Title

Simulation of Gas Production from Multilayered Hydrate-Bearing Media with Fully Coupled Flow, Thermal, Chemical and Geomechanical Processes Using TOUGH+Millstone. Part 2: Geomechanical Formulation and Numerical Coupling

Journal

Transport in Porous Media, 128(1)

ISSN

0169-3913

Authors

Queiruga, Alejandro F
Moridis, George J
Reagan, Matthew T

Publication Date

2019-05-01

DOI

10.1007/s11242-019-01242-w

Peer reviewed

Simulation of Gas Production from Multilayered Hydrate-Bearing Media with Fully Coupled Flow, Thermal, Chemical and Geomechanical Processes Using TOUGH+Millstone. Part 2: Geomechanical Formulation and Numerical Coupling

Alejandro F. Queiruga[2], George J. Moridis[1,2], Matthew T. Reagan[2]

[1] Petroleum Engineering Department, Texas A&M University, College Station, USA; [2] Energy Geosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Matthew T. Reagan: mtreagan@lbl.gov; Alejandro F. Queiruga: afqueiruga@lbl.gov; George J. Moridis: moridis@tamu.edu, GJMoridis@lbl.gov

Abstract

The TOUGH+Millstone simulator has been developed for the analysis of coupled flow, thermal and geomechanical processes associated with the formation and/or dissociation of CH_4 hydrates in geological media. It is composed of two constituent codes: (a) a significantly enhanced version of the TOUGH+HYDRATE simulator, v2.0, that accounts for all known flow, physical, thermodynamic and chemical processes associated with the behavior of hydrate-bearing systems undergoing changes and includes the most recent advances in the description of the system properties, coupled seamlessly with (b) Millstone v1.0, a new code that addresses the conceptual, computational and mathematical shortcomings of earlier codes used to describe the geomechanical response of these systems. The capabilities of the TOUGH+Millstone code are demonstrated in the simulation and analysis of the system flow, thermal and geomechanical behavior during gas production from a realistic complex offshore hydrate deposit. In the second part of this series, we describe the Millstone geomechanical simulator. The hydrate-dependent, rate-based poromechanical formulation is presented and solved using a finite element discretization. A novel multimesh coupling scheme is introduced, wherein interpolators are automatically built to transfer data between the finite difference discretization of TOUGH+ and the finite element discretization of Millstone. We provide verification examples against analytic solutions for poroelasticity and a simplified demonstration problem for mechanically induced phase change in a hydrate sediment.

Keywords

Methane hydrates, Reservoir simulation, Geomechanics, Coupled processes

1 Introduction

In this second part of the series, we describe Millstone, a new geomechanical framework designed to enable the design of novel numerical algorithms for coupled hydrogeological and geomechanical processes. A new multimesh framework was developed to enable the fully coupled geomechanics and flow simulation hydrate-bearing reservoirs with highly refined discretizations.

The Millstone framework automatically performs interpolation between discretizations for flow and geomechanics during the solution of the coupled systems of equations. Millstone can be used as either a standalone simulator or embedded through its application programming interface (API) to support an existing flow simulator. In the hydrate reservoir application in Part 3 of this series, the Millstone simulator uses a finite element method (FEM) discretization, operating embedded inside of TOUGH+HYDRATE V2.0 (see Part 1 of this series; hereafter referred to as the T+H code). As discussed in Part 1 of this series, T+H a simulator developed at Lawrence Berkeley National Laboratory (LBNL) to model non-isothermal CH_4 release, phase behavior, and flow and transport under conditions typical of CH_4-hydrate deposits. Millstone is the second part of the coupled TOUGH+Millstone simulator environment.

A complete geomechanical model is needed when large changes in pressure, P, temperature, T, and hydrate saturation, S_H, occur during hydrate formation and dissociation. In problems with changes that are small in magnitude, the T+H simulator can compute changes to porosity, ϕ, and permeability, k, using a simplified pore compressibility relationship (Moridis et al. 2008) without resorting to a full geomechanical model. However, significant changes in P and T in hydrate systems will lead to changes in S_H which, if sufficiently large, can trigger in turn changes in ϕ and k, as well as substantial displacements and (possibly) geomechanical failure. Further, hydrates may provide the bulk of the mechanical strength to unconsolidated sediments that host many natural hydrate accumulations. This being the case, there is a significant potential for substantial geomechanical responses of such geologic media in response to dissociation caused by (a) the production of hydrate-originating gas, (b) hot fluids from conventional reservoirs ascending through uninsulated wellbore assemblies crossing natural hydrate deposits or (c) even beneath the foundations of production structures that impose large loads on the hydrate-bearing sediments (HBS). Geomechanical concerns over well and reservoir stability when significant changes in P, T and/or S_H are expected to be a significant issue in oceanic hydrate deposits, as opposed to permafrost hydrate deposits where stiffer overburdens provide mechanical support to the reservoir system (Rutqvist et al. 2009).

The Millstone code was developed to directly address these issues, thus enabling the solution of complex (and large) problems of interest. Two core issues related to the code formulation required a complete rewrite from its initial conception: (1) a separate mesh can be used for the mechanical solution and (2) formulations for plane strain and axisymmetry using 2D elements are included in addition to standard 3D Cartesian formulations. By removing the one-to-one element to grid cell requirement, two separate meshes can be generated that are high quality for each of their respective numerical methods (as an optimum mesh for the T+H simulations may not be appropriate for the associated Millstone geomechanical study), avoiding

the long-and-skinny elements that pose conditioning problems in the associated matrix equations. This new approach to the geomechanics has enabled the modeling of complex and computationally expensive hydrate reservoir problems, such as the one studied in Part 3 of this series, that were beyond the capabilities of the previous generation of coupled simulators.

The history of the software development motivating new algorithmic is discussed in Sect. 2. In Sect. 3, the theoretical formulation for the geomechanics used in Millstone is described. The associated numerical methods and algorithms are described in Sect. 5. The software implementation is detailed in Sect. 6. The fully coupled code is validated against the analytical solution for consolidation and is applied to the deformation of a hydrate-bearing sample in Sect. 7. Future developments in-progress and under consideration are summarized in Sect. 8.

2 Background and Motivation

All previously developed coupled flow, thermal and geomechanics simulators in the TOUGH family of codes utilize a one-to-one coupling scheme, in which one flow gridblock is encapsulated by one mechanical element. Many hydrogeological flow solvers that incorporate multiphase and multicomponent materials, T+H included, are based on an integral finite difference method using a two-point flux approximation, which has distinct numerical properties that are materially different from the numerical discretizations used for mechanics. Because of this, structured size-gradated meshes are the best option to yield the most accurate and computationally efficient results with T+H. This leads to a numerical and computational bottleneck when extremely fine meshes are needed to resolve the multiphase flow features, which is crucial in the solution of hydrate reservoir problems. Such meshes result in cells with a long-and-skinny aspect ratio, such as those observed in the left of Fig. 1.

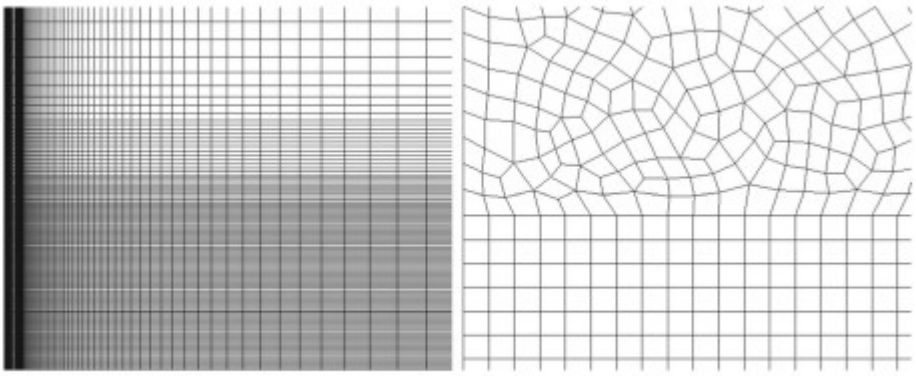

Fig. 1 Two distinct meshes appropriate for the finite difference-based flow solution, left, and the finite element-based mechanical solution, right. This mesh is used in Part 3 of this series

Millstone addresses a fundamental bottleneck when coupling geomechanical and flow codes by introducing a multimesh approach. T+H was previously

coupled to the commercial finite difference simulator $FLAC_3D$ (Itasca Consulting Group 2002) (to form T+H+FLAC3D), and LBNL staff developed the finite element simulator ROCMECH (Kim and Moridis 2012a, b, c) (to form T+H+ROCMECH), both using the one-to-one coupling scheme. This methodology has the advantage of being simple to implement and interpret, but it also means that the mechanical discretization must be the same as that for the flow discretization.

Multiple strategies for coupling mechanics and geomechanics have been explored in the literature, but with similar one-to-one grafting approaches of two codes. The methodology of one flow gridblock per iteration of mechanical element has been used outside the TOUGH software family, e.g., Dean et al. (2006). In Klar et al. (2013), the coupled flow and geomechanics hydrate reservoir simulation is implemented entirely in $FLAC_2D$. A fully coupled simulator using a mixed finite element for both the fluid fields and mechanical fields has been developed by Yang et al. (2014). A dual-mesh framework using a different methodology than that of Millstone was developed by Dana et al. (2018), in which finite elements are used for both the mechanics and flow, but the formulation was limited to single-phase poroelastic flow.

The T+H+$FLAC_3D$ and T+H+ROCMECH had a number of shortcomings that required new developments. While enjoying a strong reputation and wide industry acceptance, T+H+$FLAC_3D$ has significant shortcomings: (a) because of proprietary issues, access to the $FLAC_3D$ source code and sharing of memory between the two applications is not possible, necessitating the use of external files to share important physical variables and parameters needed by the constituent codes—a very slow process—and (b) as a desktop PC application, it was not designed to be run on high-performance computing (HPC) systems, thus limiting its applicability to simpler, smaller geomechanical problems. The analysis of pertinent, large, computationally intense problems requires in-house development to employ more advanced algorithms and target HPC resources. A significant advantage of ROCMECH over $FLAC_3D$ (and the impetus behind its development) was that it was parallelizable and thus applicable to the solution of larger, more complex problems of coupled flow and geomechanics in HBS.

However, parallelization of the previously used numerical methods was restricted by a numerical bottleneck in more challenging reservoir problems and could not address the computational requirements by itself. The meshes generated to optimize the solution of the flow, chemical and thermal processes in the hydrate simulator were of extremely poor quality with respect to the corresponding FEM formulation for the solution of the associated geomechanical problem. As a result, solving the mechanical aspects of the hydrate reservoir resulted in a linear system that was invariably too ill-conditioned and would often not converge. Past effort to overcome this difficulty with T+ROCMECH through MPI-based parallelization (The MPI Forum 1994) and the PETSc library (Balay et al. 2014) did not

alleviate the issues, but further complicated the matter with additional bottlenecks imposed by the demands for parallel load balancing of ill-conditioned matrices.

This motivated the development of the multimesh interpolation scheme to allow for a better quality mesh for the geomechanical problem. An example of the difference between the two high-quality meshes that are appropriate/optimal for each of the two methods is shown in Fig. 1. The unstructured quadrilateral mesh was generated for the finite element mesh using Gmsh (Geuzaine and Remacle 2009). The graduated structured mesh for the flow simulation was generated using MeshMaker 2.0 (Moridis 2016). Further, previous simulations relied on a wedge-like representation of axisymmetric problems. An axisymmetric formulation is included in Millstone that removes the geometric error while also yielding significant speed improvements by reducing the number of unknowns by two-thirds, and improving the stability, conditioning and accuracy of the stiffness matrix system. The additional flexibility provided by the multimesh scheme comes at the cost of interpolation of data between meshes, which increases code complexity and adds more computational steps. However, despite the new steps and even though Millstone is not yet parallelized, the serial implementation of the new algorithm is able to perform pertinent HBS simulations that were previously unsolvable.

3 Theoretical Formulation

The Millstone framework solves the mechanical balance of linear momentum using incremental small deformations with a rate-based stress formulation. The system is solved for the updated displacement $\boldsymbol{\Delta u}$ that satisfies the balance of linear momentum,

$$\rho_R \ddot{\overline{\Delta \mathbf{u}}} = \nabla \cdot (\boldsymbol{\sigma}_0 + \Delta \boldsymbol{\sigma} (\Delta \mathbf{u}, \Delta P, S)) + \rho_R \mathbf{g} \qquad (1)$$

where $\boldsymbol{\sigma}_0$ is the current (constant in time) stress state, the gravitational acceleration is denoted by $\mathbf{g}$ and the density of the rock is $\rho_R = (1-\phi)\rho_s + \phi \sum_\kappa S_\kappa \rho_\kappa$. This update is used to solve for an updated stress increment $\boldsymbol{\Delta\sigma}(\boldsymbol{\Delta u},\boldsymbol{\Delta} P,S)$. For long-term, fully coupled simulations, a quasistatic approximation is used under the assumption that the mechanical deformations equilibrate much faster than the rates of change in the flow variables, such that $\ddot{\overline{\Delta \mathbf{u}}} \approx 0.$.

In the following section, let G denote the shear modulus, K_d denote the drained bulk modulus, K_u denote the undrained bulk modulus, K_f denote the fluid bulk modulus (the inverse of its compressibility) and K_s denote the skeletal grain modulus. Let P denote the fluid pore pressure and ϕ denote the porosity of the media. The mechanical updates are calculated using a model for a linear poroelastic material, where nonlinearity is introduced through the saturation dependence on the moduli ($K_d(S_H)$ and $G(S_H)$) and

through the nonlinear models for the fluid properties in T+H that control the pressure P. Let "d" denote a differential, which will map to a discrete update ▲ in the solution steps. The constitutive response for the total stress is

$$\mathrm{d}\boldsymbol{\sigma} = -\alpha\,\mathrm{d}P\,\mathbf{I} + \left(K_d - \frac{2}{3}G\right)\mathrm{tr}\,(\mathrm{d}\boldsymbol{\epsilon})\,\mathbf{I} + 2G\,\mathrm{d}\boldsymbol{\epsilon} \qquad (2)$$

where $\mathrm{d}\boldsymbol{\epsilon}$ is the change in linear strain from current stress state and α is Biot's coefficient estimates as

$$\alpha = 1 - \frac{K_d}{K_s} \qquad (3)$$

(Biot 1941). Including the poroelastic constitutive response for the fluid pressure provides an alternative formulation of

$$\mathrm{d}\boldsymbol{\sigma} = -\alpha^2 M \nabla \cdot (\mathbf{q}\mathrm{d}t) + \left(K_u - \frac{2}{3}G\right)\mathrm{tr}\,(\mathrm{d}\boldsymbol{\epsilon})\,\mathbf{I} + 2G\mathrm{d}\boldsymbol{\epsilon} \qquad (4)$$

where $\mathbf{q}\mathrm{d}t$ is the fluid flux over the increment path, K_u is the undrained bulk modulus computed as

$$K_u = K_d + \frac{K_s\alpha^2}{\alpha - \phi\left(1 - K_s/K_f\right)} \qquad (5)$$

and M is one of Biot and Willis's coefficients given by

$$M = \frac{K_u - K_d}{\alpha^2} \qquad (6)$$

(Biot and Willis 1957). Equation 4 is not directly applicable because $\mathbf{q}$ and P are computed by T+H using a different discretization. Additionally, the fluid flow description in T+H is more complicated than this poroelastic formulation, where the bulk modulus of the fluid is derived from the Gibbs free energy functions for multiphase mixtures instead of simply one value for the fluid bulk modulus. However, this formulation will be useful for stabilizing and accelerating the convergence of our numerical scheme, as discussed in the next section.

With scant available field and experimental data, a comprehensive constitutive response including the mechanical properties has yet to be developed. Laboratory studies on the mechanical properties of HBS are an active area of research. Thus, using basic physics and a reasonable approach, within the hydrate sediment, the drained bulk modulus and shear modulus K_d and G are estimated using the simplest possible linear relationships with the hydrate saturation S_H, i.e.,

$$K_d = K_{d0}(1 - S_H) + K_{d1} S_H \tag{7}$$

and

$$G = G_0(1 - S_H) + G_1 S_H \tag{8}$$

where K_{d0} and G_0 are the respective moduli for S_H=0 and K_{d1} and G_1 are the moduli at S_H=1 (fully saturated medium) Rutqvist and Moridis (2009). The formulation of Uchida et al. (2012) and Klar et al. (2013) also linearly interpolates the stiffness based on saturation, with a different mathematical expression. For a given reservoir, the values for K_{d1} and G_1 are extrapolated from the in situ measurements of K_d and G given the initial S_H during initial reservoir characterization through wellbore analysis and on-site axial testing of well cores (e.g., see Waite et al. 2018; Hirose et al. 2018). Plastic evolution of the stress states has not yet been considered, but is a high-priority issue and this option will be included in the next code release. Some new theoretical models have been proposed recently for yielding in HBS, such as Uchida et al. (2012, 2016), but these models have yet to be implemented into Millstone at the time of this writing.

Communicating geomechanical effects to the flow description is achieved by altering the geometry of the IFDM control volumes. Each volume cell indexed by b in the flow discretization uses the following calculation to determine the mass M of each component (or total thermal energy) κ in the volume:

$$M_b^\kappa = \int_{V_b} \phi \sum_\beta S_\beta \rho_\beta X_\beta^\kappa \mathrm{d}x = V_b \phi_b \sum_\beta S_\beta \rho_\beta X_\beta^\kappa \tag{9}$$

where the terms are, as elaborated in Part 1 of this series, phase density ρ_β, phase saturation S_β and component mass fraction X_β^κ. The solved deformation is used to modify the IFDM cell volume and porosity in order to provide geomechanical effects for inclusion into the T+H computations. The geometric rate law for the volume of cell b is

$$\mathrm{d}V_b = V_b \nabla \cdot \mathrm{d}\mathbf{u}\, \mathrm{d}x = V_b \,\mathrm{tr}\,(\mathrm{d}\epsilon) \tag{10}$$

where tr(dϵ) is the differential volumetric strain at the cell location. The porosity follows the following rate evolution

$$\mathrm{d}\phi = -\left[\frac{(1-\phi)}{K_d(S)} - \frac{1}{K_s}\right] \mathrm{d}P_e \tag{11}$$

and the effective pressure is defined as

$$P_e = -\frac{\mathrm{tr}\,(\boldsymbol{\sigma})}{3} - P. \tag{12}$$

Note that the total stress tensor and fluid pressure have opposing sign conventions. The permeability, and other quantities, is related to the porosity by the various relations described in Part 1 of this series.

4 Variational Formulations and Coordinate Systems

In this section, we present the weak forms for the geomechanical formulation (based on the stress incrementation approach) that are necessary to produce the FEM formulation. The coordinate systems are introduced during the constraining of the possible solutions **u**(**x**) and performing the integration over the domain, such that there are separate weak forms for the 3D Cartesian, 2D plane strain and 2D axisymmetric coordinate systems.

The 3D Cartesian formulation follows the typical approach for FEM. The weak form is

$$W(\delta\mathbf{u}, \Delta\mathbf{u}; \Delta P, S_H) = \int_\Omega (\nabla\delta\mathbf{u} : (\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}(\Delta\mathbf{u}))) \, \mathrm{d}^3x - \int_\Omega \delta\mathbf{u} \cdot \rho_R \mathbf{g} \, \mathrm{d}^3x - \oint_\Gamma \delta\mathbf{u} \cdot \mathbf{t} \mathrm{d}^2\Gamma \tag{13}$$

which is used to solve for the updated displacement **Δu** for **W**(***δ*u**,**Δu**) for all tests *δu*. The current value of the stress is ***σ***, and the constitutive laws for the stress update mentioned in the previous sections are inserted into the equation for **Δ*σ***(**Δu**). The arguments **Δ*p*** and S_H are additional inputs corresponding to the fields that are the responsibility of the flow simulator and do not have corresponding entries in the ultimate system of equations.

In the 2D formulations, the finite element displacement field has only two components in the plane of the simulation, but the strains and stresses are still handled in 3D. Due to the path-dependent constitutive models, there are four components of the stress field that are tracked including the out-of-plane stress components. The stresses are stored in Kelvin–Voigt notation including this fourth component, i.e., as $\{\boldsymbol{\sigma}\} = \{\sigma_{xx}, \sigma_{zz}, \sigma_{xz}, \sigma_{yy}\}$ in plane strain or $\{\sigma_{rr}, \sigma_{zz}, \sigma_{rz}, \sigma_{\theta\theta}\}$ in axisymmetry.

In plane strain, the displacements only occur along *x* and *z* (choosing to match a vertically oriented nomenclature) with constraints $u_y = \partial u_x/\partial y = \partial u_z/\partial y = 0$ in the out-of-plane direction *y*. The 3×3 strain only has nonzero entries in the 2×2*x*−*z* block,

$$\boldsymbol{\epsilon}_{\mathrm{p-strain}} = \begin{bmatrix} u_{x,x} & \frac{1}{2}(u_{x,z} + u_{z,x}) & 0 \\ \frac{1}{2}(u_{x,z} + u_{z,x}) & u_{z,z} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{14}$$

in which the comma notation has been used to denote the partial derivatives. Note that there is still an out-of-plane σ_{yy} component to the stress that needs to be integrated, a computation dictated by the path-dependent constitutive models used in this formulation.

Axisymmetry adds the following constrains to the displacement solution on an *r*−*z* labeled coordinate system: $u_\theta = \partial u_x/\partial\theta = \partial u_z/\partial\theta = 0$. A $\theta\theta$ component appears in the strain, which now has the following form

$$\boldsymbol{\epsilon}_{\text{axi}} = \begin{bmatrix} u_{r,r} & \frac{1}{2}\left(u_{r,z} + u_{z,r}\right) & 0 \\ \frac{1}{2}\left(u_{r,z} + u_{z,r}\right) & u_{z,z} & 0 \\ 0 & 0 & \frac{u_r}{r} \end{bmatrix}. \tag{15}$$

Similarly to the plane strain case, the $\sigma_{\theta\theta}$ needs to be integrated as well.

The weak form for plane strain is obtained as

$$W(\delta\mathbf{u}, \Delta\mathbf{u}; \Delta P, S_H) = \int_\Omega (\nabla_{xz}\delta\mathbf{u} : (\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}))\,\mathrm{d}x\,\mathrm{d}z - \int_\Omega \delta\mathbf{u}\cdot\rho_R\,\mathbf{g}\,\mathrm{d}x\,\mathrm{d}z - \oint_\Gamma \delta\mathbf{u}\cdot\mathbf{t}\mathrm{d}^1\Gamma \tag{16}$$

where ∇_{xz} is the in-plane gradient. The traction boundary conditions **t** are applied to the region **Γ** on the boundary of the domain **Ω**. The integral is solved for **Δu** to satisfy $W(\delta\mathbf{u},\Delta\mathbf{u})=0$ for all tests $\delta\mathbf{u}$.

The corresponding weak form for the axisymmetric formulation is obtained by separating the integration along the direction from the 2D integrals along the element areas in r and z before applying the divergence theorem. In cylindrical coordinates, the variational integral $W(\delta\mathbf{u},\Delta\mathbf{u})$ after weakening and integrating is

$$W(\delta\mathbf{u}, \Delta\mathbf{u}; \Delta P, S_H) = \int_\Omega (\nabla_{rz}\delta\mathbf{u} : (\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}))\,r - \delta u_r\,(\sigma_{\theta\theta} + \Delta\sigma_{\theta\theta})\,\mathrm{d}r\,\mathrm{d}z - \int_\Omega \delta\mathbf{u}\cdot\rho_R\,\mathbf{g}\,r\,\mathrm{d}r\,\mathrm{d}z - \oint_\Gamma \delta\mathbf{u}\cdot\mathbf{t}r\mathrm{d}^1\Gamma \tag{17}$$

where $\boldsymbol{\nabla}_{rz}$ is the in-plane gradient. The integration domain **Ω** represents the 2D slice at $\theta=0$ that is to be covered by the finite element mesh, and $\mathbf{e}_r$ is the unit vector along r.

5 Numerical Algorithm

5.1 Finite Element Discretization

Millstone can solve both 3D systems using linear nodal tetrahedral elements and 2D systems using linear nodal quadrilateral (i.e., Q1) elements. The computational algorithms seamlessly handle both 2D and 3D systems. For simplicity, the 2D formulation will be used in the rest of this discussion (with some notes regarding 3D distinctions).

The displacement field is discretized by the standard nodal finite element shape functions N_A by, for an r, z-naming of the coordinates,

$$u_i(r, z) = \sum_{A=1}^{4} \tilde{u}_{(Ai)} N_A(r, z) \tag{18}$$

where $\tilde{u}$ is the vector of discrete degrees of freedom that are coefficients to the finite element basis, and (Ai) refers to the degree of freedom index for node A and component i (e.g., a standard vector ordering would be (Ai)=2A+i). The stress field is assigned to the quadrature points of the element with no implied shape functions, of whom four components need to be tracked: σ_{rr},σ_{zz},σ_{rz}, , and $\sigma_{\theta\theta}$. With 2D linear quads, there are four quadrature points and four nodes so that each element is associated with 8 displacement degrees of freedom and 16 stress degrees of freedom, resulting in the total mechanics problem having $2N_{\text{node}}+16N_{\text{elem}}$ degrees of freedom given a particular mesh with said size. However, only ($2N_{\text{node}}-N_{\text{bc}}$) -size matrices need to be solved at once; the stress degrees of freedom are incremented separately and the Dirichlet boundary conditions (N_{bc} degrees of freedom) are manipulated out of the matrix system.

The right-hand side vector **R** for a linear system is determined from the variational integral W by performing the chain rule against the shape functions,

$$R_{(Ai)} = \frac{\partial W}{\partial \delta u_i} N_A + \frac{\partial W}{\partial \nabla \delta u_i} : \nabla N_A. \tag{19}$$

The system is solved by assembling the Jacobian matrix, **K**, obtained by differentiating the components of **R** against the degrees of freedom,

$$K_{(Ai)(Bj)} = \frac{\partial R_{(Ai)}}{\partial u_j} N_B + \frac{\partial R_{(Ai)}}{\partial \nabla u_j} : \nabla N_B, \tag{20}$$

where N_B is the shape function of the cross-derivative node B and j is the component of the cross-derivate. Unlike the assembly of the flow system of equations in T+H which employs numerical differentiation, the expressions above are implemented using a compilation package, Popcorn (Queiruga 2018b), that uses symbolic differentiation. The system of equations to solve is thus

$$0 = \mathbf{K}\,\Delta\tilde{u} + \mathbf{R}\,(\tilde{u})\,. \tag{21}$$

For the linear incremental constitutive equations above, the resulting equations are linear, and the nonlinearity with respect to the hydrate saturation is treated in the staggered iteration. The above formulation is capable of handling—and was designed for—nonlinear constitutive responses and general partial differential equations.

5.2 Interpolation

The use of separate meshes in the new TOUGH+Millstone coupling requires additional computations to transfer data back and forth between the flow/thermal and mechanical components of the problem. The library interface for Millstone was designed to be easy to incorporate into any existing hydrogeological simulator. The Millstone framework automatically determines a suitable projection and interpolation mapping at simulation initialization without requiring an additional preprocessing phase or requiring extra information to be provided from the T+H simulator. A simple and robust interpolation-based approach was developed for the purpose (described below), such that Millstone requires no knowledge of the flow discretization other than a cloud of points with associated flow values. (The routine can work backwards as well to interpolate values from, for example, a Peridynamics point cloud onto the flow field point cloud.) The robustness of the interpolation approach also allows the discretizations to vary significantly in resolution (e.g., very fine flow resolution near a well) and provides accurate estimates even when the domains do not perfectly overlap (e.g., localized flow in a much larger formulation exhibiting significant deformation.)

The two discretizations are overlaid onto each other, as illustrated in Fig. 2, and appropriate basis functions are used to computer the corresponding fields at nodal and cell-center locations. The T+H numerical discretization is not formulated using a function basis, so one is constructed from a Delaunay triangulation (or tetrahedralization) of the flow gridblock centers. The coefficients to basis functions for each triangle (or tetrahedron) are a by-product of the triangulation algorithm, so no additional computation is needed. To use the pore fluid pressure P, temperature T, and hydrate saturation S_H values from T+H (explicitly P^{flow}, T^{flow} and S_H^{flow}), the fields are given FEM counterparts P^{mech}, T^{mech} and S_H^{mech} which use the Q1 interpolation. The nodal values of P^{mech} etc. are always set to the interpolated values from the triangulation of P^{flow}. The FEM basis functions are used to directly calculate fields at any point. As noted in Fig. 2, the stresses located at quadrature points do not have an assumed basis, but this is not necessary.

The geometric correspondence between flow cells and the FEM mesh is stored as a list of overlapping flow cells associated with each element. A spatial binning algorithm is used to perform the determination in $O(N)$ time.[1] The correspondence list is then used when quantities are projected or interpolated between the two meshes.

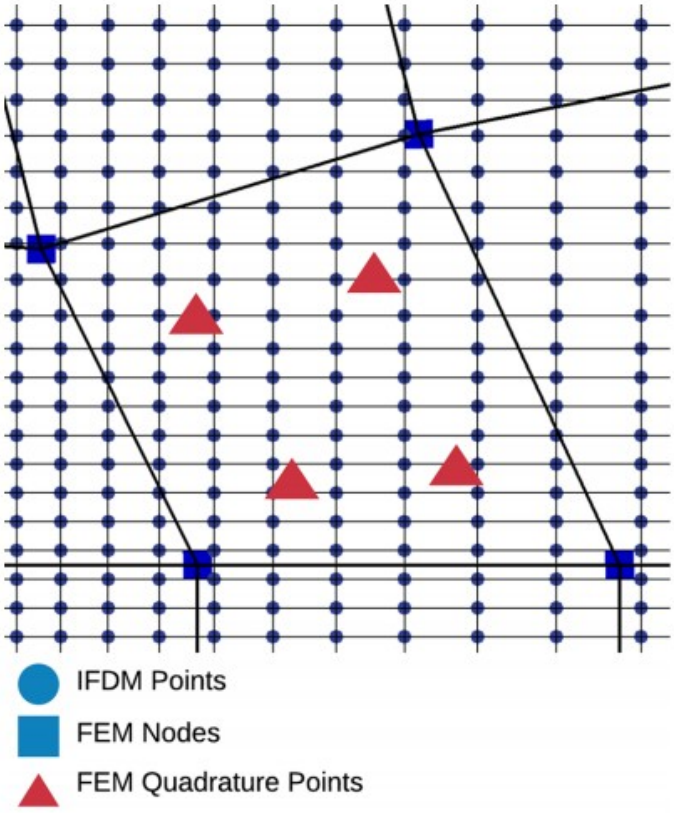

Fig. 2 Non-matching meshes used for flow and mechanics problem. Values associated with the flow, such as P, T and S_κ, are associated with the IDFM points, the displacements are associated with the FEM nodes, and the stresses are associated with the FEM quadrature points. (This illustration was generated using the two meshes illustrated in Fig. 1)

The finite element shape functions are used to interpolate from the mechanics mesh to the flow mesh. For some field Y, let Y_A denote the nodal value used to evaluate $Y(\mathbf{x})=\sum_A N_A(\mathbf{x})Y_A$. The reference coordinate on the element at the flow cell location is obtained by inverting the isoparametric coordinates, which must be done using Newton's method to solve:

$$\mathbf{x}_b^{\text{flow}} = \sum_{A=1}^{4} \mathbf{x}_A^{\text{mech}} N_A^{\text{quad}}\left(\boldsymbol{\xi}_b\right). \tag{22}$$

In this manner, expressions that are more complex than just a field interpolation can be evaluated, such as the porosity rate (Eq. 11) evaluated at the point $\mathbf{x}_b$:

$$\mathrm{d}\phi(\mathbf{x}_b) = \mathrm{d}\phi\left(S_H\left(\mathbf{x}_b\right), \quad \mathrm{tr}\left[\left(\mathbf{x}_b\right)\right]\right) = \mathrm{d}\phi\left(S_{Hb}, \mathrm{tr}\left[\sum_{A=1}^{4} \nabla^s \mathbf{u}_A N_a^{\text{quad}}\left(\boldsymbol{\xi}_b\right)\right]\right). \tag{23}$$

Millstone performs a Delaunay triangulation of the flow cell locations to build a suitable interpolation scheme. (The algorithm works the same way using the Delaunay tetrahedralization in 3D.) For each node in the FEM mesh, a geometric search obtains the triangle within which it lies, and the shape

functions for that triangle are evaluated at the point. The standard linear nodal shape functions on the triangle (equal to the area coordinates) are used. The shape function values are saved as an interpolation matrix M that performs the operation

$$Y_A^{\text{mech}} = M_{\text{Ab}}\, Y_b^{\text{flow}}, \tag{24}$$

where the matrix entries are equal to the triangular shape functions for flow cell b evaluated at the mechanical node A,

$$M_{\text{Ab}} = N_b^{\text{tri}}\left(\mathbf{x}_A^{\text{mech}}\right); \quad \mathbf{x}_A^{\text{mech}} \in \text{tri}, \quad \mathbf{x}_b^{\text{flow}} \in \text{tri}. \tag{25}$$

A spatial hash is used to quickly determine in which triangle tri the mechanical point resides, which has the corresponding IFDM grid points $\mathbf{x}_b^{\text{flow}}$ as its vertices. Only three values are nonzero per row, corresponding to the vertices of the triangle within which the FEM node resides. (There are four values in 3D for the vertices of the tetrahedron.) A fringe case occurs when an FEM node lies outside of the span of the triangulation, which almost always occurs in a given simulation because the flow cell centers are not located on the domain boundary. For the FEM nodes lying on the boundary, the values are extrapolated using the closest cell in the flow mesh, modifying the expression to be

$$M_{\text{Ab}} = \delta_{\text{Ab}}; \quad \mathbf{x}_A^{\text{mech}} \notin \{\text{tri}\}, \quad \mathbf{x}_b^{\text{flow}} \quad \text{closest to} \quad \mathbf{x}_A^{\text{mech}}. \tag{26}$$

This also allows geomechanical meshes that are much larger than the flow domain, which are necessary in certain problems where significant geomechanical response occurs outside the permeable region.

5.3 Iterative Solution

The initial stress field is obtained by solving the quasistatic mechanical system of equations once, using the initial state given from T+H with no preprocessing phase required to generate initial conditions. The displacement field is reset back to 0, as it was only solved to obtain an admissible stress field.

The solvers are fully coupled inside of the Newton's method loop, such that the mechanical system of equations is solved every time the flow Jacobian matrix is solved. After each nonlinear solution to the multiphase flow fields, T+H passes the current flow states to Millstone, which, after the solution of the mechanical linear system, passes the resulting stresses and strains back to T+H. The current values for the fields P, T and S_H at k are interpolated onto the finite element mesh after every update. The procedure is repeated

until the Newton's method loop for the flow convergences before advancing to the next time step.

The algorithm for the fully coupled geomechanical multiphase flow iteration is illustrated in the flowchart in Fig. 3. Let k denote the current iteration index of the Newton's method loop. The increment for the pressure is obtained by $\boldsymbol{\Delta} P = P^{k+1} - P^k$. In the incremental fashion, after solving for a new admissible $\boldsymbol{\Delta}\mathbf{u}$, the stress state and total displacement are updated by

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^k + \Delta\boldsymbol{\sigma}(\Delta P, \Delta u) \tag{27}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}^k \tag{28}$$

The update for the cell volume and porosity is obtained by analytically integrating the path from k to $k + 1$ for Eqs. 10 and 11,

$$V_b^{k+1} = V_b^k + V_b^k \exp\left(\mathrm{tr}\Delta\epsilon^k\right) \tag{29}$$

and

$$\phi_b^{k+1} = 1 - \frac{K_d}{K_s} + \left(\frac{K_d}{K_s} - 1 + \phi_b^k\right) \exp\left(\frac{P_e^{k+1} - P_e^k}{K_d}\right). \tag{30}$$

Note that it is the update of the effective pressure P_e that is used in the equation of porosity. The values of ϵ, P_e and K_d in the right-hand sides of Eqs. 29 and 30 are evaluated using the finite element basis representation for $\mathbf{u}(x)$, $P(x)$ and $S_H(x)$ at the corresponding flow point $\mathbf{x}_b^{\text{flow}}$. The values stored at the flow points are used for V_b^k and ϕ_b^k and directly modified for V_b^{k+1} and ϕ_b^{k+1}.

The steps are described as:

1. Use the interpolation matrix to obtain P, T, S_κ on the FEM nodes (Eq. 24) and compute $\Delta P = P^{k+1} - p^k$.

2. Assemble and solve FEM stiffness matrix for $\boldsymbol{\Delta}\mathbf{u}$, Eq. 21, using either Eqs. 2 or 4 for $\boldsymbol{\Delta\sigma}(\boldsymbol{\Delta}\mathbf{u})$.

3. Increment stress and displacement fields (Eqs. 27 and 28). Equation 2 is used for the stress update.

4. Increment $V_b^{k+1} = V_b^k$ and $\phi_b^{k+1} = \phi_b^k$ at IFDM cells (Eqs. 29 and 30).

5.4 Iteration Stability

The different stress measures and rate formulations that can be used to describe poroelasticity give additional options when designing a numerical

algorithm. The two ways of describing the stress allow two calculation options, the ones given by Eqs. 2 and 4. The scheme “Kd and α” uses Eq. 2 to solve for the new displacement using the pressure update given by T+H. The “Ku” scheme uses Eq. 4 and solves for the incremented displacement by including the compressibility of the fluid into the mechanical stiffness matrix. For this scheme, the fluid bulk modulus K_f is used to compute the undrained bulk modulus K_u and is an additional input to Millstone. However, in T+H, the effect of the fluid compressibility is obtained from the empirical Gibb’s free energy correlations for fluid mixtures. Because the fluid pressure is solved by T+H, the effective modulus is only an estimate, and the stress must be updated using ΔP from the value passed from T+H using Eq. 2 in both cases.

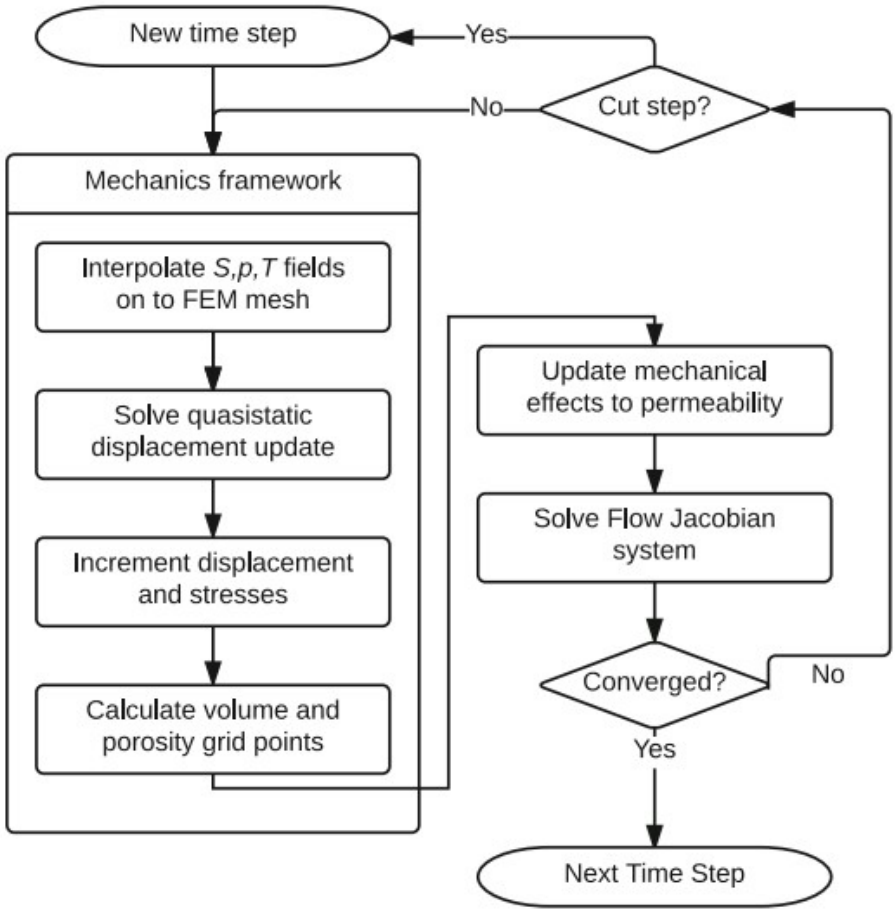

Fig. 3 Flowchart of the TOUGH+Millstone simulation loop. The box labeled “Mechanics framework” represents the interface “boundaries” for the Millstone package, where the steps within the box are implemented in Python and C, and everything outside is the T+H code

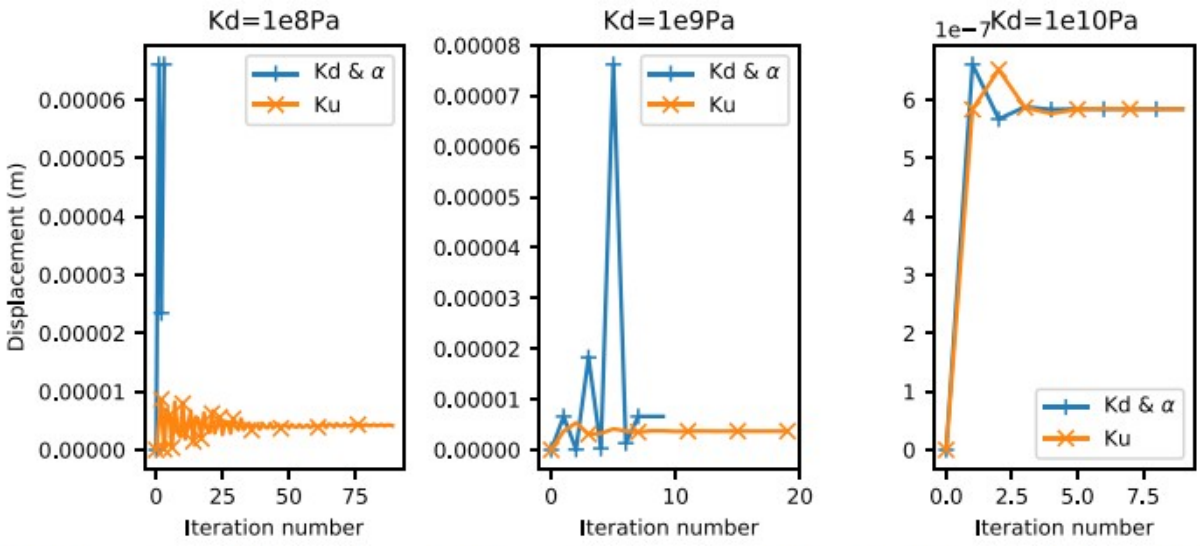

Fig. 4 Convergence of the displacement for the first step of the consolidation problem for three different bulk moduli with grain moduli $K_s = 2.205$ GPa, 2.205 GPa and 22.05 GPa, respectively. The drained modulus cannot exceed the bulk modulus, hence the last alteration. The key "Kd and α" corresponds to the non-preconditioned iteration, and the key "Ku" corresponds to the preconditioned iteration. The "Kd" iteration terminates upon detecting invalid properties for the two softer cases. The fluid bulk modulus is approximately 2.205 GPa

The two algorithms are tested on the first loading step of the consolidation problem in Sect. 7. The values of the displacement during the convergence process at the top of the domain are shown in Fig. 4. In these cases, the pores are filled with pure liquid water such that the fluid bulk modulus is approximately 2.205 GPa which is used as the estimate for the preconditioner. The "Kd" iteration fails to converge at the softer drained moduli. It is clear that the "Ku" scheme offers faster convergence and is necessary for softer geomaterials where the fluid is the main load-bearing component. Preconditioning the iteration becomes more challenging when compositional and phase changes occur can that drastically change the bulk modulus of the pore fluid; improving the robustness of the solution technique for such a scenario is an active area of development in Millstone.

6 Implementation

Millstone is implemented using the Cornflakes/Popcorn open-source domain-specific language and scientific package (Queiruga 2018a, b). (The name "Millstone" was chosen by the Cornflakes/Popcorn pun scheme: Millstone generates and assembles rock-related kernels.) The program is written in both Python and C to easily express the high-level simulation structure while maintaining performance, allowing for flexible experimentation with different numerical methods and solution techniques. The software package SWIG is used to generate the wrapping code between C and Python. The Python-based code generation system of Popcorn is used to produce the finite element compute kernels by expressing the variational form and calculation

procedure with a high-level abstraction. The same system was applied in Queiruga and Moridis (2017) to generate Peridynamics expressions, which was an experimental capability prototyped for the Millstone code. The code generation system transforms the expressions using SymPy (SymPy Development Team 2016) and outputs C code that is compiled and linked to Millstone. Current work seeks to extend this system to plastic constitutive models by generating return mapping routines after specifying a yield surface and a hardening rule in tensor notation.

The Millstone framework can be used to solve a mechanics-only problem in a standalone Python script. An API is provided that is designed for simple incorporation into a code through Python, C, or Fortran interfaces. It is coupled to T+H at compile time through a Fortran interface, where an embedded Python interpreter executes the Millstone library and invokes data from the input files. Communication between Millstone and the parent flow simulator is achieved by wrapping the data arrays with NumPy array objects (Van der Walt et al. 2011) through the API to be read and modified by Millstone. The API exposes the following routines to a flow simulator to set up and run a quasistatic geomechanical simulation:

1. Provide the flow mesh
2. Register a flow field with a name and an array pointer
3. Initialize the stress field
4. Solve the incremented displacement and stress
5. Accept or Reject the time step
6. Output a visualization file.

At a minimum, the flow simulator must register the pressure P, porosity ϕ and gridblock volume V fields, and the hydrate saturation S_H is further required for the hydrate-dependent constitutive responses. The Python-based input files allow the user to specify complicated behavior by inserting code that will be executed inside the TOUGH+Millstone simulator to, for example, implement time-varying boundary conditions, generate a new mesh in a convergence study or add post-processing calculations.

For problems where the fully coupled simulation is intractable and time constraints only permit a few flow-only simulations, one-way coupling can be performed to estimate the geomechanical response using a Python script that parses output files using the utility tough_convert (Queiruga and Reagan 2018).

Assembly of the mechanical stiffness matrix and load vectors can be performed in parallel using OpenMP. Matrix solution in both the mechanics and flow codes is performed using either LIS (Nishida 2010) (which has OpenMP support), PETSc (Balay et al. 2014) or SciPy (Jones et al. 2001–2018).

7 Verification Examples

7.1 Terzaghi's Problem

A standard problem used to verify poroelastic simulators is Terzaghi's problem of consolidation in response to draining of the pore fluid. It is difficult to design a problem where the numerical results will exactly the analytical solutions because T+H derives all properties, such as the bulk modulus and viscosity, from empirical relations. That is, T+H uses a more accurate nonlinear representation for the properties of water than the fixed property values used in the analytical solution. To match the approximate solution, the fluid bulk modulus and viscosity empirical relations are evaluated at P and T and fed into the analytical solution to a very high precision. The theory of poroelasticity uses a small-strain approximation for both the fluid and skeleton, so a very slight load is applied.

In this setup, the domain is a 1 m by 1 m square with plane strain conditions. The porosity is initially ϕ=0.3, the drained bulk modulus is K_d=220 GPa, the shear modulus is G=220 GPa and the grain modulus is K_s=100K_d. The medium permeability is k=10×10^{-13} m^{-2}. The solid and the fluid are initially in equilibrium with the reservoir pressure along the top edge, P=1 atm=335 MPa. A load $\bar{t}$ =10 MPa is applied instantaneously and the system equilibrates in a very short timescale. The equilibration of the initial "instantaneous" loading is performed using a few very large timesteps in the coupled simulator. This corresponds to a relaxation process with the correct answer to within the iteration toleration because backward Euler is used to integrate the time-varying flow equations and the mechanical equations are solved quasistatically. Gravity is neglected to match to the analytical solution. The problem is one dimensional along the height.

The derivation to the following analytical solution can be found in Verruijt (2013). The instantaneous displacement is

$$u_0 = \left(K_u + \frac{4}{3}G\right)^{-1} H\bar{t} \qquad (31)$$

and the instantaneous pressure increase is

$$P_0 = \frac{K_u - K_d}{\alpha}\left(K_u + \frac{4}{3}G\right)^{-1} \bar{t}, \qquad (32)$$

with respect to the initial pressure, where H is the domain height. The analytical solution to the pressure field is

$$\frac{P(z,t)}{P_0} = \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{-1^{k-1}}{2k-1}\cos\left(\frac{(2k-1)\pi z}{2H}\right)\exp\left(\frac{-(2k-1)^2\pi^2 ct}{4H^2}\right) \tag{33}$$

where c is a time constant equal to

$$c = \frac{k/\eta}{s + \alpha^2\left(K_u + \frac{4}{3}G\right)^{-1}} \tag{34}$$

where $s = \phi K_f^{-1} + (\alpha - \phi)\, K_s^{-1}$, k is the permeability and η is the viscosity. The analytical solution to the displacement field is

$$u_z(z,t) = a\bar{t}(H-z) - (a-a_i)H\bar{t}\frac{8}{\pi^2}\sum_{k=1}^{\infty}\frac{-1^{k-1}}{2k-1}\cos\left(\frac{(2k-1)\pi z}{2H}\right)$$
$$\exp\left(\frac{-(2k-1)^2\pi^2 ct}{4H^2}\right) \tag{35}$$

This series converges fairly quickly; 30 terms are used to calculate the solutions to compare to the numerical results.

To rigorously validate the behavior of the numerical approximation and its implementation, the order of convergence is estimated by performing successive simulations with grid refinement. The timestep size, flow mesh size and mechanical mesh size are refined at the same rate by a factor h from a baseline choice: $\Delta t = h\Delta t^*,\ \Delta x_{\text{flow}} = h\Delta x^*_{\text{flow}}$, and $\Delta x_{\text{mech}} = h\Delta x^*_{\text{mech}}$. The error is calculated by the mean square error of the differences at various nodal coordinates at snapshots in time against the analytical solution,

$$e(h) = \frac{1}{N}\sqrt{\sum_{i}^{N}\left(u_{\text{num}}(z_i,t_i) - u_{\text{anal}}(z_i,t_i)\right)^2} \tag{36}$$

which is then used to estimate the order of convergence by performing the standard log–log regression,

$$\log e(h) = a\log h + b \tag{37}$$

where *a* will be the order of convergence. For linear finite elements, two-point fluxes, backward Euler and the linear interpolation mesh transfer scheme, we expect a linear order of convergence.

The displacement results for various discretization refinements for the pressure unloading are plotted along side the analytical solution in Fig. 5. The displacements are probed at the center of the domain, $u_z(1/2,t)$. ("Oracle" refers to the nature of analytical solution to provide the exactly correct output of the computer program.) The error obtained for both P and u_z during the discretization refinement process is plotted in Fig. 6. The log–log regression—which calculates the slopes of the curves on the log–log plot—yields an order of convergence of 0.950 for P and 0.979 for u_z, which confirms the expected numerical accuracy of TOUGH+Millstone.

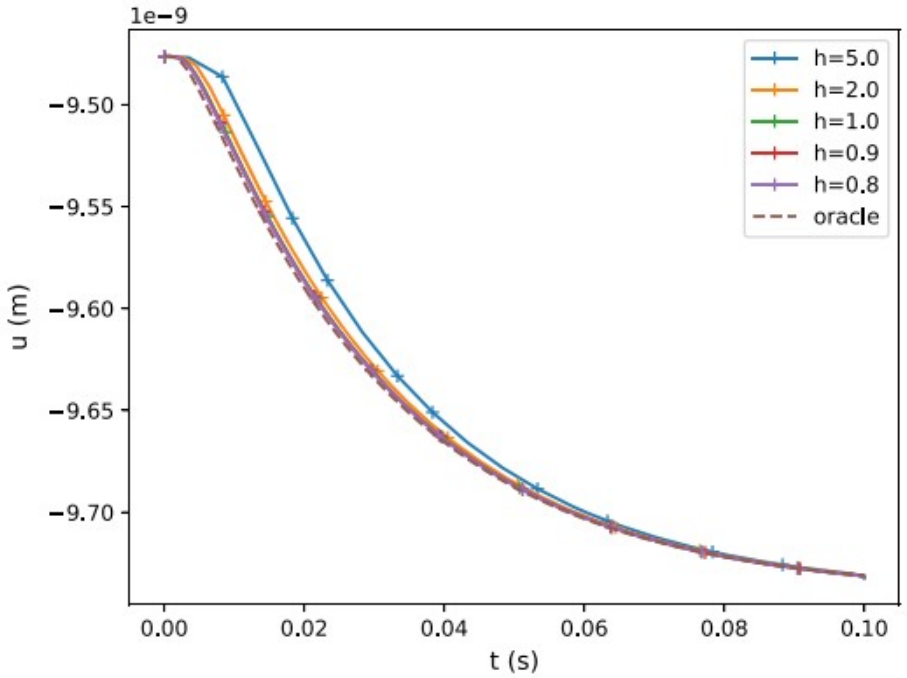

Fig. 5 Comparison of the numerical solution of Terzaghi's problem with Millstone against the analytical solution by probing the displacement field at the center of the domain ($z = 0.5$ m). The factor h controls the mesh size and time step size

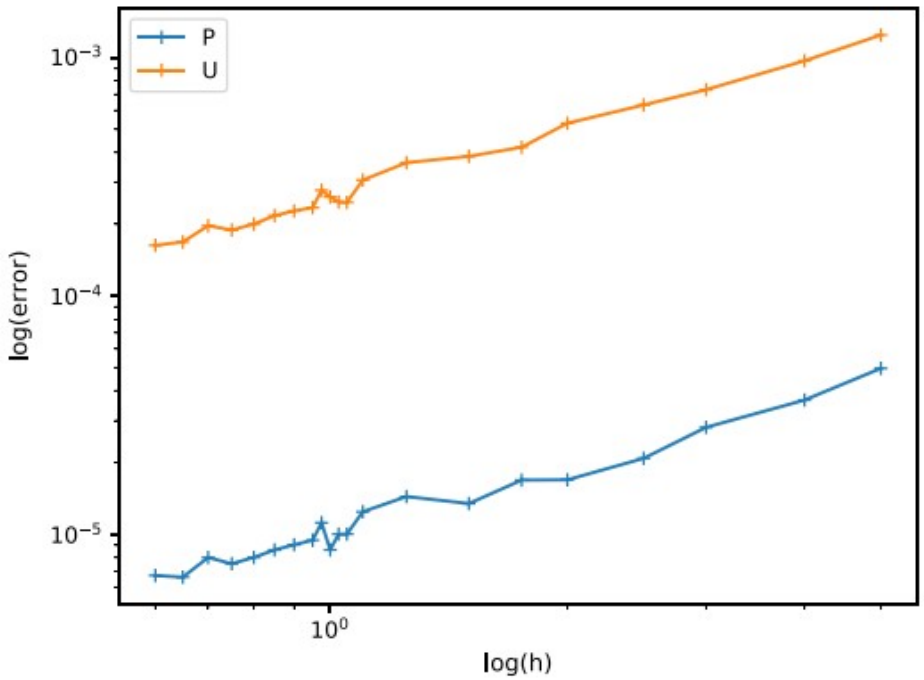

Fig. 6 Logarithm of the error of numerical solution of Terzaghi's problem with Millstone compared to the analytical solution for both the pressure and displacement fields. The factor h controls the mesh size and time step size

7.2 De Leeuw's Problem

A cylindrical soil sample is held between two rigid plates at its top and bottom, and its side is open to the air. A load is instantly applied along the outer ring, and fluid is allowed to drain out of its sides. This is similar to Mandel's problem with a cylindrical sample, making it appropriate to verify the axisymmetric implementation. The parameters used in the verification problem are exactly the same as the previous example, except that the domain is now a cross section of a cylinder with height 1 m and radius 1 m. The problem is one dimensional along the radius.

The analytical solution to this problem is taken from Verruijt (2013) as well. The solutions for the instantaneous displacement and pressure are

$$u_0 = \left(K_u + \frac{1}{3}G\right)^{-1} R\bar{t} \tag{38}$$

and

$$P_0 = \frac{K_u - K_d}{\alpha}\left(K_u + \frac{1}{3}G\right)^{-1} \bar{t}. \tag{39}$$

Let

$$m = \frac{1}{2\alpha^2}\left(\frac{K_d + 4G/3}{2G}\right)\left(\alpha^2 + s\left(K + \frac{1}{3}G\right)\right). \tag{40}$$

The roots of the equation $2m\xi J_0(\xi) - J_1(\xi)$ are denoted by ξ_j, where J_0 and J_1 are the zeroth- and first-order Bessel functions of the first kind. These must be determined numerically. The pressure at a point r away from the center column at time t is then given by the infinite sum

$$\frac{P(t,r)}{P_0} = \sum_{j=1}^{\infty} \frac{J_0\left(\xi_j\right) - J_0\left(\xi_j r/a\right)}{\left(1 - m\xi_j^2 - 1/(4m)\right) J_0(\xi_j)} \exp\left(-\xi_j^2 c_v t/a^2\right). \tag{41}$$

This series converges very slowly; 600 terms are used to calculate the solutions when computing the error.

The results for various discretizations are plotted in Fig. 7, and the errors obtained by sweeping through all discretizations are plotted on a log–log plot in Fig. 8. Following the same procedure as before, the approximation error is observed to converge at a rate of 1.033, which again confirms the expected behavior of the code.

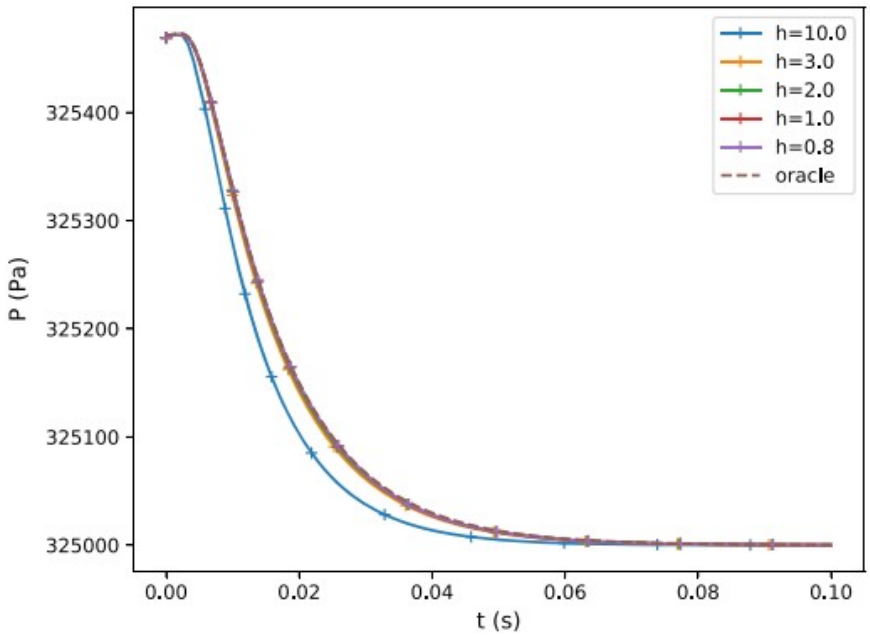

Fig. 7 Comparison of the pressure field solution at the center of the domain ($r = 0.5$ m) for the analytical solution ("oracle") and numerical solutions using various mesh discretizations and timestep sizes

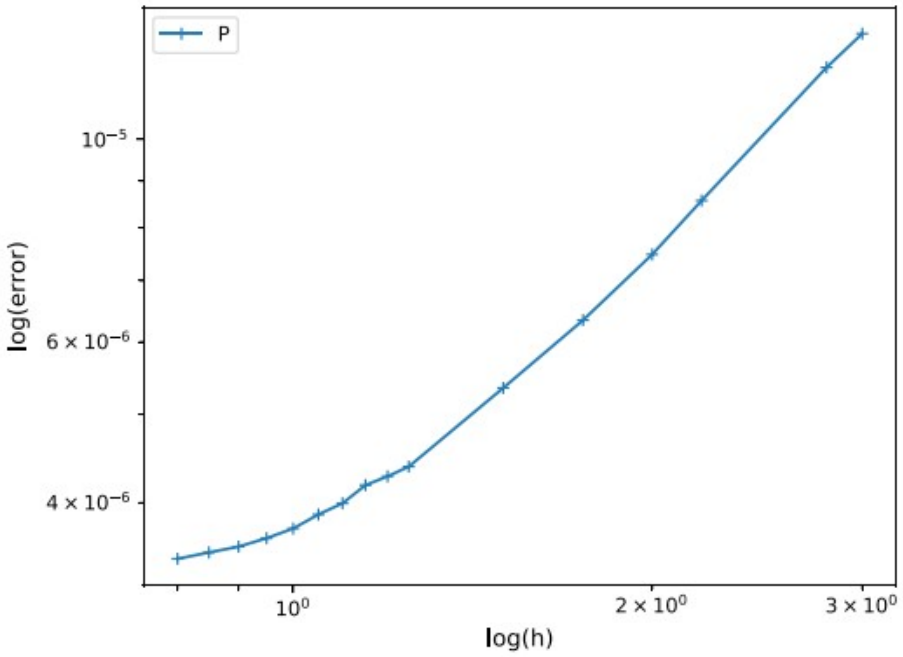

Fig. 8 Log–log plot of the error of TOUGH+Millstone solving de Leeuw's problem against the analytical series

7.3 Compression of a Methane Hydrate-Bearing Sample

Due to the nonlinear nature of the empirical responses for the hydrate phases, it is difficult to develop an analytical solution to a coupled

mechanics–hydrate system (and no such solution is available in the literature). We consider an analogue to Terzaghi's consolidation problem above with a hydrate-filled domain. The properties of the sample are the same as in the above verification example, but it is at a higher pressure and different temperature to support a stable and uniformly distributed hydrate system at an initial saturation of S_H=0.5 in the hydrate–liquid water region. The sample is initially at T=7.2 C and P=20 MPa, with an equivalent applied traction at the top. Instead of an instantaneous loading and slow draining, the load is increased to 320 MPa over a course of 300 s. This problem was designed from prior knowledge of the methane gas–hydrate equations of state to pass through an expected maximum in the hydrate equilibrium saturation as the pressure is increased.

The evolution of the fields is shown in Fig. 9. We demonstrate that through mechanical loads as the only driving force to a simulation, TOUGH+Millstone is able to capture phase responses in the hydrate system. At the beginning of the process, hydrate formation occurs, until a maximum saturation is achieved, after which point the hydrate dissociates as the applied load continues to increase. This reversal is caused by the temperature increase associated with the pressure increase: Once the equilibrium temperature for a given pressure is reached, thermal dissociation begins leading to a reduction in S_H in accordance with the thermodynamics of the methane=H_2O phase diagram.

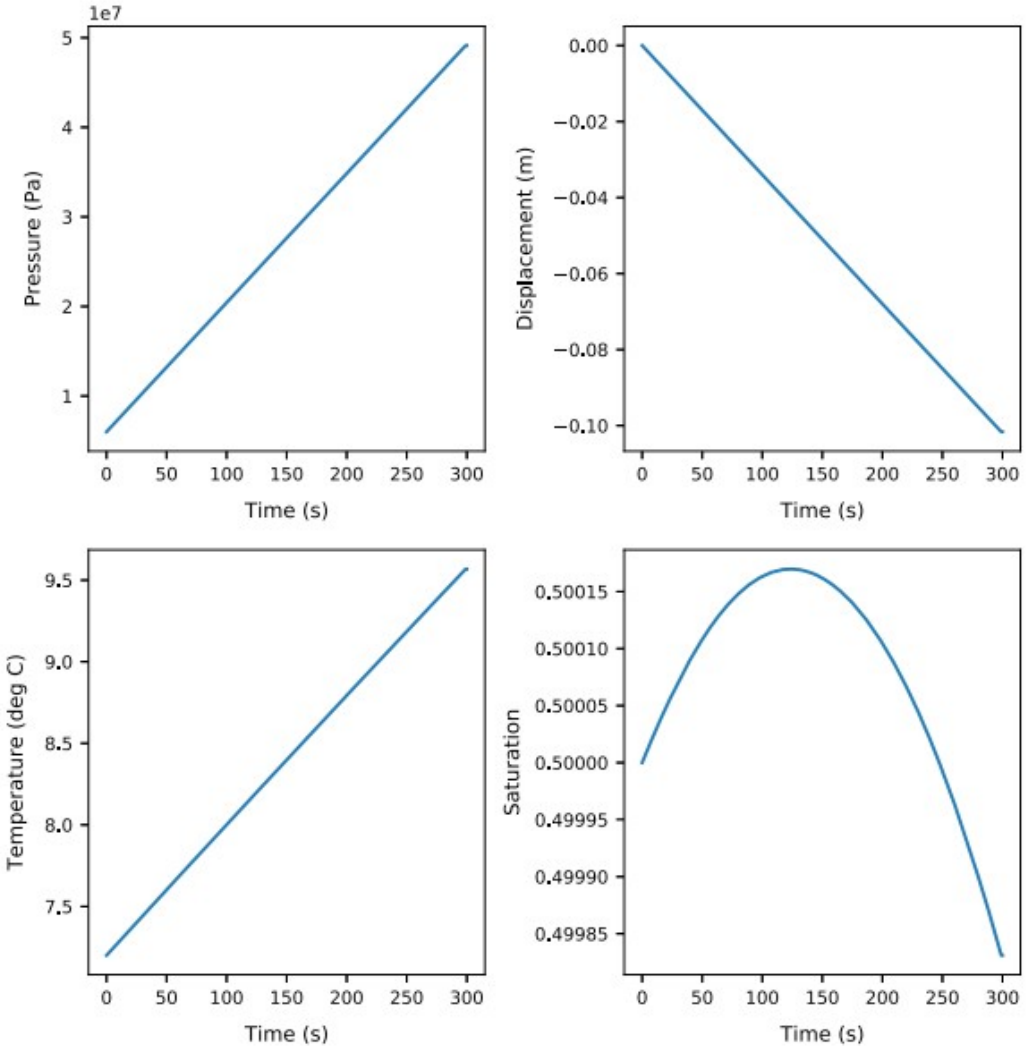

Fig. 9 Plot of the fluid pressure, top displacement, temperature and hydrate saturation during the loading process

8 Conclusion

In this series, we document the development and use of the TOUGH+Millstone simulator, which involves a coupling of the fully implicit TOUGH+HYDRATE v2.0 (T+H) simulator, describing flow, thermal and chemical processes in hydrate-bearing media, with the Millstone v1.0 geomechanical model to describe the corresponding geomechanical response. Here, we described the geomechanical simulator component, Millstone, that was written with the new Cornflakes/Popcorn scientific package to utilize state of the art software architecture paradigms. We described the novel coupling numerical formulation and software architecture approach used to integrate a geomechanical solver with a hydrogeological flow solver, namely the TOUGH+HYDRATE solver detailed in the first part of this series. A new multimesh coupling methodology was

presented to resolve accuracy and conditioning problems that plagued one-to-one problems as TOUGH+HYDRATE was applied to more complex reservoir problems. The coupling methodology was verified against analytical solutions to Terzhaghi's and de Leeuw's consolidation problems. Analysis of discretization refinement confirms that the combination of FEM, IFDM, time stepping and mesh coupling has a linear approximation error. The geomechanical coupling with the hydrate phase was also explored, demonstrating the ability to change its state through only mechanical loading.

In the third and final part in this series, we present the application of the fully coupled TOUGH+Millstone software package to the simulation of production from a realistic hydrate-bearing reservoir. The problem studied in this part requires extremely refined meshes for the flow that "hit the limits" of the previous generation of geomechanical solvers that were based on a one-to-one coupling scheme, and was the impetus to development of Millstone and the new algorithms presented. This example will illustrate the importance of including a complete geomechanical solver to fully capture the behavior of the reservoir, and distinct effects not observed in simplified geomechanical treatments.

Footnotes

1. The algorithm is described in detail in "Appendix A" of Queiruga (2015).

Acknowledgements

This work was supported by the Assistant Secretary for Fossil Energy, Office of Natural Gas and Petroleum Technology, through the National Energy Technology Laboratory, under the U.S. Department of Energy, Contract No. DE-AC03-76SF00098, and also through a funded collaboration with Chevron.

References

Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zhang, H.: PETSc web page. http://www.mcs.anl.gov/petsc (2014)

Biot, M.A., Willis, D.G.: The elastic coefficients of the theory of consolidation. J. Appl. Mech. 24, 594–601 (1957)

Biot, M.A.: General theory of three-dimensional consolidation. J. Appl. Phys. 12(2), 155–164 (1941)

Dana, S., Ganis, B., Wheeler, M.F.: A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. J. Comput. Phys. 352, 1–22 (2018)

Dean, R.H., Gai, X., Stone, C.M., Minkoff, S.E., et al.: A comparison of techniques for coupling porous flow and geomechanics. SPE J. 11(01), 132–140 (2006)

Geuzaine, C., Remacle, J.F.: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int. J. Numer. Methods Eng. (2009)

Hirose, T., Tanikawa, W., Hamada, Y., Lin, W., Hatakeda, K., Tadai, O., Wu, H.Y., Nomura, S., Abe, N., Gupta, L.P., Sugihara, T., Masaki, Y., Kinoshita, M., Yamada, Y.: Strength characteristics of sediments from a gas hydrate deposit in the Krishna–Godavari basin on the eastern margin of India. Mar. Petrol. Geol. (2018). https://doi.org/10.1016/j.marpetgeo.2018.08.017. http://www.sciencedirect.com/science/article/pii/S0264817218303416

Itasca Consulting Group: Flac3d: Fast Lagrangian analysis of continua in 3 dimensions. Technical report. Minneapolis, Minnesota (2002)

Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–2018). http://www.scipy.org/. Accessed 01 Jan 2016

Kim, J., Moridis, G.J.: Gas flow tightly coupled to elastoplastic geomechanics for tight and shale gas reservoirs: material failure and enhanced permeability. In: Americas Unconventional Resources Conference, Pittsburgh, Pennsylvania, June 2012a

Kim, J., Moridis, G.J.: Modeling and numerical simulation for coupled flow and geomechanics in composite gas hydrate deposits. In: 46th U.S. Rock Mechanics/Geomechanics Symposium, Chicago, Illinois, 24–27 June 2012b

Kim, J., Moridis, G.J.: Numerical studies for naturally fractured shale gas reservoirs: Coupled flow and geomechanics in multiple porosity/permeability materials. In: 46th U.S. Rock Mechanics/Geomechanics Symposium, Chicago, Illinois, 24–27 June 2012c

Klar, A., Uchida, S., Soga, K., Yamamoto, K., et al.: Explicitly coupled thermal flow mechanical formulation for gas-hydrate sediments. SPE J. 18(02), 196–206 (2013)

Moridis, G.J.: User’s Manual of the Meshmaker v1.5 Code: A Mesh Generator for Domain Discretization in Simulations of the Tough+ and Tough2 Families of Codes. Technical Report LBNL-1005134, Lawrence Berkeley National Laboratory (2016)

Moridis, G.J., Kowalsky, M.B., Pruess, K.: Tough+Hydrate v1.0 User’s Manual: A Code for the Simulation of System Behavior in Hydrate-Bearing Geologic Media. Technical Report LBNL-0149E, Lawrence Berkeley National Laboratory (2008)

Nishida, A.: Experience in developing an open source scalable software infrastructure in Japan. In: Computational Science and Its Applications—ICCSA 2010. Lecture Notes in Computer Science 6017 (2010)

Queiruga, A.F.: Cornflakes: first public release. https://doi.org/10.5281/zenodo.1166194 (February 2018a)

Queiruga, A.F.: Popcorn: first public release. https://doi.org/10.5281/zenodo.1166208 (February 2018b)

Queiruga, A.F., Moridis, G.: Numerical experiments on the convergence properties of state-based peridynamic laws and influence functions in two-dimensional problems. Comput. Methods Appl. Mech. Eng. 322, 97–122 (2017)

Queiruga, A.F., Reagan, M.T.: tough_convert: version 1.0, February 2018. https://doi.org/10.5281/zenodo.1164418

Queiruga, A.F.: Microscale Simulation of the Mechanical and Electromagnetic Behavior of Textiles. Ph.D. Thesis, University of California, Berkeley (2015)

Rutqvist, J., Moridis, G.J.: Numerical studies on the geomechanical stability of hydrate-bearing sediments. SPE J. 14(2), 267–282 (2009). https://doi.org/10.2118/126129-PA

Rutqvist, J., Moridis, G.J., Grover, T., Collett, T.: Geomechanical response of permafrost-associated hydrate deposits to depressurization-induced gas production. J. Petrol. Sci. Eng. 67, 1–12 (2009). https://doi.org/10.1016/j.petrol.2009.02.013

SymPy Development Team: SymPy: Python library for symbolic mathematics, 2016. http://www.sympy.org

The MPI Forum: MPI: A Message-Passing Interface Standard. Technical report, Knoxville, TN, USA (1994)

Uchida, S., Soga, K., Yamamoto, K.: Critical state soil constitutive model for methane hydrate soil. J. Geophys. Res. Solid Earth 117(B3) (2012)

Uchida, S., Klar, A., Yamamoto, K.: Sand production model in gas hydrate-bearing sediments. Int. J. Rock Mech. Min. Sci. 86, 303–316 (2016). https://doi.org/10.1016/j.ijrmms.2016.04.009

Van der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Comput. Sci. Eng. 13(2), 22–30 (2011)

Verruijt, A.: Theory and Problems of Poroelasticity. Delft University of Technology, Delft (2013)

Waite, W.F., Jang, J., Collett, T.S., Kumar, P.: Downhole physical property-based description of a gas hydrate petroleum system in NGHP-02 Area C: a channel, levee, fan complex in the Krishna–Godavari Basin offshore eastern India. Mar. Petrol. Geol. 2018. https://doi.org/10.1016/j.marpetgeo.2018.05.021. http://www.sciencedirect.com/science/article/pii/S0264817218302241

Yang, D., Moridis, G.J., Blasingame, T.A.: A fully coupled multiphase flow and geomechanics solver for highly heterogeneous porous media. J. Comput. Appl. Math. 270, 417–432 (2014)