mapping on this NM-side;
10      break;
```

---

Algorithm 2 outlines the high-level steps to compute the projector while Figure 4 reveals the mortar and non-mortar grids in the computational plane. These grids correspond to the problem that Figure 2 describes. To obtain a given point’s (PT) hat function (HFNT), we first compute the compact-support (CS),  $\mathcal{N}_{PT}(\mathcal{T}_h)$ , of that PT, i.e., the array of all elements (ELMT) that include the given PT. For instance, for quadrilateral meshes, the number of ELMT in the CS can be one, two or four. ELMT in the corners, in the boundary and interior ones (excluding corners), explain these numbers. Figure 4 also depicts the HFNT, that exists only on the CS of the given PT. Notice that the CS has an associated convex-hull, outside of this box the HFNT is nil. Steps 2 and 6, in Algorithm 2, correspond to computing these CS and their HFNT.

**Algorithm 2:** High-level steps to compute the projector.

```

Data: Mortar interfaces  $S_i$  and space  $\Phi_h$  and NM-sides surface data
Result: Mortar projectors  $\beta^{(ji)}$ 
1 for  $j = 1; j \leq nMOR; j = j + 1$  do
2   Compute  $dim_h(\overline{\mathcal{T}}_h^M)$  hat functions;
3   for  $i = 1; i \leq 2; i = i + 1$  do
4     Obtain  $ji$ -th NM-side data;
5     Map NM-side mesh to computational space  $\overline{\Omega}$ ;
6     Compute  $dim_h(\overline{\mathcal{T}}_h^{NM})$  hat functions;
7     for  $k = 1; k \leq dim_h(\mathcal{T}_h^M); k = k + 1$  do
8       for  $l = 1; l \leq dim_h(\mathcal{T}_h^{NM}); l = l + 1$  do
9         if  $\varphi_k^M \cap \Phi_l^{NM}$  then
10           $\beta_{kl}^{(ji)} = \text{Integral in Eq. (22)}$ ;

```

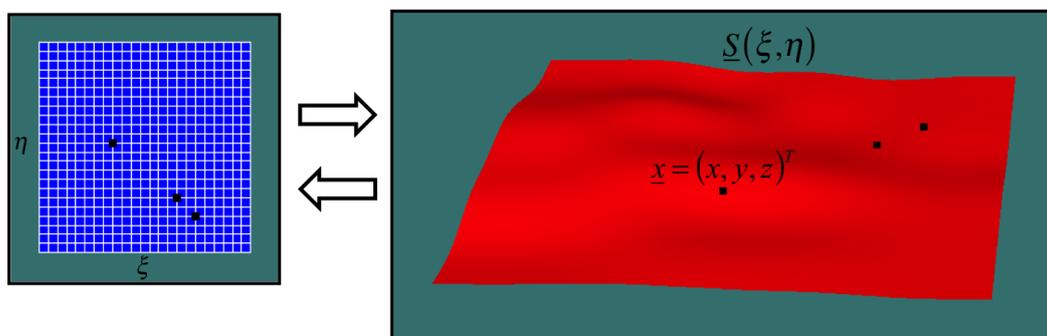


Figure 3. Physical space to computational mappings and vice-versa.

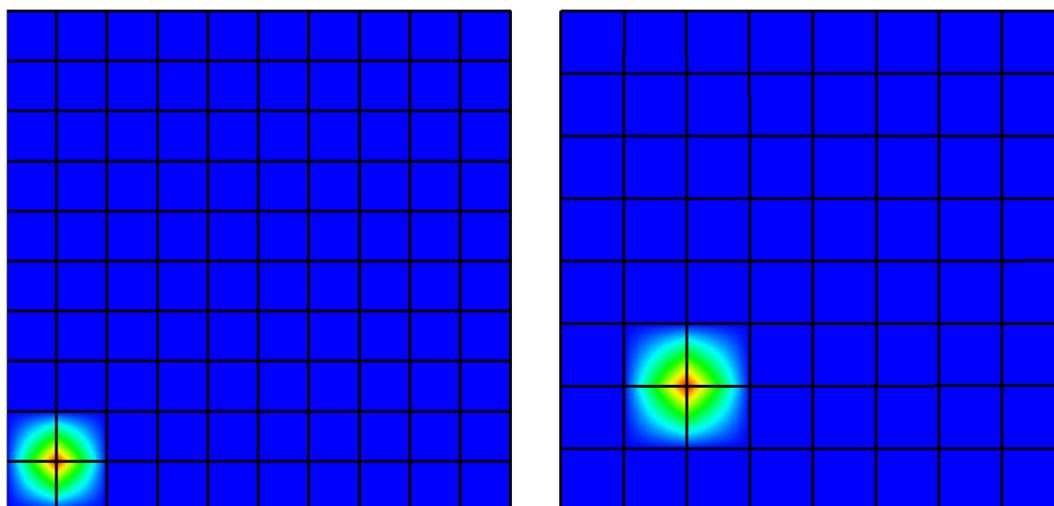


Figure 4. Mortar and non-mortar hat functions for their respective pair of mesh points.

Now, the conditional in step 9 intersects the convex hulls of the given CS. Notice that the intersection can be empty. Figure 5 represents the intersection in which the product of HFNT exists. The volume beneath this surface is the given projector entry, according to Equation (22). To accurately compute the integral, we exploit the resulting box in four quadrilateral elements as shown, and we then use a six order Gauss-Legendre quadrature rule to evaluate every sub-integral. This latter provides enough accuracy for our purposes. Since the computational burden to calculate the projector lies in crossing CS, then we can implement an efficient range-search algorithm based on quadtrees [1,40] to expedite this task.

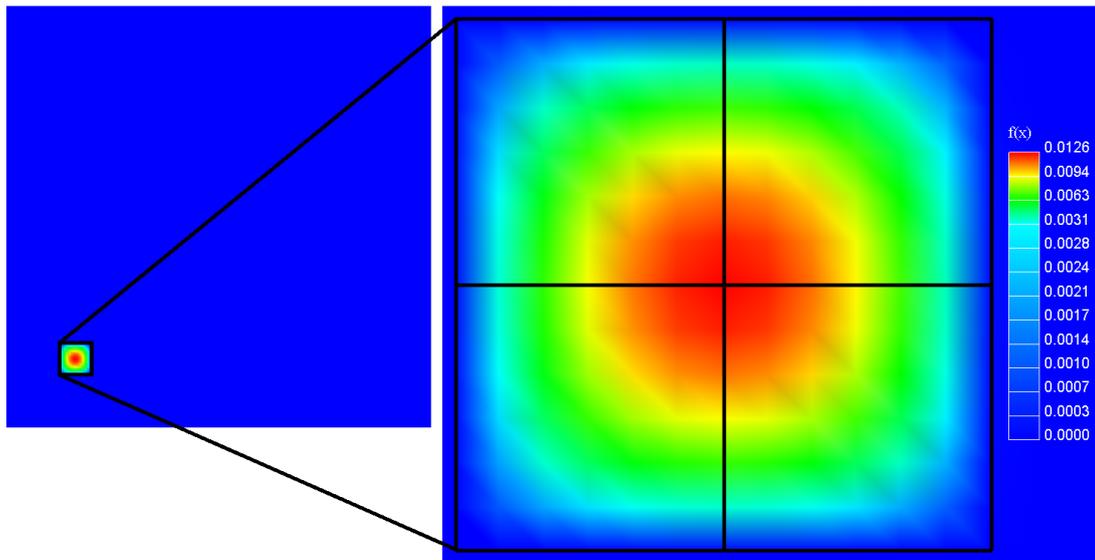


Figure 5. Zoom-in the given hat-function.

Finally, Figure 6 depicts the mortar projector as a matrix, for the problem described in Figure 5, its dimensions are  $(11 \times 11) \times (9 \times 9)$ . It is indeed, in this case, a symmetric matrix. The highest values correspond to those entries where the mortar and non-mortar points are at their closest locations. The mortar projector resembles a  $L^2$  projector which permits mapping onto the mortar space by using the matrix  $\underline{\underline{\beta}}^{(i)}$ . Also, mapping to a non-mortar side relies on another method that we particularly described in [1].

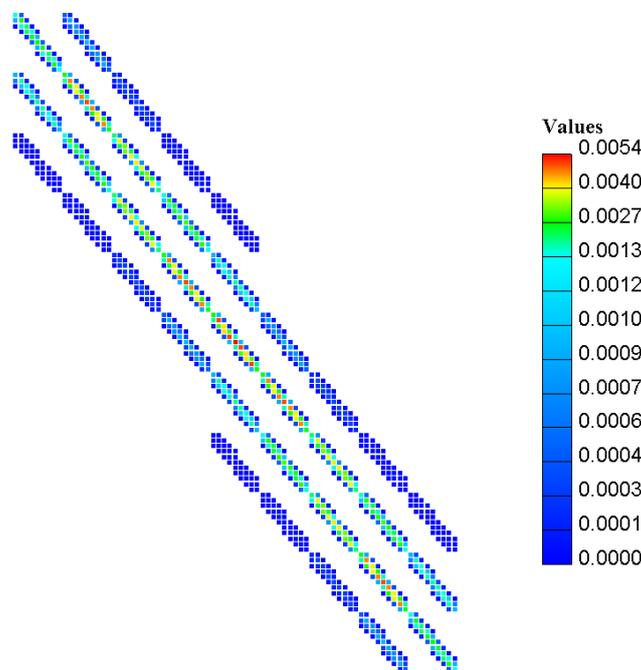


Figure 6. The mortar projector is a sparse matrix.

#### 4. Domain Decomposition Techniques

Domain Decomposition Methods (DDM) are robust algorithms to determine sizable problems on high-performance computing machines. These techniques partition the global domain into various

subdomains to determine the overall answer through iterative-coupling, i.e., solving the unknown BCS on the interfaces of them [5,50]. There is vast literature relating these schemes, but we present a simple foundation herein for the sake of completeness. Bjørstad and Widlund [51], Bramble et al. and Marini and Quarteroni [52] examined the Dirichlet-Neumann scheme (DNDDM), for instance, among other scholars [53].

Let  $\mathcal{L}$  be a symbolic differential linear operator, i.e., Laplace’s operator for example. The DNDDM scheme comprises determining a sequence of problems in the proper chain that demands a coloring tool (see Figure 7). We paint the Dirichlet subdomains in white color while Neumann’s are black. Notice too that the border separating subdomains is  $\Gamma$ . After guessing initial values for the primary variables over  $\Gamma$ , i.e., we must provide  $\gamma^k$ , we can then resolve the problem on the white subdomains (type Dirichlet) that corresponds to step 1 in Equation (23).

$$(1) \begin{cases} \mathcal{L}u_1^{k+1} = f \text{ in } \Omega_1 \\ u_1^{k+1} = 0 \text{ on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{k+1} = \gamma^k \text{ on } \Gamma \end{cases} \quad (2) \begin{cases} \mathcal{L}u_2^{k+1} = f \text{ in } \Omega_2 \\ u_2^{k+1} = 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \\ \partial_n u_2^{k+1} = \kappa^{k+1} \text{ on } \Gamma \end{cases} \quad (23)$$

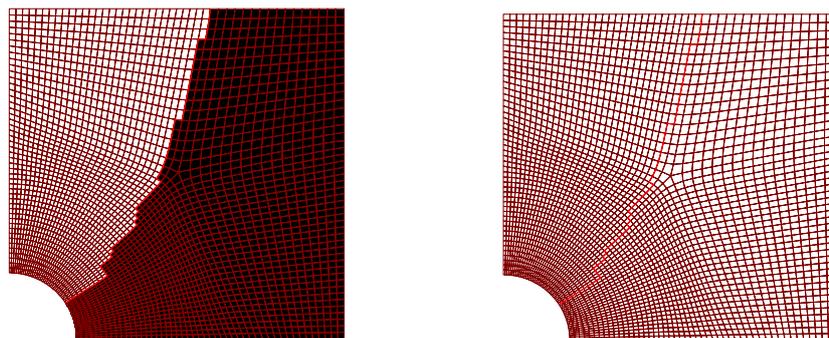


Figure 7. Dirichlet-Neumann (DN, left) and the Neumann-Neumann (right) schemes.

We denote the primary variable “displacements” while “tractions” correspond to their orthogonal derivative in the border. We calculate the former on the boundary  $\Gamma$  after determining the first step on the white subdomains. We then transfer them to complete the second step on the Neumann subdomains. On them, since we know the tractions, we thus determine the displacements to update them for the next iteration. We relax both quantities to improve the convergence where the loosening parameters  $\theta^D$  and  $\theta^N$  belong to  $I = [0, 1]$  as shown in Equation (24):

$$\begin{aligned} \kappa^{k+1} &= \left( -\theta^N \cdot \partial_n u_1^{k+1} + (1 - \theta^N) \cdot \partial_n u_2^k \right) \text{ on } \Gamma \\ \gamma^{k+1} &= \left( \theta^D \cdot u_2^{k+1} + (1 - \theta^D) \cdot u_1^k \right) \text{ on } \Gamma \end{aligned} \quad (24)$$

We repeat scheme (23) until the traction residual, namely,  $\|\partial_n(u_2 - u_1)\|_\Gamma$ , in the interface lies below a given tolerance. We also fix an allowable number of iterations to stop the process in the event of divergence. It happens that this approach entails at minimum a two-color palette for coloring or yet more [5]. We lose parallelism since black subdomains need to wait for the whites to report their tractions. We might provide an initial-guess for tractions to alleviate this difficulty, but it is not simple for most situations. We can easily produce such guess for the multiplier  $\gamma^k$  by determining a coarser-run that intends obtaining the answer over a coarser grid and then interpolating  $\gamma^k$  on  $\Gamma$ . We refer the reader to [54,55] for more details.

## 5. Numerical Cases

The author coded these FE formulations into a parallel C++ application termed “Integrated Parallel Finite Element Analysis (IPFA)” whose principal features were described in [2,4,5]. We ran all samples herein on a MacBook Pro laptop armed with an Intel(R) Quad-Core(TM) i7-7920HQ CPU @ 3.1GHz and sixteen GB of RAM. Notice that, we attain parallelism because the CPU is multi-core. Our iterative linear solver, i.e., conjugate gradient, uses incomplete LU decomposition (ILU) as the preconditioner, and we also use a postprocessor called “LogProc” [56]. Currently, both IPFA and LogProc are proprietary applications [56], but a free LogProc’s community version will be available for download. We are also considering to offer IPFA as an open-source project. We are still defining a timetable to achieve that, and we thus refer readers interested to the website listed in the above reference.

Our continuous shape functions for the spatial discretization are Lagrange polynomials,  $\mathbb{P}_k(e)$ , as well as mortar entities  $\mathbb{P}_k(e^M)$ . We only consider piecewise linear polynomials and the following numerical values for  $\theta^D = \theta^N = 0.5$  for all DN examples below. We include four sample problems next. The first one tackles single-phase steady-state flow where we prescribe the pressure field to allows us to compute errors in  $L^2$  for MFEM via SPP and DN. The second one corresponds to the near-borehole section that we solve both serially, in parallel, and with MFEM. We provide time data and relative errors on the displacements to compare these DD schemes. The last two examples encompass coupled flow and mechanics computations on realistic reservoirs to prove the application of the recommended approaches to industrial size problems. We also estimate relative errors on displacements for comparison purposes.

### 5.1. Example 1: Manufactured Steady-State, Single-Phase Flow

The example encompasses a fabricated problem where we propose a pressure field, and then plug it into the continuity equation to obtain the reciprocal source term. The problem’s partial differential equation is as follows:

$$\begin{aligned} -\nabla \cdot (\underline{K} \nabla p) &= f \text{ in } \Omega, \\ p &= p_0 \text{ on } \Gamma_D = \Gamma, \end{aligned} \quad (25)$$

we consider a unitary cube as the domain with Dirichlet BCS. We assume the pressure to be:

$$p(x, y, z) = 10^3 xyz \cdot (x - 1) \cdot (y - 1) \cdot (z - 1) \cdot \exp \left[ - \left( x^2 + y^2 + z^2 \right) \right]; \underline{K} = \underline{I}, \quad (26)$$

knowing  $p$  allows us computing the error of our MFEM approximation.

Figure 8 shows the three subdomains whose meshes are hexahedral. Table 1 summarizes the metrics of these meshes. A NURBS interpolation surface represents the mortar geometry, whose finite element space corresponds to a tensor-product grid of  $(20 \times 20)$  linear piecewise rectangular elements. We employed a frontal-direct solver to determine the global SPP (21) [1]. The results that Figure 9 depicts have good accordance with the assumed pressure field. The  $L^2$  error is  $5.68 \times 10^{-1}$  in the pay-zone level for instance.

Snapshots in Figure 10 display unfolding matching steps for DN, where the iteration levels progress in reading order as indicated. The discontinuities among subdomains are evident in the three first snaps. The DNDDM used twelve internal iterations to lessen the residual in tractions below  $10^{-4}$ .

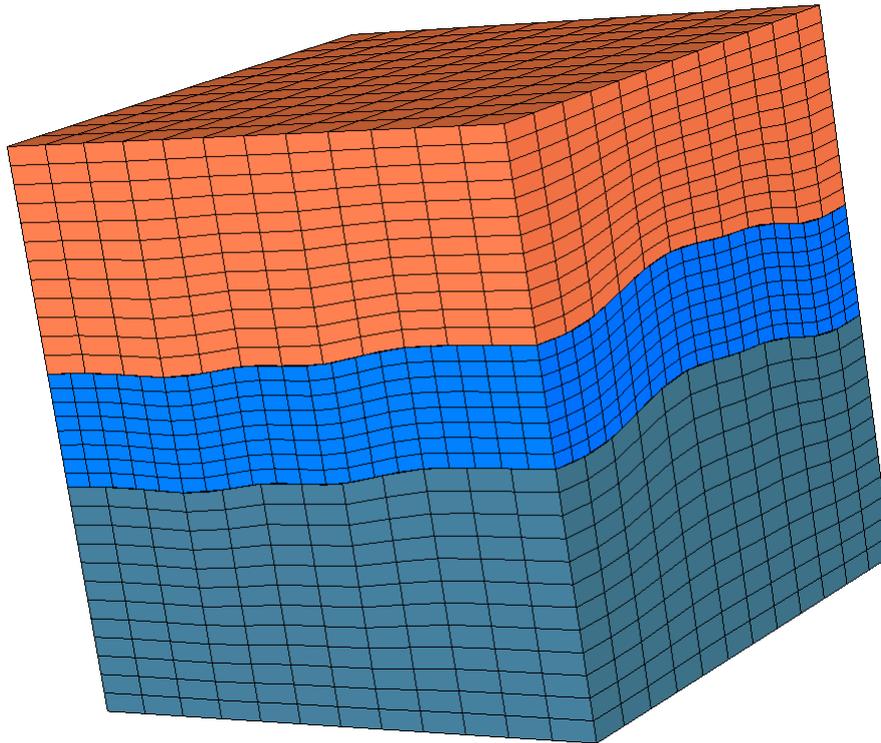


Figure 8. Subdomain meshes in Example 1 in Section 5.1.

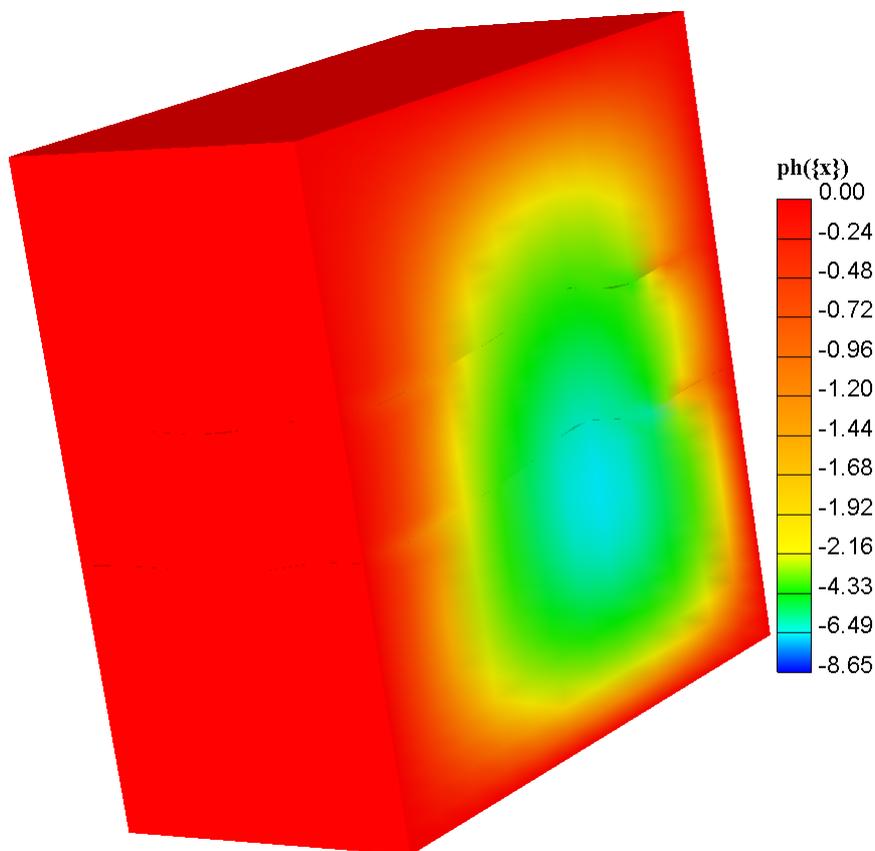


Figure 9. SPP (saddle-point problem) solution to Example 1 in Section 5.1.

Table 1. Grids for Example 1 in Section 5.1.

Level	# Elements	# Points	Coloring
Overburden	1728	2197	Dirichlet
Pay-zone	3200	3969	Neumann
Underburden	1728	2197	Dirichlet

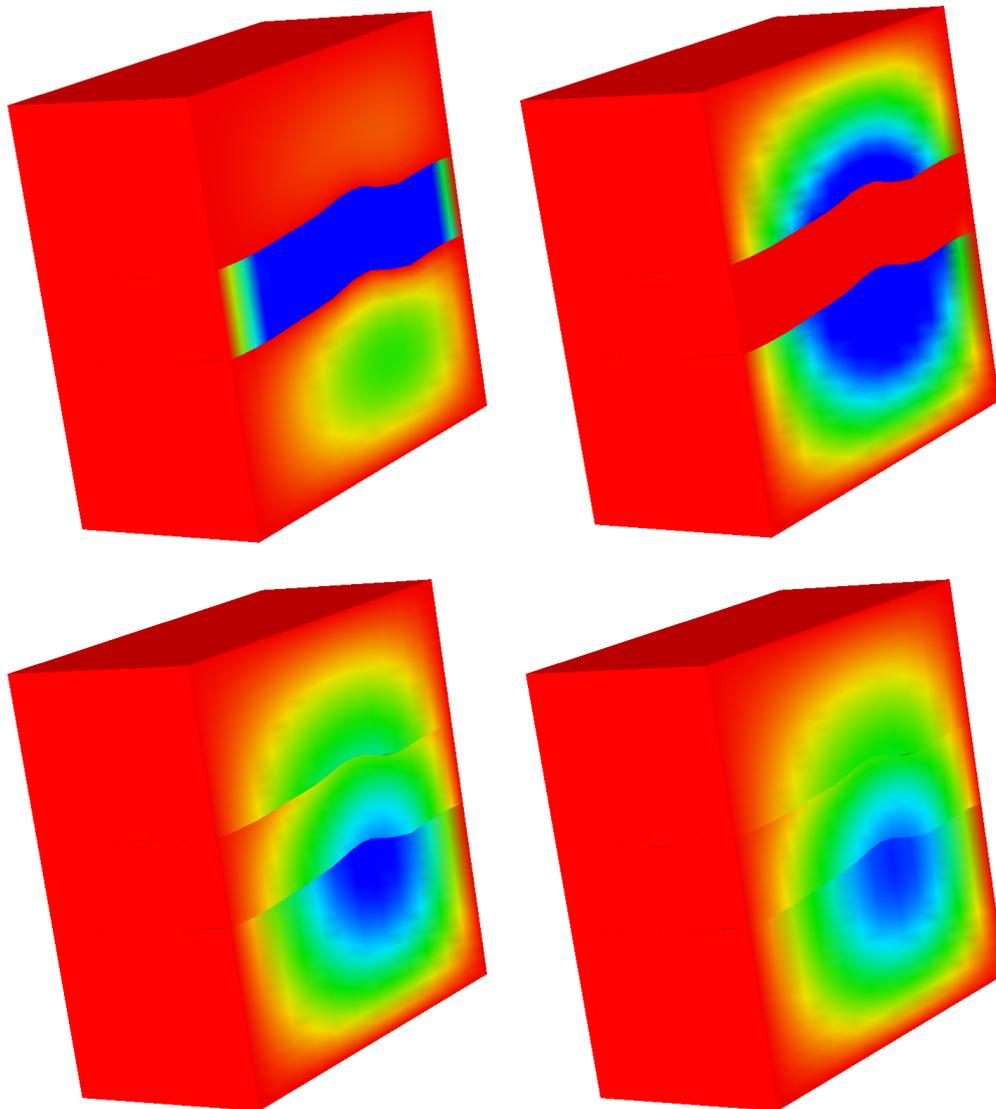


Figure 10. Pressure variation during DN steps.

Figure 11 pictures the final parallel answer, which agrees well with the above-assumed pressure. The  $L^2$ -error for the Dirichlet subdomain is about  $3.60 \times 10^{-1}$ , and for the Neumann improves to  $4.05 \times 10^{-2}$ . The resulting error values are small related to the magnitude of  $p$ .

We now proceed to compare the DNDDM and MFEM for a simple manufactured 2-D problem. We now assume for Equation (25) the following pressure field:

$$p(x, y) = 100 \times xy \cdot (x - 1) \cdot (y - 1) \cdot \exp \left[ - \left( x^2 + y^2 \right) \right] ; \underline{K} = \underline{I} \tag{27}$$

and the domain is the unitary square with pure homogeneous Dirichlet boundary conditions.

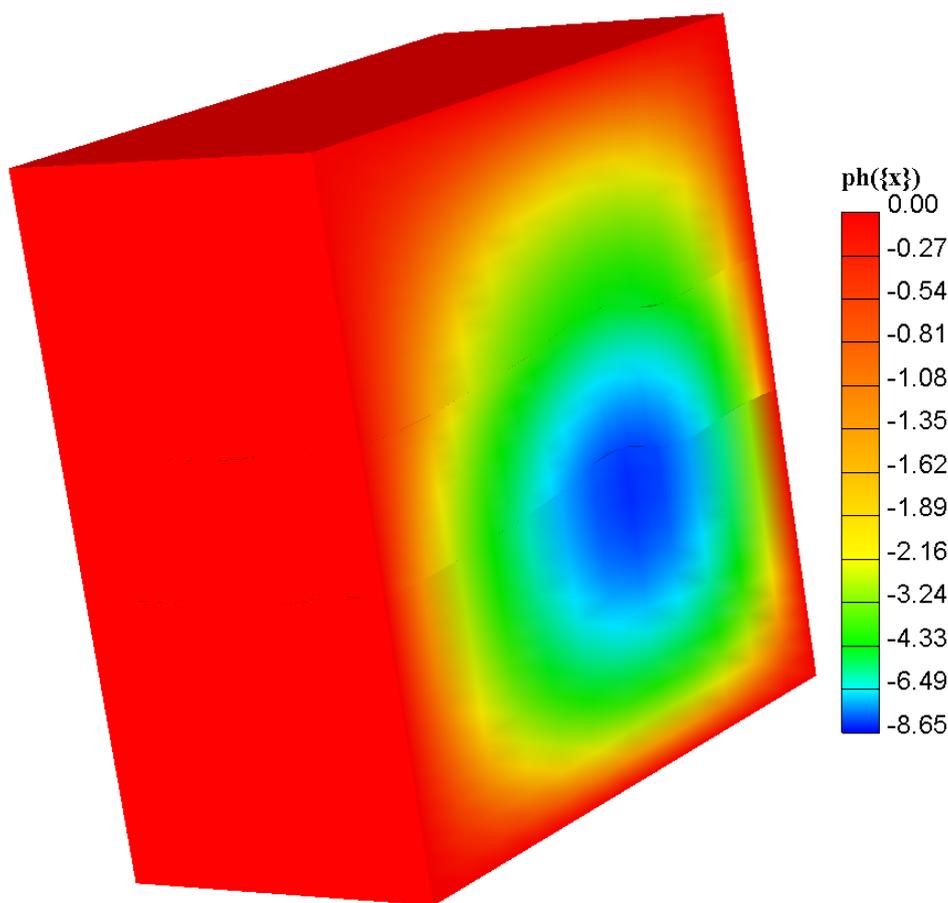


Figure 11. DNDDM solution to Example 1 in Section 5.1.

Figure 12 depicts the meshes partitioning and coloring to solve the problem with both schemes. In the left-hand side, we employ for the DNDDM a global conforming mesh that encompasses 9600 quadrilateral elements, and we partition it into three almost even subdomains as depicted. We color the center domain as Neumann and the remainders as Dirichlet type. Similarly, for MFEM we consider three subdomains, two of them (the top and bottom ones) are triangular meshes while the one in the middle is a regular Cartesian quadrilateral mesh as shown in the right-hand-side of the figure. Table 2 describes the type, the number of elements and points of each mesh from top to bottom. We first focus on the MFEM problem and thus the mortars as geometrical entities consist of two B-Splines interpolants that we constructed by interpolating a sinusoidal wave as revealed. We employed thirty-two quadratic mortar elements per curve to stick these three subdomains by solving the global saddle-point problem (21) with a direct frontal solver [57]. The results depicted on Figure 13 reproduce the analytical solution, we thus also display the absolute error against the later, which is on the order of  $10^{-2}$ . The error increases towards the interfaces as expected.

Table 2. DNDDM and MFEM meshes.

DNDDM				MFEM		
Kind of mesh	Elements	Points	Type	Kind of mesh	Elements	Points
Hexahedral	3200	3321	Dirichlet	Triangular	1814	980
Hexahedral	3120	3240	Neumann	Quadrilateral	1472	1560
Hexahedral	3280	3402	Dirichlet	Triangular	7858	4090

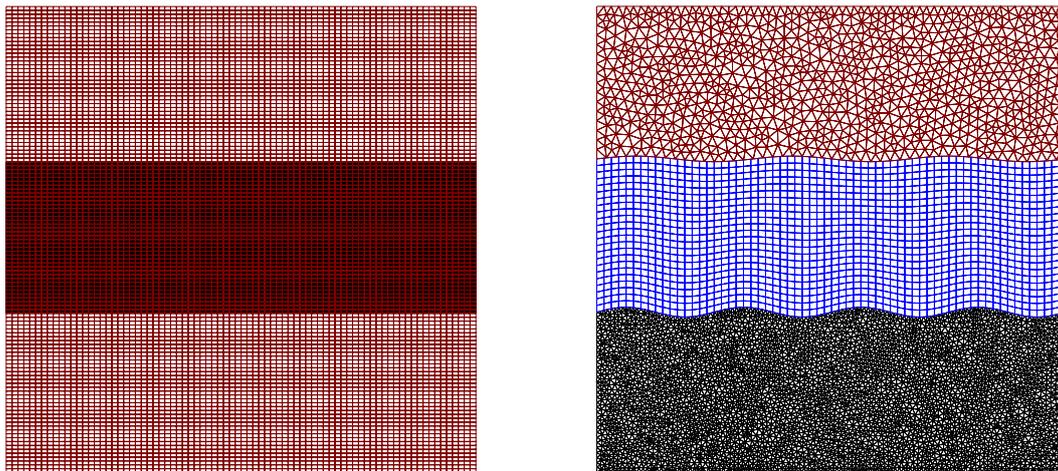


Figure 12. Partitioning for DN (left) and MFEM (right) problems.

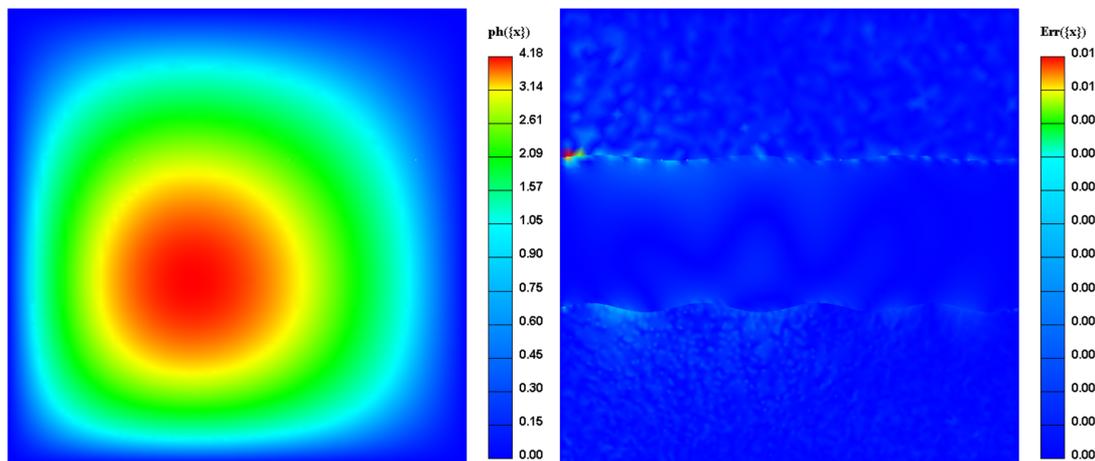


Figure 13. MFEM solution to problem (25) with pressure field (27) (left) and its error (right).

We tried to manufacture a good benchmark problem to compare these schemes, but the DN subdomains have a more refined mesh that the MFEM ones. However, for the topological point of view, the meshes are equivalent. MFEM offers the flexibility of gluing nonmatching interfaces while DN operates over conforming ones. We wish to emphasize that these schemes are rather complementary than competitive. Indeed, we foresee that DN will help us decouple MFEM’s SPP for realistic 3-D problems via the mortar mappings that we presented in [1].

Figure 14 displays pressure snapshots that correspond to four different Dirichlet-Neumann iteration levels, namely 2, 4, 6 and 8, evolving from left-to-right and top-to-bottom. We employ the mesh in the left-hand-side of Figure 12. We provide no initial guess for pressure, which explains the mismatch in the first snapshots. Notice then how the process to match up those subdomains could reduce discrepancies in just a few iterations. As for stopping criterion, we require that the residual in the tractions in the interface lies below a given tolerance. For this problem, we spent sixteen iterations to achieve a traction residual smaller than  $10^{-4}$ .

The DN final solution that we obtained after merging the subdomain results reproduces the analytical solution as well but exhibits larger errors compared to the MFEM, see Figure 15. Indeed, the error triplicates in particular in the bottom interface among subdomains. Part of the issue may be the fact that we are partitioning in the area where the pressure experiences strong gradients, as smarter partitioning strategy might lead to better results but with a detrimental in balance loading.

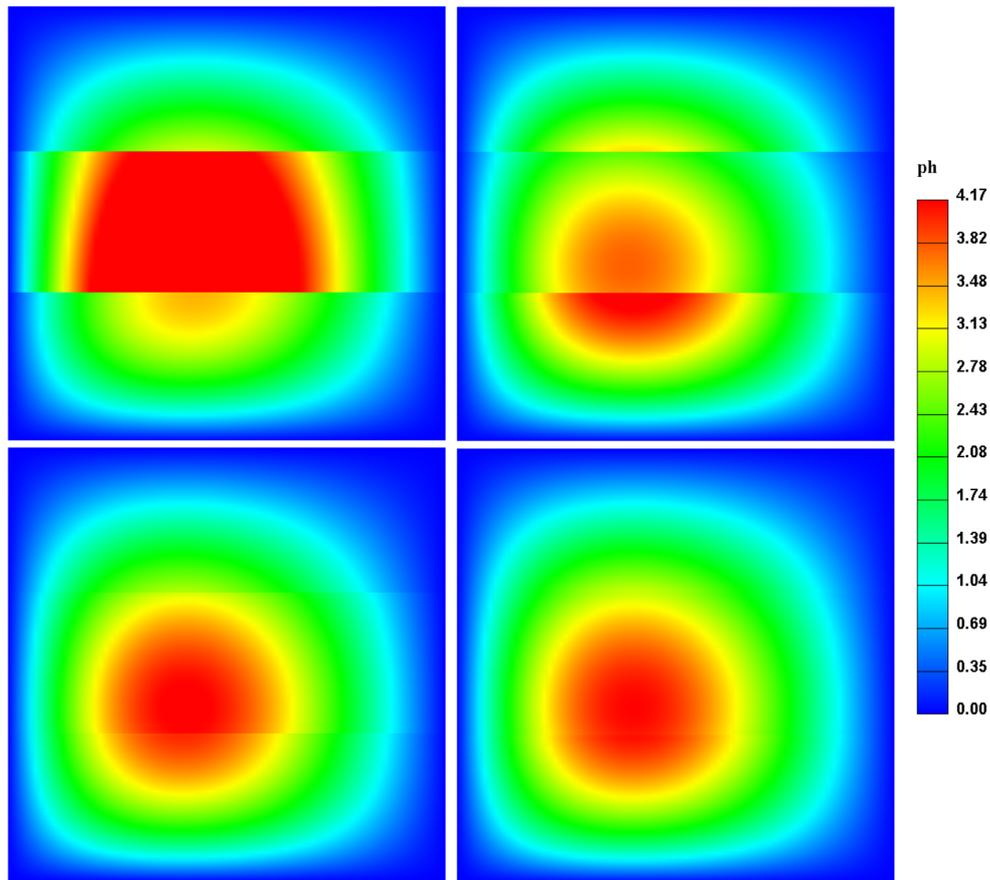


Figure 14. DN snapshots to solving (25) with pressure field (27).

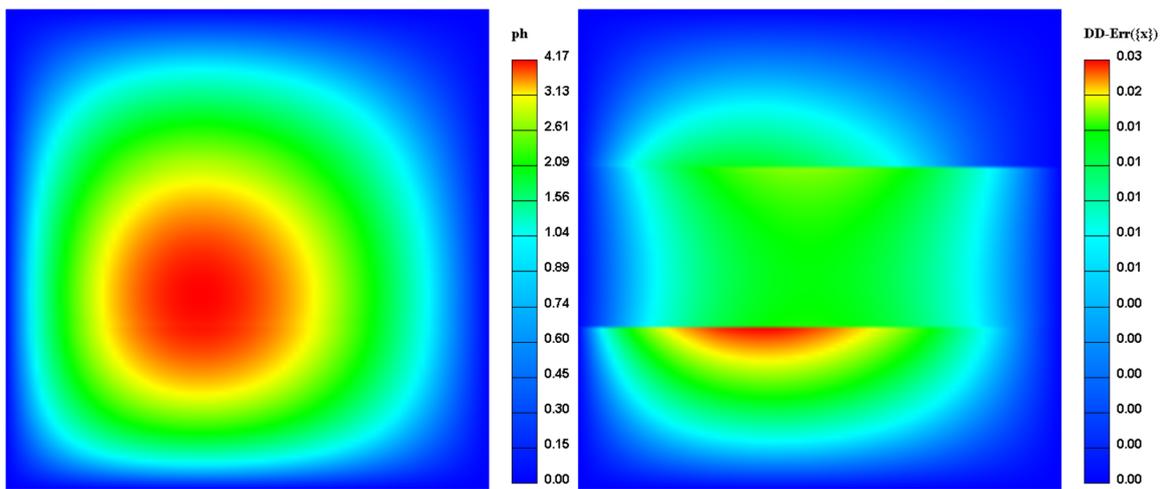


Figure 15. DN solution to problem (25) with pressure field (27) (left) and its error (right).

Finally, Table 3 compares timing data for both schemes. We employ multi-threading assembling for all subdomains. MFEM duplicates DN preprocessing time since we need to compute the mortar projectors and still need communication tables. However, since we employ the SPP approach for MFEM, we do not need to match subdomains iteratively as DN does, which explains the time savings. The DN meshes were finer than MFEM ones, and thus it takes longer to both assemble and solve those subdomains problems. The next example will cover on detail the overhead associated with DN. There is a clear winner for this benchmark problem, but results are relative, we repeat that these are complementary methods and not competitive.

Table 3. MFEM versus DNDDM.

Scheme	Prep. Time	Solving Runtime
MFEM	6 s, 964 ms	12 s, 475 ms
DNDDM	3 s, 572 ms	26 s, 463 ms

5.2. Example 2: 3-D Near Wellbore Problem

We revisit herein the well-known Bradley’s problem that we discussed already in [1] (see Figure 16). We consider homogeneous linear isotropic elasticity. Figure 16 depicts the problem geometry and BCS for a 2-D case where we prescribed pressure in the borehole, symmetrical displacements becomes zero on the bottom and left surfaces while we enforce the in situ or remote stresses in the remaining surfaces, as depicted in the figure. The hole’s radius is 5.0 inches, and the square length is 25.0 inches. Table 4 below shows the mechanical properties, pressure unit is MPa, and values for BCS where we divide the parameters by the mean remote stress, i.e.,  $\sigma = (\sigma_H + \sigma_h)/2.0$ ,  $(x)^* = (x/\sigma)$ , to normalize them. We extend Bradley’s problem to 3-D as proceeds. The geometrical entity encompasses a cube at the origin with a side of 25.0 inches, and we subtract an aligned cylinder whose radius is 5.0 inches as pictured in Figure 17. Additionally to the 2-D conditions above, we axially enforce, i.e.,  $z$ -direction, no  $u_z$  displacement in the bottom face while the top surface is traction-free, as displayed in the figure.

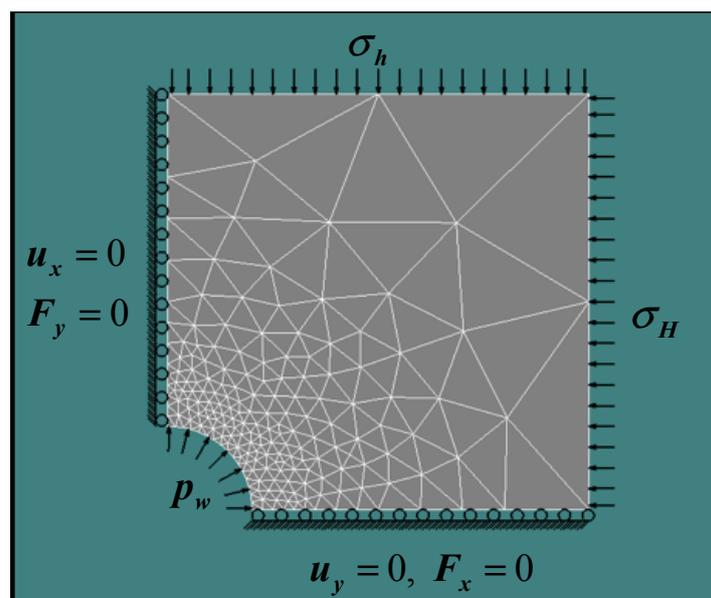


Figure 16. Bradley’s 2-D geometry, BCS, and a sample mesh.

Table 4. Parameters for Bradley’s problem.

Variable	Value
$p_w$	50.0
$\sigma_H$	80.0
$\sigma_h$	20.0
$E$	$5.0 \times 10^3$
$\nu$	0.25

Our objectives with this example are twofold. First, we compare the global serial solution with its parallel counterpart for a conforming mesh case. Second, we employ MFEM to glue nonmatching interfaces that discretize the problem accordingly. We compare the runtimes of the above procedures. We first tackle Bradley’s problem serially with a global conforming mesh that has 98,304 elements and

105,105 points. Assembling the stiffness matrix takes 3 min, and 14 s and solving the sparse system last 39.5 s. We use ILU as preconditioner and CG as an iterative solver which spent 283 iterations to achieve a tolerance of  $9.315 \times 10^{-9}$ . Table 5 summarizes the overall runtime for getting a solution. The global solution renders the displacements to lie within the following intervals:  $u_x = [-0.42786, 0]$ ,  $u_y = [-0.01154, 0.103155]$ , and  $u_z = [0, 0.188838]$ . We now employ the DNDDM over a pair of processors as depicted in Figure 18, which also shows on the right-hand-side the resulting parallel solution obtained after executing ten internal iterations to attain traction residual of  $4.8 \times 10^{-4}$ . We number the conforming subdomains starting at the bottom, Table 5 compiles timing and partitioning data.

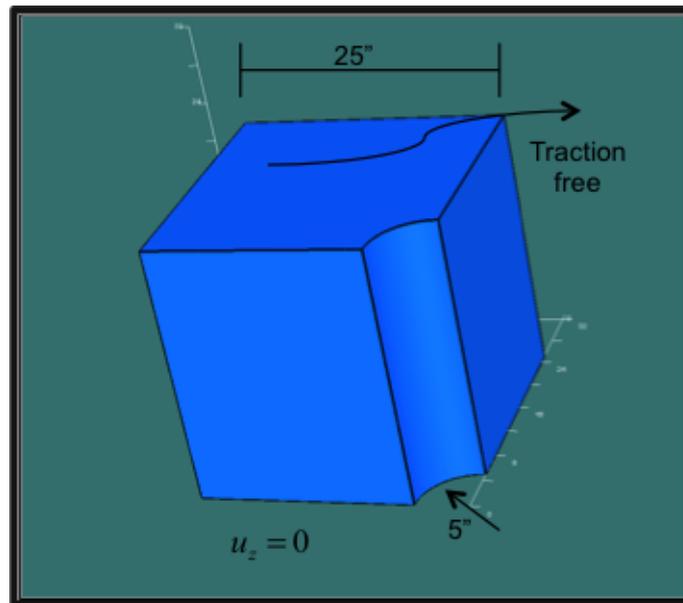


Figure 17. Bradley’s 3-D problem.

Table 5. Bradley’s problem partitioning and parallel solution.

Subdomain	# Elements	Type	Runtime
-	98304	Global	4 min, 48 s
0	52224	Neumann	9 min, 4 s
1	46080	Dirichlet	6 min, 20 s

Snaps in Figure 19 show the DN matching process evolution, where we illustrate the  $u_x$  displacement. We show the first four iterations in reading order. This parallel answer matches its serial counterpart which we gauged against commercial software, and we obtained excellent accordance [2] (see Figure 18 for instance). Indeed, for example, the relative error in the  $u_x$  displacement against the global solution is less than 0.14%. For the other two displacements, namely,  $u_y$  and  $u_z$ , the errors are 0.19% and 1% respectively. We can claim that the parallel solution reproduces the global solution. However, its runtimes are disappointing. Table 5 highlights the fact that the parallel solution took more than twice the time spent by the global one. The Neumann subdomain contributed the longer runtime. Besides we solve smaller subdomain problems, i.e., the sparse systems are about a half in rank, we must match the subdomain which takes up to 10 iterations. We solve the smaller matrixes that many times. Also, for the Neumann’s subdomain, we compute another solution by switching the boundary conditions in the interface to Dirichlet type, so that we can overrelax the tractions as indicated in Equation (24). Bottom line, Neumann’s subdomain acts as preconditioner and thus requires two local solves per iteration. This overhead explains the additional runtime. We must point out that partitioning and iterative coupling by the BCS makes plausible to solve huge problems that we cannot tackle in a single computer, though.

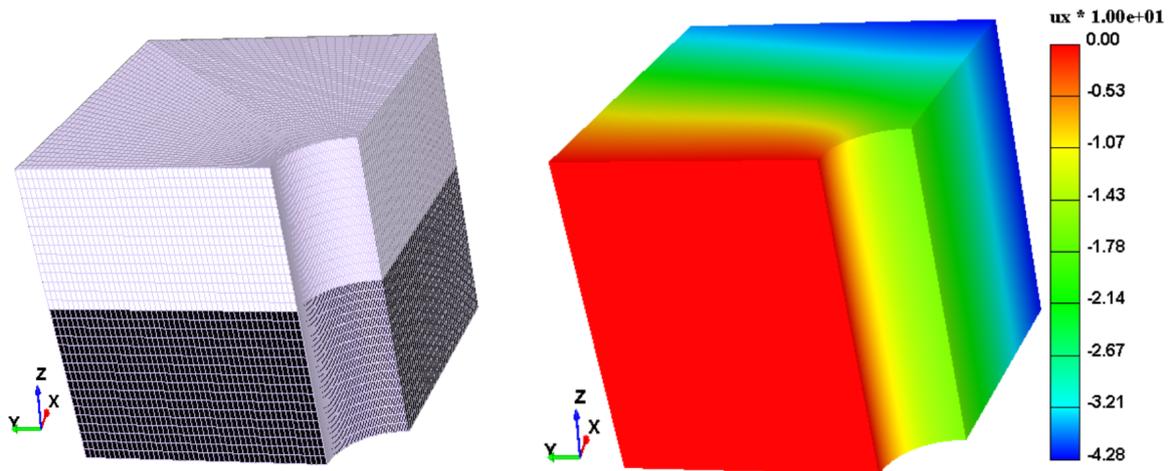


Figure 18. Partitioning (left) and parallel (right) answer for Bradley 3-D example, ( $u_x$  (in)).

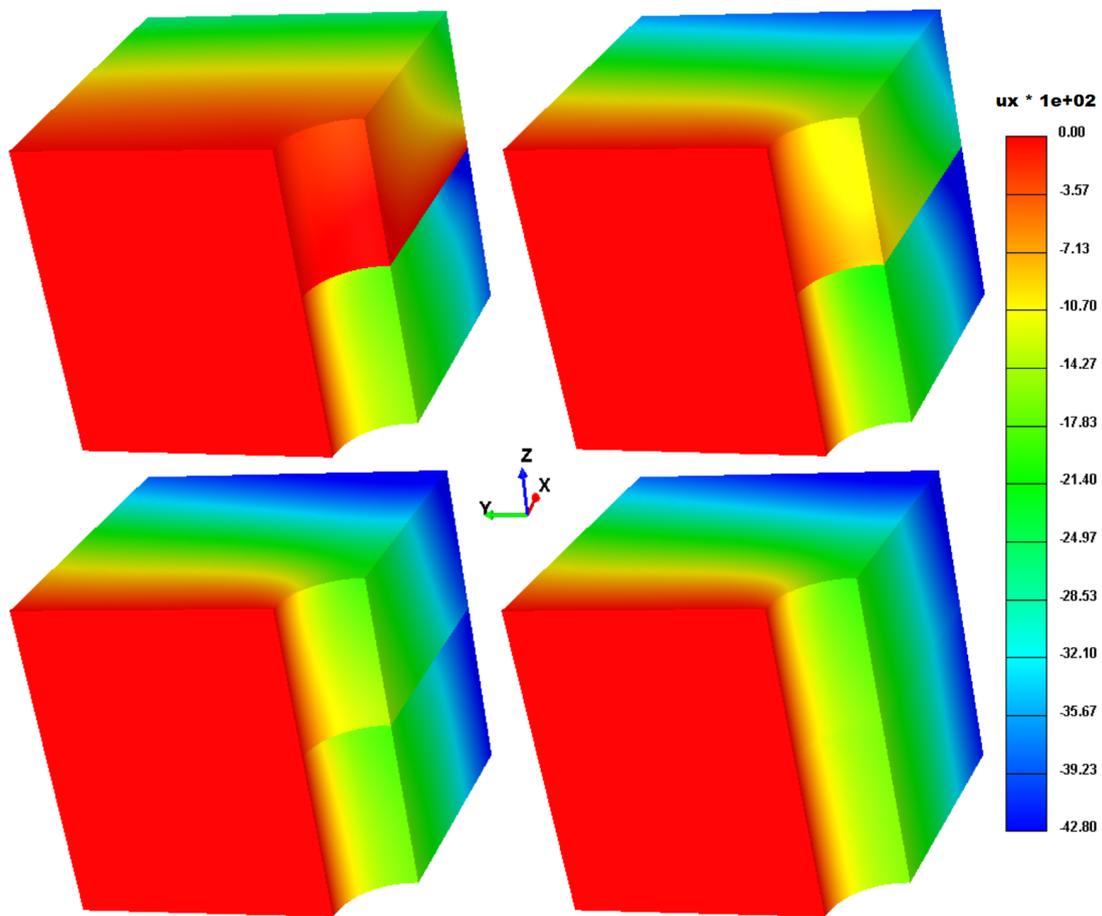


Figure 19. DN snapshots for  $u_x$  (in).

We undertake Bradley’s 3-D problem with non-conforming grids as follows. Hence, we radially divided the domain into two subdomains as shown in Figure 20. The inner subdomain has a finer mesh than the outer one. Both are tensor-product hexahedral meshes. Table 6 summarizes information concerning these subdomain. In the table, the mesh’s size refers to the tangential times radial times the axial number of points respectively. We consider two successively refined meshes for MFEM purposes via SPP. Figure 20 depicts the case with the finer meshes. The table also includes the number of elements and points on those subdomains as well as the runtime to assemble the local stiffness matrixes.

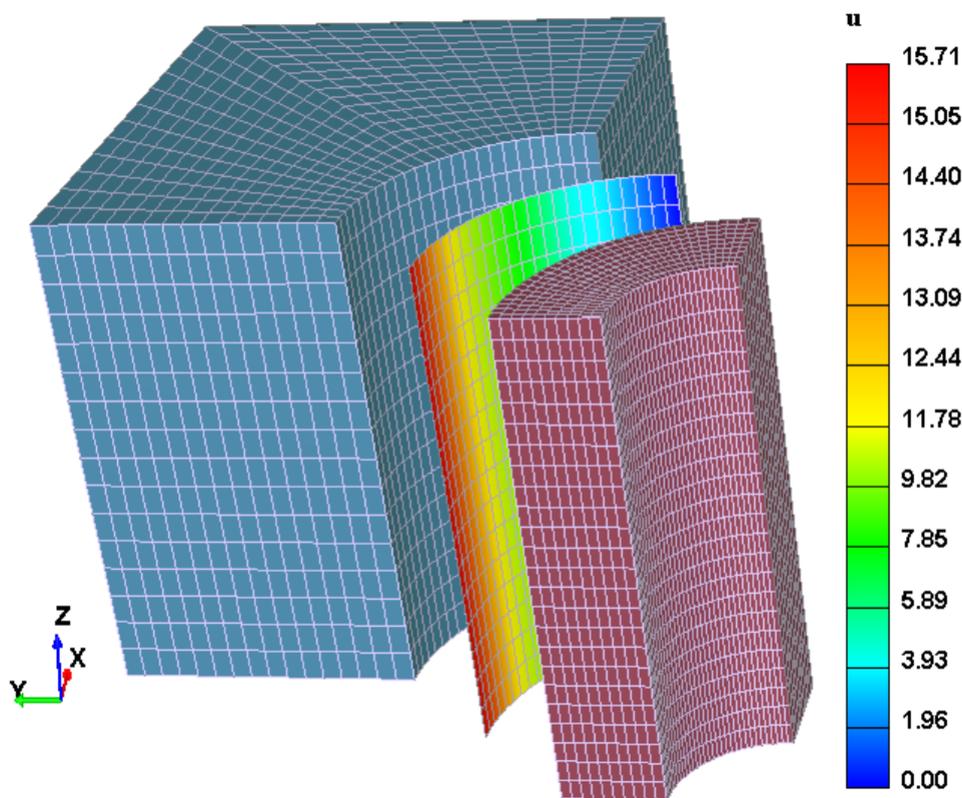


Figure 20. Meshes for MFEM.

Table 6. Meshes employed with MFEM.

Subdomain	Mesh's Size	Elements	Points	Assembling Time
Case 1: coarser meshes				
Inner	$25 \times 7 \times 17$	2304	2975	4 s, 484 ms
Outer	$17 \times 13 \times 13$	2304	2873	4 s, 453 ms
Case 2: finer meshes				
Inner	$33 \times 11 \times 25$	7680	9075	10 s, 781 ms
Outer	$25 \times 17 \times 17$	6144	7225	8 s, 515 ms

We geometrically describe the mortar plane by using a NURBS circular surface as pictured in Figure 20. The pseudo-color map presents the  $u \equiv \zeta$  variable of the entity [2,41]. The mortar grid resembles a regular Cartesian mesh of size  $25 \times 17$ . We consider piecewise linear rectangular elements.

Table 7 presents relevant information concerning these MFEM computations. We include details such as the preprocessing time, i.e., computing the projectors and preparing BCS, the time that the direct solver employed to solve the SPP, and also the overall runtime for MFEM. We observe that Case 1 attains proportional  $u_z$ -error to the DN case but with a substantial speedup. If we refine the subdomain further, as Case 2, we reduce the error in half, i.e., commensurable to DN, with a coarser mesh in the outer ring, but runtime is similar to the global case. Indeed, the SPP becomes highly ill-conditioned, and the direct frontal solver [57] struggles to solve it. However, the fact of gluing these nonconforming interfaces is what makes MFEM attractive. It is also possible to improve the direct solver performance by using dual mortar spaces that yield a diagonal projector or to decouple the SPP via DD-schemes. Wohlmuth [38] insists that her procedure is superior because the pairing condition is simpler to accomplish because fewer elements are in the compact-support of the basis functions. In such a scheme, we can represent the mortar projector by a diagonal matrix [37]. Finally, Figure 21 depicts these MFEM solutions that look relatively smooth.

Table 7. MFEM solutions via SPP.

Case	Prep. Time	SPP Time	Overall Runtime	$u_z$ Errors (%)
1	2 s, 328 ms	15 s, 968 ms	22.7 s	2.02
2	2 s, 93 ms	4 min, 4 s, 500 ms	4 min, 18.3 s	0.89

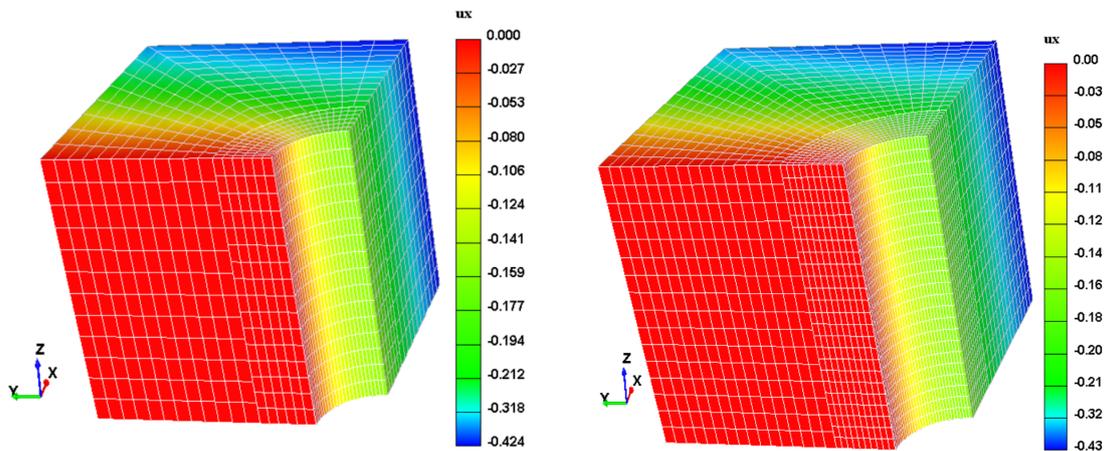


Figure 21. MFEM solutions for Bradley 3-D.

5.3. Example 3: Synthetic 3-D Reservoir

This example encompasses a realistic reservoir geometry including a pay-zone of size  $(20 \times 20 \times 10)$  and its surroundings as depicted in Figure 22. We enforce far-field BCS to all cuboid faces, i.e., no displacement in the orthogonal direction to the given face, except in the top one, which is traction-free. For the sake of simplicity, we assume a constant pressure drop of 1000.0 Psi in the pay-zone, which resembles a reservoir compaction and subsidence scenario [58,59]. The global conforming mesh, see Figure 22, consists of 13,824 hexahedral elements. We consider isotropic linear elasticity with constant mechanical properties, for both the reservoir and its vicinity thus  $E = 30.0$  Ksi and  $\nu = 0.30$ .

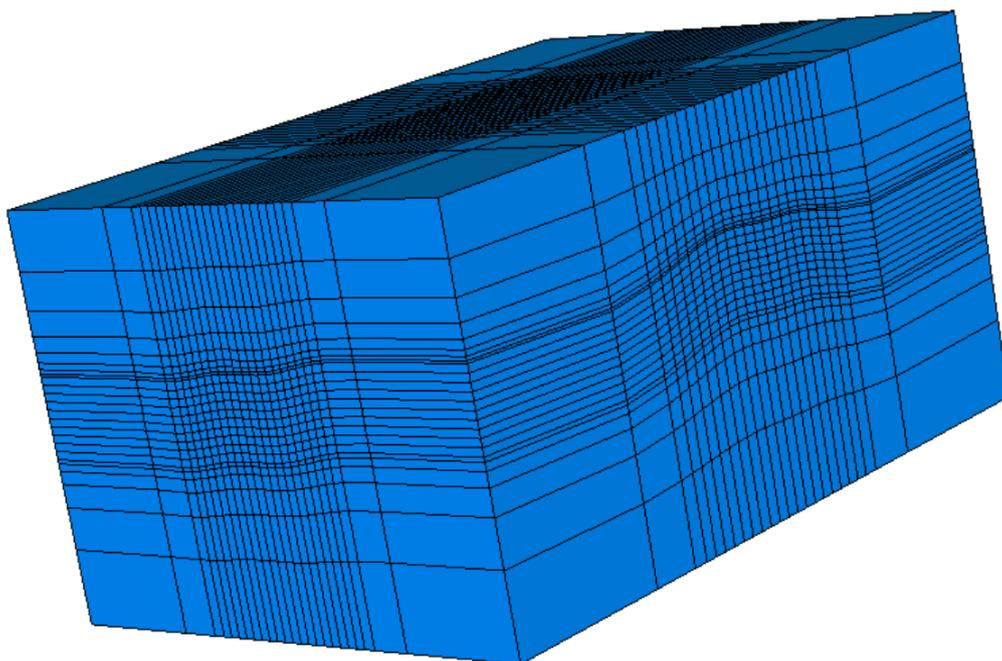


Figure 22. Conforming mesh.

Figure 23 depicts half of the domain as cut-away to show the resulting vertical displacement field. The deformation pattern is the standard blue compaction dome in the cap and the red build up in the base. The displacement is not symmetric, though. We employ this global conforming solution to benchmark our MFEM procedure.

We now partition the conforming case in the fashion that Figure 24 depicts. We keep the same discretization in the pay-zone level, but we have aeriually coarser overburden ( $12 \times 12 \times 8$ ) and underburden ( $12 \times 12 \times 6$ ). The mesh was attracted towards the pay-zone as usual. Since this is a graded mesh, we also have to grade the mortar space mesh, in the parametric plane, in a similar manner as Figure 25 depicts. Table 8 compiles information concerning these subdomains. We geometrically represent the mortar spaces using NURBS surfaces as in the earlier example.

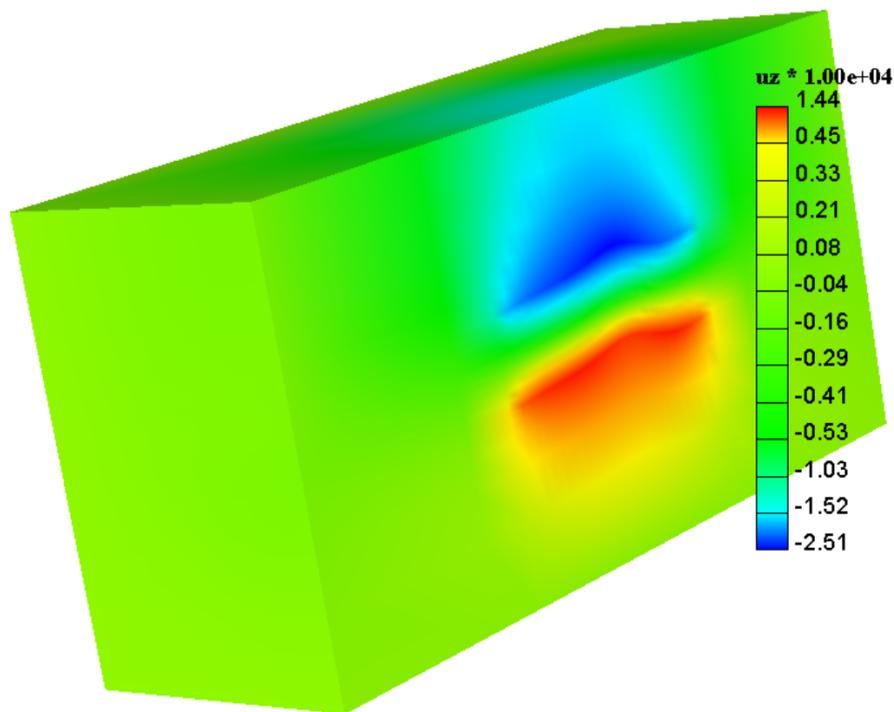


Figure 23. Solution  $u_z$  displacement.

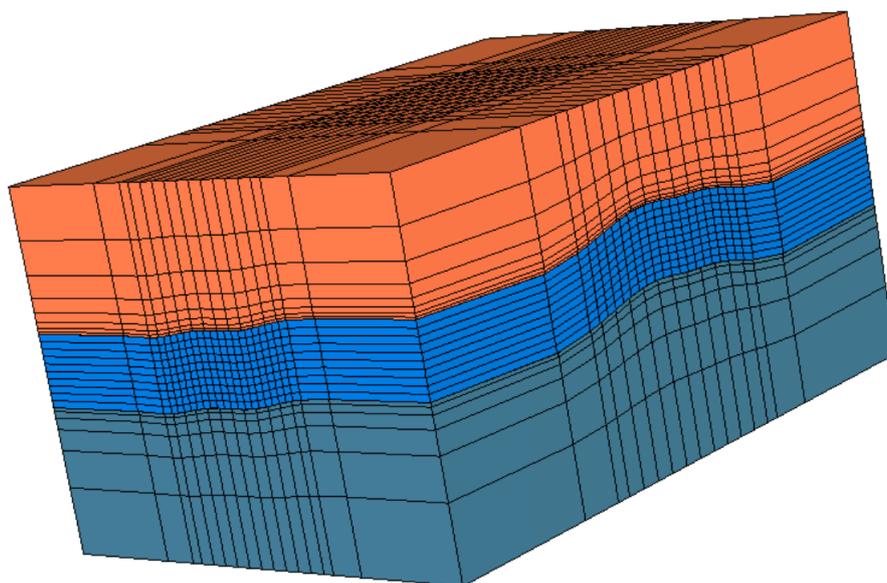


Figure 24. Meshes to use with MFEM.

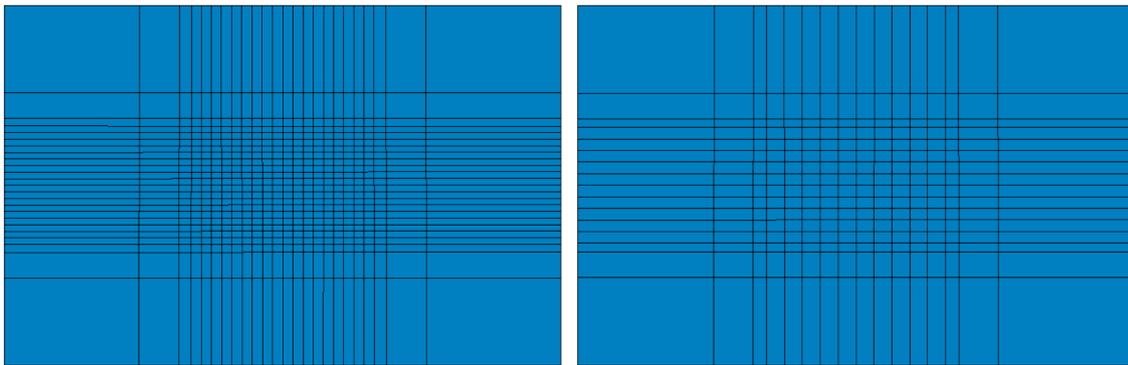


Figure 25. Mortar (left) and non-mortar (right) meshes.

Table 8. Grids for Example 3 in Section 5.3.

Level	# Elements	# Points	Coloring
Overburden	2048	2601	Dirichlet
Pay-zone	5760	5760	Neumann
Underburden	1536	2023	Dirichlet

The mortar space considers a graded tensor-product grid of  $25 \times 25$  linear piecewise rectangular elements. Figure 26 presents the MFEM answer via SPP. This solution is a suitable approximation to the conforming one, notice that the maximum compaction is within 6% while the build-up has a 7% relative error. These results are striking besides the mesh in the overburden and underburden have four times fewer elements than their equivalents in the conforming case.

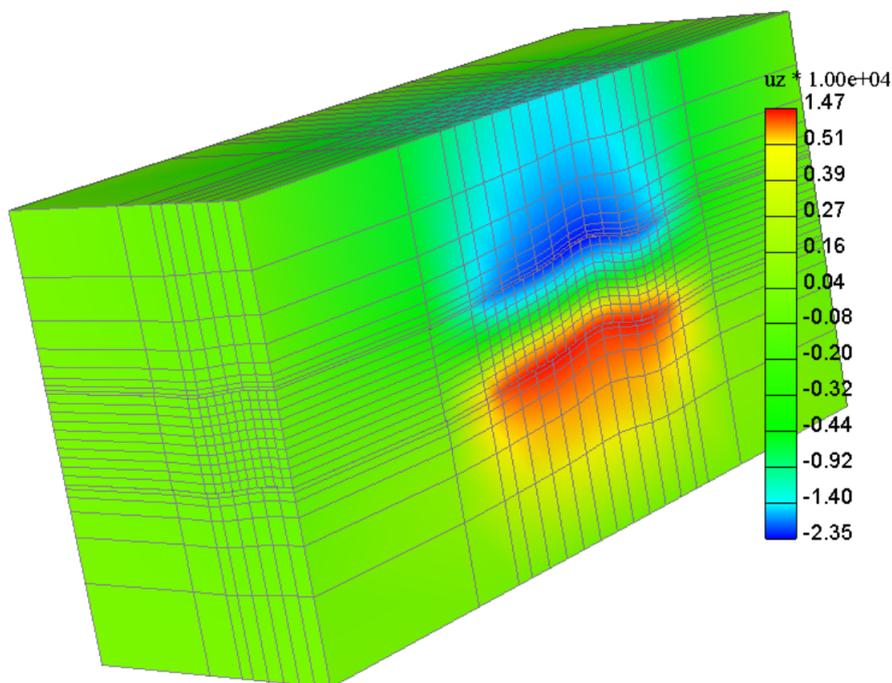


Figure 26. MFEM solution for Example 3 in Section 5.3.

Alternatively, successive snapshots, in reading order, in Figure 27 show the evolution of the DN pairing workflow,  $u_z$  is the displacement depicted; this decoupled solution agrees very well with both the serial and the one based on SPP. Figure 27 depicts the answer achieved after completing 12 iterations to attain traction residual less than  $10^{-4}$ . This last snapshot looks relatively smooth.

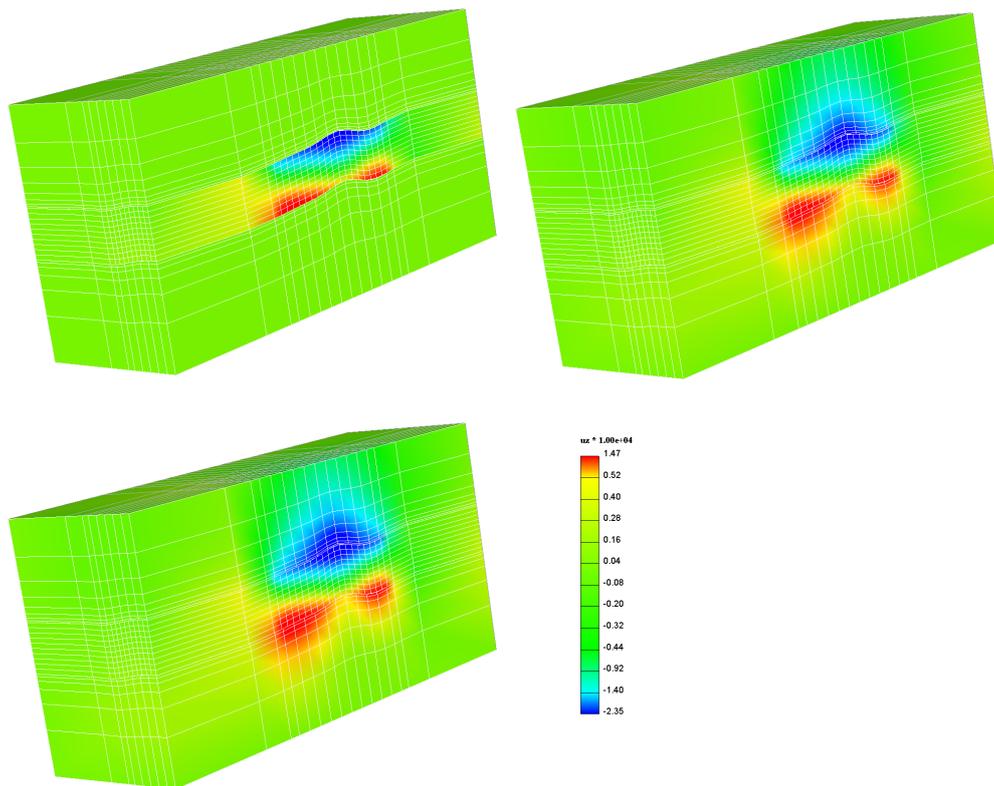


Figure 27. DN solution for Example 3 in Section 5.3.

5.4. Example 4: Realistic Synthetic Reservoir

We now modify the third example to tackle a more realistic reservoir geometry as shown in Figure 28. We employ the loosely coupled formulation (16). The isotropic porosity and permeability fields arise from the SPE10 related project (see Figure 29). The fluid viscosity equals 0.0133 cp while the entire compressibility is  $1.4 \times 10^{-5} \text{ Psi}^{-1}$ . We consider no gravity for both flow and elasticity equations. We retain the same mesh sizes, mechanical properties, and BCS as the example before.

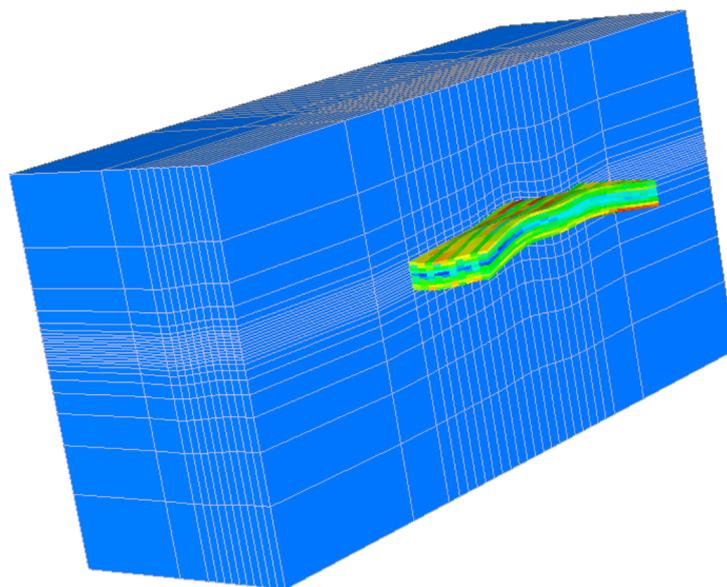


Figure 28. Conforming mesh for mechanics and the pay-zone’s mesh.

We consider two vertical producer wells in the corners, with prescribed bottom hole pressure, as depicted in Figure 30. The initial condition comprises a uniform pressure of 1000.0 Psi in the entire pay-zone while we assume  $p$  in the wells as 0.0 Psi that resembles a depletion situation. We also consider no-flow BCS on all reservoir surfaces. The initial displacement field is zero as well. Figure 30 depicts, in reading order, pressure evolution snapshots after 0, 20, and 40 years of production. We analyze these last two snapshots for MFEM purposes.

Figure 31 depicts pictures with the unfolding of the vertical displacement, for the conforming mesh. A compaction dome develops just above the section where the most significant pressure drop occurs. The deformation-pattern is again the usual one where a compaction skull sits on the head while a build-up happens in the base of the reservoir.

We repeat the same numerical experiment, this time by solving the SPP, and we observe that the resulting MFEM solutions once again accord very well with the conforming case as depicted in Figure 32. The discrepancies between the conforming and MFEM solutions are within 6%.

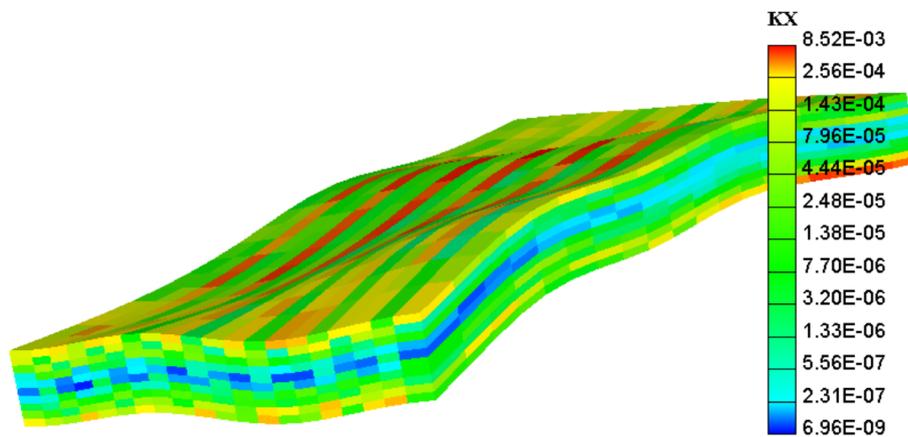


Figure 29. KX (mD) permeability in the pay-zone.

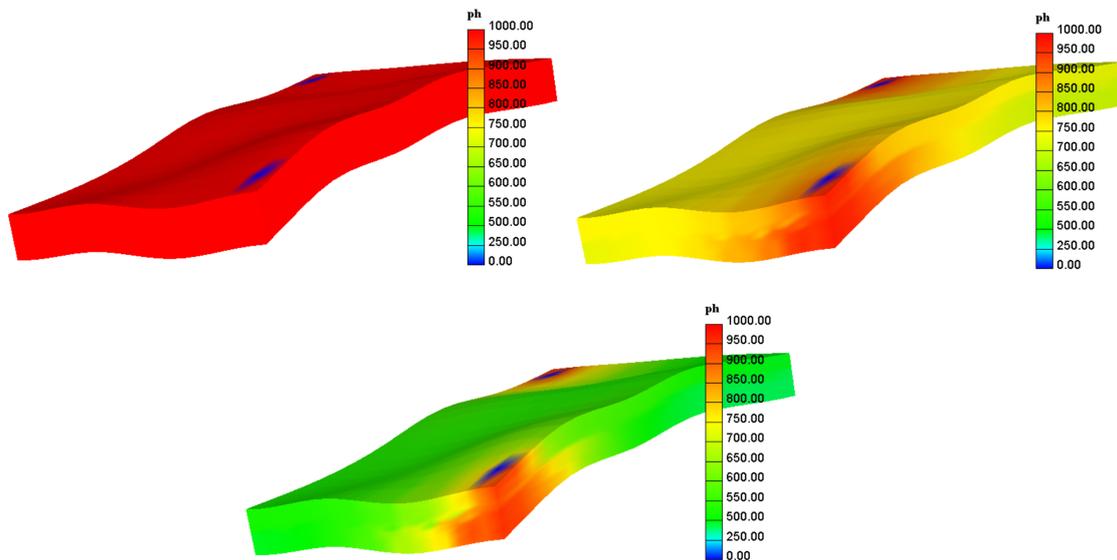


Figure 30. Pressure (Psi) snapshots at 0, 20, and 40 years.

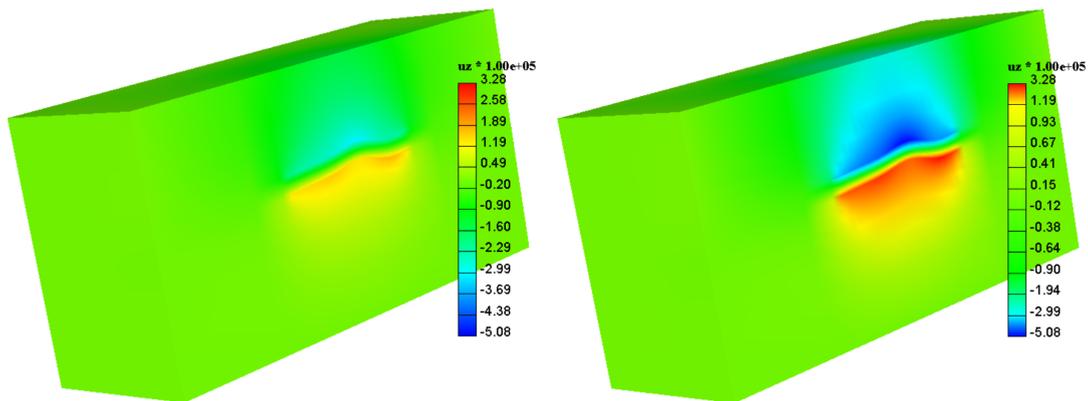


Figure 31.  $u_z$  snapshots at 20 and 40 years of evolution for conforming case.

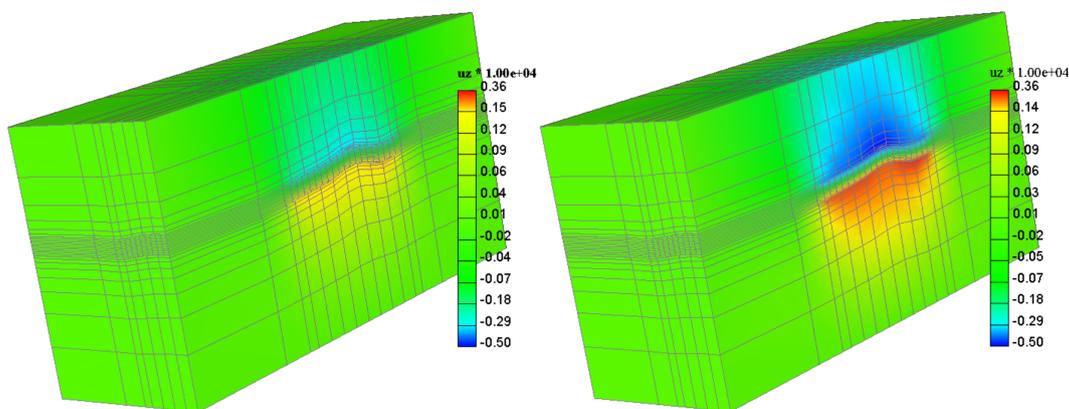


Figure 32.  $u_z$  snapshots at 20 and 40 years of evolution for MFEM.

## 6. Concluding Remarks

We introduced herein a comparably comprehensive MFEM scheme on curved interfaces described using NURBS surfaces. We explained in detail the steps that are necessary to compute the mortar projector, and we also included their algorithms in pseudo-code. The examples considered a steady-state single-phase flow problem, Bradley’s near-borehole problem, and two coupled flow and mechanics simulations with a realistic reservoir geometry. These last two field level compaction computations showed that the suggested method has the potential to tackle problems of practical interest.

We benchmarked these schemes for a topologically equivalent 2-D problem and we find out about the pros and cons of every approach. MFEM offers the flexibility of gluing nonmatching interfaces while DN operates over conforming ones. DN must match the subdomains in an iterative fashion which consumes CPU. Regarding runtime, MFEM certainly speeds up a given problem where we can glue coarser meshes in those areas with smaller gradients. On the other hand, MFEM via SPP requires solving an ill-condition problem that may not be tractable in challenging 3-D problems. We thus emphasize that these schemes are rather complementary than competitive.

We demonstrated that the DNDDM could decouple the global SPP that is imperative to continue this methodology to 3-D domains with a substantial number of DOF, where solving the latter is not feasible at all. We found that often parallel solution runtimes are disappointing when compared with the serial ones. Indeed, the Neumann subdomain contributed the longer runtime since we compute another solution by switching the BCS in the interface to Dirichlet type so that we can overrelax the tractions. Bottom line, Neumann’s subdomain acts as preconditioner and thus requires two local solves per iteration. We must point out that partitioning and iterative coupling by the BCS makes plausible to solve huge problems that we cannot tackle in a single computer, though. This study reveals new prospects to advance parallel codes for reservoir simulation linked with poroelasticity.

## 7. Path Forward

1. Hook up IPFA to given black oil and compositional reservoir simulators by using loosely and iterative couplings.
2. We need further testing on Linux cluster machines. IPFA was coded and examined already on MS-Windows for the sake of directness as well as the existence of free-of-charge debugging software. Our expectation remains compiling, linking and running only on different operating systems to release in a cross-platform fashion.
3. We will extend the scheme presented herein to nonlinear problems such as those arising from thermal poroplasticity. We also plan to reduce the DN solving time per iteration employing model-order reduction ideas, e.g., creating an oblique subspace via proper orthogonal decomposition to speed up solving the subdomain problems.

**Funding:** This research received no external funding.

**Acknowledgments:** The author acknowledges “GeoFeMOR, LLC” for partially funding this research, allowing access to IPFA and LogProc and permitting to publish these results. The author thanks the anonymous reviewers for their suggestions to improve this manuscript.

**Conflicts of Interest:** The author declares no conflicts of interest.

## References

1. Florez, H.; Wheeler, M. A mortar method based on NURBS for curved interfaces. *Comput. Methods Appl. Mech. Eng.* **2016**, *310*, 535–566. [[CrossRef](#)]
2. Florez, H. Domain Decomposition Methods for Geomechanics. Ph.D. Thesis, The University of Texas, Austin, TX, USA, 2012.
3. Florez, H.; Wheeler, M.; Rodriguez, A. A mortar method based on NURBS for curved interfaces. In Proceedings of the 46th US Rock Mechanics/Geomechanics Symposium, Chicago, IL, USA, 24–27 June 2012.
4. Florez, H.; Wheeler, M.; Rodriguez, A. A mortar method based on NURBS for curved interfaces. In Proceedings of the 13th European Conference on the Mathematics of Oil Recovery (ECMOR XIII), Biarritz, France, 10–13 September 2012.
5. Florez, H.; Wheeler, M.; Rodriguez, A.; Monteagudo, J. Domain Decomposition Methods Applied to Coupled Flow-Geomechanics Reservoir Simulation. In Proceedings of the SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 21–23 February 2011.
6. Dean, R.; Gai, X.; Stone, C.; Minkoff, S. A comparison of techniques for coupling porous flow and geomechanics. In Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, USA, 3–5 February 2003.
7. Pao, W.; Lewis, R.; Masters, I. A fully coupled hydro-thermo-poro-mechanical model for black oil reservoir simulation. *Int. J. Numer. Anal. Methods Geomech.* **2001**, *25*, 1229–1256. [[CrossRef](#)]
8. Gutierrez, M.; Lewis, R. The role of geomechanics in reservoir simulation. In Proceedings of the SPE/ISRM Rock Mechanics in Petroleum Engineering, Trondheim, Norway, 8–10 July 1998.
9. Longuemare, P.; Mainguy, M.; Lemonnier, P.; Onaisi, A.; Gérard, C.; Koutsabeloulis, N. Geomechanics in reservoir simulation: Overview of coupling methods and field case study. *Oil Gas Sci. Technol. Rev.* **2002**, *57*, 471–483. [[CrossRef](#)]
10. Yin, S.; Dusseault, M.B.; Rothenburg, L. Thermal reservoir modeling in petroleum Geomechanics. *Int. J. Numer. Anal. Methods Geomech.* **2009**, *33*, 449–485. [[CrossRef](#)]
11. Ferronato, M.; Janna, C.; Gambolati, G. Mixed constraint preconditioning in computational contact mechanics. *Comput. Methods Appl. Mech. Eng.* **2008**, *197*, 3922–3931. [[CrossRef](#)]
12. Rice, J.; Cleary, M. Some basic stress-diffusion solutions for fluid saturated elastic porous media with compressible constituents. *Rev. Geophys. Space Phys.* **1976**, *14*, 227–241. [[CrossRef](#)]
13. Biot, M. General theory of three-dimensional consolidation. *J. Appl. Phys.* **1941**, *2*, 155–164. [[CrossRef](#)]
14. Liu, R. Discontinuous Galerkin Finite Element Solution for Poromechanics. Ph.D. Thesis, The University of Texas, Austin, TX, USA, 2004.
15. Coussy, O. *Poromechanics*; Wiley: New York, NY, USA, 2004.

16. Lewis, R.; Schrefler, B. *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*, 2nd ed.; John Wiley and Sons: New York, NY, USA, 1998.
17. Phillips, P.; Wheeler, M. A coupling of mixed and continuous Galerkin finite element methods for poroelasticity I: The continuous in time case. *Comput. Geosci.* **2007**, *11*, 131–144. [[CrossRef](#)]
18. Phillips, P.; Wheeler, M. A coupling of mixed and continuous Galerkin finite element methods for poroelasticity II: The discrete-in-time case. *Comput. Geosci.* **2007**, *11*, 145–158. [[CrossRef](#)]
19. Sandhu, R.; Wilson, E. Finite element analysis of seepage in elastic media. *J. Eng. Mech. Div. ASCE* **1969**, *95*, 641–652.
20. Muller, A.; do Amaral Vargas, E., Jr.; Vaz, L.; Goncalves, C.J. Borehole stability analysis considering spatial variability and poroelastoplasticity. *Int. J. Rock Mech. Min. Sci.* **2009**, *46*, 90–96. [[CrossRef](#)]
21. Ghaboussi, J.; Wilson, E. Flow of compressible fluid in porous elastic solids. *Int. J. Numer. Methods Eng.* **1973**, *5*, 419–442. [[CrossRef](#)]
22. Kim, J.; Tchelepi, H.; Juanes, R. Stability, Accuracy and Efficiency of Sequential Methods for Coupled Flow and Geomechanics. In Proceedings of the 2009 SPE Reservoir Simulation Symposium, SPE, The Woodlands, TX, USA, 1 January 2009.
23. Zienkiewicz, O.; Humpheson, C.; Lewis, R.W. A unified approach to soil mechanics problems including plasticity and visco-plasticity. In *Finite Elements in Geomechanics*; Gudehus, G., Ed.; Wiley: New York, NY, USA, 1977; pp. 151–177.
24. Booker, J.; Small, J. An investigation of the stability of numerical solution of Biot's equations of consolidation. *Int. J. Solids Struct.* **1975**, *2*, 907–917. [[CrossRef](#)]
25. Zienkiewicz, O.; Shiomi, T. Dynamic behavior of saturated porous media: The generalized Biot formulation and its numerical solution. *Int. J. Numer. Anal. Methods Geomech.* **1984**, *8*, 71–96. [[CrossRef](#)]
26. De Souza Neto, E.; Peric, D.; Owen, D. *Computational Methods for Plasticity: Theory and Applications*; John Wiley and Sons Ltd.: Chichester, UK, 2008.
27. Wheeler, M. An Elliptic Collocation-Finite Element Method with Interior Penalties. *SIAM J. Numer. Anal.* **1978**, *15*, 152–161. [[CrossRef](#)]
28. Chin, L.; Raghavan, R.; Thomas, L. Fully-coupled geomechanics and fluid-flow analysis of wells with stress-dependent permeability. In Proceedings of the SPE International Conference and Exhibition, Beijing, China, 2–6 November 1998.
29. Settari, A.; Walters, D.A. Advanced in coupled geomechanics and reservoir modeling with applications to reservoir compaction. In Proceedings of the SPE Reservoir Simulation Symposium, Houston, TX, USA, 3–6 October 1999.
30. Shao, J. A numerical solution for a thermo-hydro-mechanical coupling problem with heat convection. *Int. J. Rock Mech. Min. Sci.* **1997**, *34*, 163–166. [[CrossRef](#)]
31. Belgacem, F. The mixed mortar finite element method for the incompressible Stokes problem: Convergence analysis. *SIAM J. Numer. Anal.* **2000**, *37*, 1085–1100. [[CrossRef](#)]
32. Bernardi, C.; Maday, Y.; Patera, A. A new nonconforming approach to domain decomposition: The mortar element method. In *Nonlinear Partial Differential Equations and their Applications*; Longman Sci. & Tech.: Paris, France, 1994; pp. 13–51.
33. Magoules, F.; Roux, F.; Series, L. Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 3742–3759. [[CrossRef](#)]
34. Hauret, P.; Ortiz, M. BV estimates for mortar methods in linear elasticity. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 4783–4793. [[CrossRef](#)]
35. Krause, R.; Wohlmuth, B. Nonconforming domain decomposition techniques for linear elasticity. *East-West J. Numer. Math.* **2000**, *8*, 177–206.
36. Puso, M. A 3D mortar method for solid mechanics. *Int. J. Numer. Methods Eng.* **2004**, *59*, 315–336. [[CrossRef](#)]
37. Flemisch, B.; Puso, M.A.; Wohlmuth, B.I. A new dual mortar method for curved interfaces: 2D elasticity. *Int. J. Numer. Methods Eng.* **2005**, *63*, 813–832. [[CrossRef](#)]
38. Wohlmuth, B. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM J. Numer. Anal.* **2000**, *38*, 989–1012. [[CrossRef](#)]
39. Fritz, A.; Heber, S.; Wohlmuth, B.I. A comparison of mortar and Nitsche techniques for linear elasticity. *Calcolo* **2004**, *41*, 115–137. [[CrossRef](#)]

40. Florez, H. Application of mortar methods to subsidence computations. In Proceedings of the 51st US Rock Mechanics/Geomechanics Symposium, San Francisco, CA, USA, 25–28 June 2017.
41. Florez, H.; Manzanilla-Morillo, R.; Florez, J.; Wheeler, M.F. Spline-based reservoir's geometry reconstruction and mesh generation for coupled flow and mechanics simulation. *Comput. Geosci.* **2014**, *18*, 949–967. [[CrossRef](#)]
42. Florez, H.; Ceberio, M. A Novel Mesh Generation Algorithm for Field-Level Coupled Flow and Geomechanics Simulations. In Proceedings of the 50th US Rock Mechanics/Geomechanics Symposium, Houston, TX, USA, 26–29 June 2016.
43. Gai, X. A Coupled Geomechanics and Reservoir Flow Model on Parallel Computers. Ph.D. Thesis, The University of Texas, Austin, TX, USA, 2004.
44. Phillips, P. Finite Element Methods in Linear Poroelasticity: Theoretical and Computational Results. Ph.D. Thesis, The University of Texas, Austin, TX, USA, 2005.
45. Becker, E.; Carey, G.; Oden, J. *Finite Elements: An Introduction, Vol. I of The Texas Finite Element Series*; Prentice-Hall Inc.: Englewood Cliffs, NJ, USA, 1981.
46. Oden, J.T.; Carey, G.F. *Finite Elements: Special Problems in Solid Mechanics, Vol. V of The Texas Finite Element Series*; Prentice-Hall Inc.: Englewood Cliffs, NJ, USA, 1984.
47. Sun, S.; Wheeler, M.F. Projections of velocity data for the compatibility with Transport. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 653–673. [[CrossRef](#)]
48. Girault, V.; Pencheva, G.V.; Wheeler, M.; Wildey, T. Domain decomposition for linear elasticity with DG jumps and mortars. *Comput. Methods Appl. Mech. Eng.* **2009**, *198*, 1751–1765. [[CrossRef](#)]
49. Badia, S.; Nobile, F.; Vergara, C. Fluid–structure partitioned procedures based on Robin transmission conditions. *J. Comput. Phys.* **2008**, *227*, 7027–7051. [[CrossRef](#)]
50. Maday, Y.; Magoules, F. Absorbing interface conditions for domain decomposition methods: A general presentation. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 3880–3900. [[CrossRef](#)]
51. Bjorstad, P.; Widlund, O. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* **1986**, *23*, 1093–1120. [[CrossRef](#)]
52. Marini, L.; Quarteroni, A. A relaxation procedure for domain decomposition methods using finite elements. *Numer. Math.* **1989**, *55*, 575–598. [[CrossRef](#)]
53. Funaro, D.; Quarteroni, A.; Zanolli, P. An Iterative Procedure with Interface Relaxation for Domain Decomposition Methods. *SIAM J. Numer. Anal.* **1988**, *25*, 1213–1236. [[CrossRef](#)]
54. Quarteroni, A.; Valli, A. *Domain Decomposition Methods for Partial Differential Equations*; Oxford University Press: Oxford, UK, 1999.
55. Toselli, A.; Widlund, O.B. Domain Decomposition Methods: Algorithms and Theory. In *Computational Mathematics*; Springer: Berlin, Germany, 2004.
56. Florez, H.; Borges, B. *LogProc's Technical Manual*; Version 1.2; GeoFeMOR, LLC.: Adelphi, MD, USA, 2018.
57. Davis, T. *UMFPACK User Guide*; Version 5.5.1.; University of Florida: Gainesville, FL, USA, 2011.
58. Charlez, P. *Rock Mechanics, Volume I: Theoretical Fundamentals*; Editions Technip: Paris, France, 1991.
59. Fjar, E.; Fjar, E.; Holt, R.M.; Raaen, A.M.; Risnes, R.; Horsrud, P. *Petroleum Related Rock Mechanics*, 2nd ed.; Elsevier: Budapest, Hungary, 2008.

