Response to Reviewer Comments 3

We thank the reviewer for evaluating our manuscript and for the constructive comments. Note that we use the abbreviation **RC3** for reviewer comments and **AR** for our authors' response in the following. Removed text is shown in red, e.g., ~~this text has been removed.~~. New text is shown in blue, e.g., this text has been added.

RC3: The manuscript describes a new contribution to the modelling of hydro-mechanical problems in form of a MOOSE-based simulator. While I think that it deserves publication in GMD, it needs substantial improvements as outlined below. This particularly concerns Section 3 which should be made much more detailed and explicit to be of any value. Moreover, the literature review on existing implementation efforts should be considerably improved. So far, it arbitrarily picks three open-source packages and discredits them with oversimplifying statements.

> **AR:** We apologize for our phrasing regarding the 3 open-source packages. We will expand our literature review of numerical codes, improve formulation about their capabilities and extend Section 3 to include more details.

RC3: - l.44ff: I don't agree with this statement. The mentioned references indicate that more sophisticated sequential schemes like iterated fixed-stress perform quite well. It's the "naive" or "straightforward" approaches like drained-split or non-iterated fixed-stress which might perform poorly. Please rephrase.

> **AR:** We agree and will revise accordingly:
>
> Line 44: "Although sequential codes allow flexible and efficient code management in conjunction with reasonable computational costs, they tend to perform poorly in tightly coupled processes, since transient interaction between variables may not be computed accurately (Kim et al., 2011). However, sequential coupling can significantly improve its numerical accuracy when is combined with iterative schemes. In such implementations, feedback between variables is perform by transferring flow variables to the geomechanics then inserting calculated stress and strain back into the flow problem for the next iteration (Beck et al., 2020). Stability of the iterative methods is discussed by Kim et al. (2011); Mikelić and Wheeler (2013).

RC3: - l.55ff: There's a more recent "official" paper on PorePy which is better to cite than the 2017 one: https://link.springer.com/article/10.1007/s10596-020-10002-5. I think that the statement "these codes are in an immature stage" doesn't properly reflect the apparently

rather big efforts behind, at least, PorePy. I'm also not sure about the argument with the "point and line sinks", maybe they have already been integrated or maybe it's very easy to integrate them as a user of the framework. To me, the intention of these codes is just very different from company efforts like COMSOL or coupling frameworks such as MOOSE and probably not what the authors need or aim for.

> **AR:** We agree and will revise this accordingly. In our opinion PorePy is a novel open source software for fast prototyping of modelling tools that can later on be applied to more general frameworks such as MOOSE or FEniCS, Dune, etc.
>
> Suggested revision at line 55: "The second is called Porepy and was specifically developed to simulate HM processes in rock fractures Keilegavlen et al. (2021). Despite the fact that python-based coding offers the advantage of high-level programming within a relatively friendly user interface, ~~in an immature stage and relevant subsurface boundary conditions such as point and line sinks have not yet been integrated.~~ these codes are design to facilitate rapid development of features that cannot be properly represented by standard simulation tools rather than general multiphysics problems.."

RC3: - l.60f: I also don't like the negative statements here or at least the general way in which they are formulated. I'm sure that the developers of OpenGeoSys would disagree. It's open source, anyone can adapt the "fundamental governing equations" to her needs. In principle. Please relate things with something like "from our experience, it's rather difficult to..."

> **AR:** Thank you for identifying our negativity! We agree and will revise as follow:
>
> Suggested revision at line 60: "The code is well documented and features several examples in different subsurface areas, ~~its fundamental governing equations are fixed and it lacks a flexible API to customise them. Furthermore, the authors are unaware of a peer-reviewed verification which includes full geomechanical heterogeneity~~ further new developers are constantly contributing with new features to the source code (Graupner et al., 2011; Kosakowski and Watanabe, 2014; Li et al., 2014). From our experience, however, users without some computer science background, experience programming in C++ or python and experience using a debugger, may require a significant amount of time to take full advantage of the features that OGS offers. Furthermore, the authors are unaware of a peer-reviewed verification which includes full geomechanical heterogeneity."

RC3: - The list of other mentioned packages is very short and subjective. There are many other efforts for hydro-mechanical modelling based on other frameworks, such as

* https://doi.org/10.1016/j.cam.2016.10.022 or https://doi.org/10.1142/S1793962321500033 based on deal.II

* https://doi.org/10.1007/s10596-020-09987-w based on Dumux

* https://doi.org/10.1029/2019JB017298 based on MRST

* PFLOTRAN can do geomechanics

All of these frameworks can deal with heterogeneous material parameters.

AR: Thank you for providing these references. While we agree that the the list of numerical codes could be made more exhaustive, none of the suggested codes can integrate spatially distributed heterogeneity of both hydraulic and geomechanical properties like RHEA does. To the best of our knowledge, all other codes can merely integrate spatial geomechanical heterogeneity in the form of layers or as pre-defining material blocks in the mesh which makes the workflow tedious. RHEA significantly simplifies this task by allowing the user to pre-define spatially distributed properties in the Python environment.

Suggested revision at line 44: Another massively parallel subsurface flow code PFLORTRAN, an open-source, multiscale and multiphysics code for subsurface and surface processes (Hammond et al., 2014). The code solves for non-isothermal multiphase flow, reactive transport and geomechanics in porous medium. PFLOTRAN has been applied to model hydro geomechanical systems (Lichtner and Karra, 2014).

Suggested revision at line 59: MRST is an open-source code developed for fast prototyping of new tool in reservoir modelling (Lie, 2019) coded on MATLAB. Although MRST is not a simulator itself, supports multyphase flow with THM physics. MRST has been used in hydro geomechanical in fracture rock in the past (Zhao and Jha, 2019) as well as several other subsurface applications (Garipov et al., 2018; Ahmed et al., 2017; Edwards et al., 2017)

Suggested revision at line 62: DuMux is a fully coupled numerical simulator for multi phase flow and transport in porous medium, free and open source (Flemisch et al., 2011). It is based on the Distributed Unified Numeric Environments (DUNE) which is a C++ based ecosystem to solve FEM based on PETSc solvers. DuMux is well known for its strong focus on multi-phase flow and transport in porous media, its recent realise adds extra features which facilitates physics coupling such as Navier-stokes models (Koch et al., 2021).

Suggested revision at line 67: Other novel codes include (Dang and Do, 2021; Tran and Jha, 2020; Reichenberger, 2003; Martin et al., 2005; Frih et al., 2012)

RC3: - l.70f: "for example an experienced user can easily modify the source code to add desired features such as..." I'm convinced that this also holds for the open-source codes which are mentioned above and which have been put in a negative light.

AR: We agree and will revise accordingly.

Suggested revision at line 67: We have found that mastering the basic concepts of the MOOSE workflow requires a steep learning curve. However, it requires minimum C++ coding skills which facilitates the learning experience from users that not necessarily have a computer science background. Once the basics are mastered the benefits are significant, for example an experienced user can easily modify the source code to add desired features such as multi-scale physics, non-linear material properties, complex boundary conditions or even basic post-processing tools with only a few lines of code.

RC3: - Section 3 does a very poor job in describing the implementation of the model, as it doesn't connect enough to Section 2. There's only a few lines 135ff which make an explicit connection, the rest is generic blabla. I consider this the most important section of the manuscript. Please be much more precise here. How are the two modules integrated? How are they coupled? What finite elements are used? What time integration scheme? What about local mass conservation?...

AR: We have added the following material to the end of Section 2, in an attempt to succinctly describe these features.
As derivative of the MOOSE framework, RHEA has access to a wide array of options for tuning a simulation. Solver options such as numerical schemes, adaptive timestepping, and PETSc options are available. By default, RHEA uses a first order fully-implicit time integration (backward Euler) for unconditional stability and solves the above equations simultaneously (full coupling) (Kavetski et al., 2002; Manzini and Ferraris, 2004; Gaston et al., 2009). RHEA also allow operator splitting to implement loose coupling (solving the fluid flow while keeping the mechanics fixed, then solving the mechanics while keeping the fluid-flow fixed) and the latter can even occur on separate meshes with different time-stepping schemes, but this feature is not explored in the current article (Martineau et al., 2020)

Explicit time integration (with full or loose coupling) and other schemes such as Runge-Kutta are available in MOOSE and RHEA, but stability limits the time-step size, so these are rarely used in the type of subsurface problems handled by RHEA. By default, MOOSE and RHEA use linear Lagrange finite elements (tetrahedra, hexahedra and prisms for 3D problems, triangles and quads for 2D problems), but higher-order elements may be easily chosen if desired (Hu, 2017)
RHEA does not implement any numerical stabilization for the fluid equation to eliminate overshoots and undershoots, however, fluid volume is conserved at the element level (Cacace and Jacquey, 2017). Although not explored in this article, RHEA's fluid flow may be extended to multi-phase, multi-component flow with high-precision equations of state, as well as finite-strain elasto-plasticity (Wilkins et al., 2020).

RC3: - l.349: I saw that you are required to link to Zenodo. That's ok, but please keep also the link to GitHub, that's where your code is developed further, hopefully.

AR: Done.

RC3: - l.67: What are "gold-standard numerical solvers"?

AR: We will explain this in our revised version.

Suggested revision at line 67: "It offers ~~gold-standard numerical solvers as well as a host of useful features such as adaptive meshing"~~ clean and effective numerical PDEs solvers as well as mesh capabilities with a uniform approach for each class of problem. This design enables easy comparison and use of different algorithms (for example, to experiment with different Krylov subspace methods, preconditioners, or truncated Newton methods) which are under constant development.

RC3:
l.32, 92, 102, ...: a single "porous mediUM", many porous media
- l.161: "designED"
- l.189: "undraiNed"
- (19): "L" instead of "H"
- l.336: "play" without "s"

AR: Corrected.

References

E. Ahmed, J. Jaffré, and J. E. Roberts. A reduced fracture model for two-phase flow with different rock types. *Mathematics and Computers in Simulation*, 137:49–70, 2017.

M. Beck, A. P. Rinaldi, B. Flemisch, and H. Class. Accuracy of fully coupled and sequential approaches for modeling hydro-and geomechanical processes. *Computational Geosciences*, 24(4):1707–1723, 2020.

M. Cacace and A. B. Jacquey. Flexible parallel implicit modelling of coupled thermal–hydraulic–mechanical processes in fractured rocks. *Solid Earth*, 8(5):921–941, 2017.

H.-L. Dang and D.-P. Do. Finite element implementation of coupled hydro-mechanical modeling of transversely isotropic porous media in deal. ii. *International Journal of Modeling, Simulation, and Scientific Computing*, 12(01):2150003, 2021.

R. W. Edwards, F. Doster, M. A. Celia, and K. W. Bandilla. Numerical modeling of gas and water flow in shale gas formations with a focus on the fate of hydraulic fracturing fluid. *Environmental science & technology*, 51(23):13779–13787, 2017.

B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Müthing, P. Nuske, A. Tatomir, M. Wolff, et al. Dumux: Dune for multi-{phase, component, scale, physics,...} flow and transport in porous media. *Advances in Water Resources*, 34(9): 1102–1112, 2011.

N. Frih, V. Martin, J. E. Roberts, and A. Saâda. Modeling fractures as interfaces with nonmatching grids. *Computational Geosciences*, 16(4):1043–1060, 2012.

T. T. Garipov, P. Tomin, R. Rin, D. V. Voskov, and H. A. Tchelepi. Unified thermo-compositional-mechanical framework for reservoir simulation. *Computational Geosciences*, 22(4):1039–1057, 2018.

D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

B. J. Graupner, D. Li, and S. Bauer. The coupled simulator eclipse–opengeosys for the simulation of co2 storage in saline formations. *Energy Procedia*, 4:3794–3800, 2011.

G. E. Hammond, P. C. Lichtner, and R. Mills. Evaluating the performance of parallel subsurface simulators: An illustrative example with pflotran. *Water resources research*, 50 (1):208–228, 2014.

R. Hu. A fully-implicit high-order system thermal-hydraulics model for advanced non-lwr safety analyses. *Annals of Nuclear Energy*, 101:174–181, 2017.

D. Kavetski, P. Binning, and S. W. Sloan. Adaptive backward euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow. *International Journal for Numerical Methods in Engineering*, 53(6):1301–1322, 2002.

E. Keilegavlen, R. Berge, A. Fumagalli, M. Starnoni, I. Stefansson, J. Varela, and I. Berre. Porepy: An open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences*, 25(1):243–265, 2021.

J. Kim, H. A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1591–1606, 2011.

T. Koch, D. Gläser, K. Weishaupt, S. Ackermann, M. Beck, B. Becker, S. Burbulla, H. Class, E. Coltman, S. Emmert, et al. Dumux 3–an open-source simulator for solving flow and transport problems in porous media with a focus on model coupling. *Computers & Mathematics with Applications*, 81:423–443, 2021.

G. Kosakowski and N. Watanabe. Opengeosys-gem: a numerical tool for calculating geochemical and porosity changes in saturated and partially saturated media. *Physics and Chemistry of the Earth, Parts A/B/C*, 70:138–149, 2014.

D. Li, S. Bauer, K. Benisch, B. Graupner, and C. Beyer. Opengeosys-chemapp: a coupled simulator for reactive transport in multiphase systems and application to co 2 storage formation in northern germany. *Acta Geotechnica*, 9(1):67–79, 2014.

P. C. Lichtner and S. Karra. Modeling multiscale-multiphase-multicomponent reactive flows in porous media: Application to co2 sequestration and enhanced geothermal energy using pflotran. In *Computational Models for CO2 Geo-sequestration & Compressed Air Energy Storage*, pages 121–176. CRC Press, 2014.

K.-A. Lie. *An introduction to reservoir simulation using MATLAB/GNU Octave: User guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge University Press, 2019.

G. Manzini and S. Ferraris. Mass-conservative finite volume methods on 2-d unstructured grids for the richards' equation. *Advances in Water Resources*, 27(12):1199–1215, 2004.

V. Martin, J. Jaffré, and J. E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM Journal on Scientific Computing*, 26(5):1667–1691, 2005.

R. Martineau, D. Andrs, R. Carlsen, D. Gaston, J. Hansel, F. Kong, A. Lindsay, C. Permann, A. Slaughter, E. Merzari, et al. Multiphysics for nuclear energy applications using a cohesive computational framework. *Nuclear Engineering and Design*, 367:110751, 2020.

A. Mikelić and M. F. Wheeler. Convergence of iterative coupling for coupled flow and geomechanics. *Computational Geosciences*, 17(3):455–461, 2013.

V. Reichenberger. *Numerical simulation of multiphase flow in fractured porous media*. PhD thesis, 2003.

M. Tran and B. Jha. Coupling between transport and geomechanics affects spreading and mixing during viscous fingering in deformable aquifers. *Advances in Water Resources*, 136:103485, 2020.

A. Wilkins, C. P. Green, and J. Ennis-King. Porousflow: a multiphysics simulation code for coupled problems in porous media. *Journal of Open Source Software*, 5(55):2176, 2020.

X. Zhao and B. Jha. Role of well operations and multiphase geomechanics in controlling fault stability during co2 storage and enhanced oil recovery. *Journal of Geophysical Research: Solid Earth*, 124(7):6359–6375, 2019.