Cite this article
Chen Q and Lai Z
Hydromechanical modelling of CO_2 sequestration using a component-based multiphysics code.
Environmental Geotechnics,
https://doi.org/10.1680/jenge.18.00038

Research Article
Paper 1800038
Received 09/02/2018; Accepted 06/12/2018

ICE Publishing: All rights reserved

Keywords: energy geotechnics/
geoenvironment/numerical methods

Environmental Geotechnics

ice Publishing

Hydromechanical modelling of CO_2 sequestration using a component-based multiphysics code

1 **Qiushi Chen** PhD
Assistant Professor, Glenn Department of Civil Engineering,
Clemson University, Clemson, SC, USA (corresponding author:
qiushi@clemson.edu) (Orcid:0000-0002-0394-6710)

2 **Zhengshou Lai** PhD
Research Assistant, Glenn Department of Civil Engineering, Clemson
University, Clemson, SC, USA (Orcid:0000-0002-2378-9193)

Geologic carbon dioxide sequestration (GCS) is a complex process with coupled multiphysics mechanisms, including non-linear solid deformations and fluid flow. When applying a finite-element method to solve such a complex problem, implementing and solving non-linear partial differential equations with sophisticated material models pose great challenges. In this work, a generic component-based multiphysics analysis code – Albany – is introduced for the modelling and analysis of GCS problems. The component-based approach allows the application code development effort to focus on writing physics models. A notable feature of Albany is the powerful automatic differentiation utilities that can be used to calculate automatically system Jacobians and consistent tangents. Two numerical problems, namely, wellbore injection and carbon dioxide transport in a confined underground channel, are developed to demonstrate the performance of the Albany analysis framework. The results are verified against and matched well with those from Comsol – a robust commercial multiphysics finite-element package. The numerical results are also thoroughly analysed to gain insights into distribution profiles and temporal evolutions of pore pressure and displacement fields during a GCS process.

Notation

B	Biot coefficient
$\boldsymbol{b}$	left Cauchy–Green tensor
$\boldsymbol{D}$	constitutive tangent modulus
$\boldsymbol{F}$	deformation gradient
$\boldsymbol{f}_{\mathrm{b}}$	body force
$\boldsymbol{g}$	gravitational acceleration
I	second-order identity matrix
J	determinant of deformation gradient (Jacobian)
K, K_{s}, K_{f}	bulk moduli of the skeleton, solid and fluid, respectively
$\boldsymbol{k}$	intrinsic permeability of the porous material
M	Biot modulus
$\boldsymbol{N}_{\mathrm{p}}$, $\boldsymbol{N}_{\mathrm{u}}$	shape functions for pore pressure and displacement, respectively
$\boldsymbol{n}$	unit normal vector to the boundaries
p	pore pressure
$\bar{p}$	prescribed pore pressure
$\tilde{\boldsymbol{p}}$, $\tilde{\boldsymbol{u}}$	nodal pore pressure and displacement vectors, respectively
$\bar{q}$	prescribed fluid flux
r	radius
$\bar{\boldsymbol{t}}$	prescribed traction
$\boldsymbol{u}$ (u_x, u_y, u_z)	displacement vector
$\bar{\boldsymbol{u}}$	prescribed displacement
W	strain energy
Γ	boundary condition
Γ_p, Γ_q	pressure and flux boundary conditions, respectively
Γ_u, Γ_t	displacement and traction boundary conditions, respectively
Δt	time increment
η, ψ	weight functions associated with the deformation and pore pressure, respectively
κ, μ	bulk and shear moduli of the porous material, respectively
μ^{f}	fluid dynamic viscosity
ρ, ρ^{s}, ρ^{f}	density of porous media, solid and fluid, respectively
σ, σ'	total and effective stresses, respectively
ϕ	porosity
χ	unknown variable to be solved
Ω	domain of interest of a boundary value problem

Introduction

Geologic carbon dioxide (CO_2) sequestration (GCS) is a process where large volumes of carbon dioxide are injected in deep geologic formations and is considered one of the more promising approaches to greenhouse gas control (Haszeldine, 2009; Lackner, 2003; Orr, 2009). The process of GCS involves various health, safety and environmental risks (Damen *et al*., 2006; Pan *et al*., 2011), and its success necessitates the fundamental understanding of the injection and migration process of carbon dioxide in storage aquifers and the response of surrounding geologic formations. Numerical tools, such as the finite-element method, have been widely used to provide insight into and to predict the performance of a GCS process (e.g. Azadi *et al*., 2016; Bao *et al*., 2014, 2016; Celia and Nordbotten, 2009; Cholewinski and Leonenko, 2018; Douglas *et al*., 2010; Fu *et al*., 2017; Harp *et al*., 2014; Xu *et al*., 2004, 2007).

The numerical modelling of GCS, however, remains a difficult endeavour. On the one hand, the conservation equations for the multiphysics problem of GCS are highly coupled. Even a simple numerical model for GCS is formulated as a two-way coupled diffusion–deformation problem, where the deformation of the solid may trigger fluid flow within the porous media, while the transient nature of fluid diffusion may induce excess pore pressure, limit volumetric deformation and introduce rate dependence to the mechanical response of the solid skeleton (Sun *et al*., 2014). A more sophisticated numerical model may need to consider multiphase fluid flow, mass diffusion and exchange between solid and fluid phases, non-isothermal conditions and chemical reactions, even fracturing of the geologic formation (Fu *et al*., 2017). On the other hand, capturing the complex and non-linear behaviour of the porous media often means sophisticated constitutive models need be developed and implemented (e.g. Fossum and Fredrich, 2000; Foster *et al*., 2005; Sun *et al*., 2014), which poses challenges such as deriving closed-form consistent tangent moduli for linearisation and solution of a global non-linear system of equations. These challenges in the numerical modelling of GCS call for an efficient and robust platform for finite-element code development, motivating the work presented in this paper.

Developed at Sandia National Laboratories by a team of researchers across multiple departments, Albany is an emerging powerful and open-source platform for finite-element-method-based solutions of tightly coupled partial differential equations (Salinger *et al*., 2016). The most salient feature of Albany, compared to those of conventional finite-element codes, is its component-based generic programming approach, which makes broad use of template objectives, computational science libraries, abstract interfaces and software engineering tools (Salinger *et al*., 2016). It allows finite-element code developers to focus on the derivation and implementation of the weak form of conservation equations (e.g. residual equations for the balance of momentum and mass), while having access to algorithm libraries developed by subject matter experts. Derivatives and polynomial propagations associated with the residual equations are computed automatically. This feature greatly benefits the testing of new physics and new discretisation. Another key feature of Albany is the exploitation of the automatic differentiation (AD) technique (Chen *et al*., 2014a; Pawlowski *et al*., 2012a, 2012b). In Albany, there is no need to derive Jacobian matrices or analytical consistent tangent moduli manually; instead, they are computed automatically through the AD technique. The values of these matrices computed using AD are exact, rather than numerical approximations that one would obtain if a finite-difference approach were used (Miehe, 1996). In addition, Albany employs message passing interface and open multi-processing for parallel computation, which supports generic multicore computing across a variety of platforms and is palatable to various software engineering tools such as Dakota (Adams *et al*., 2009) for uncertainty quantification and inverse problems.

The purpose of this paper is twofold. First, it introduces Albany code to the numerical modelling of GCS problem. The features of Albany will be highlighted in the formulation and implementation of hydromechanical conservation equations and a material constitutive model for a basic GCS problem. Second, the performance of Albany for large-scale simulations of wellbore injection and carbon dioxide sequestration will be demonstrated based on two numerical examples, and results are verified against a robust commercial multiphysics code. Numerical issues such as mesh sensitivity and pore pressure oscillations of the hydromechanical problem are identified. It should be noted that the model currently implemented in Albany still contains a number of key simplifications and limitations, particularly with respect to single-phase fluid flow, isothermal condition, no phase transition or chemical reaction, no mass exchange between the solid and fluid constituents and so on. Modelling tools that consider the complete physics of a GCS system under realistic field conditions do not yet exist, and substantial efforts are still needed for future development. Nevertheless, the work presented in this paper is a first step towards this direction. It is anticipated that the methodology and numerical example results presented in this work would shed light on the application of Albany for modelling more complex GCS problems.

Formulation of coupled hydromechanical problems

In this section, the formulation of coupled hydromechanical problems for porous materials, including conservation equations, constitutive relations and governing equations for the boundary value problem, is presented. Although the formulation is presented for a simple model for carbon dioxide sequestration simulations, capabilities of Albany for other complex multiphysics models have been demonstrated in a number of recent works (e.g. Roy *et al*., 2018; Sun *et al*., 2013, 2014; Tezaur *et al*., 2015).

Conservation equations

The degrees of freedom (DOFs) for coupled hydromechanical problems include the deformation of the solid phase, $\boldsymbol{u}$, and the pore pressure of the fluid phase, p. Following Terzaghi's effective stress theory, the total stress σ consists of two components, the effective stress, σ', and the pore pressure, p. The effective stress comes from solid grain-to-grain contacts and dictates the deformation of the porous media. The total stress equation is written as

$$\sigma = \sigma' - Bp\mathbf{I} \tag{1}$$

where $\mathbf{I}$ is the second-order identity matrix and the scalar parameter B is the Biot coefficient that can be calculated from (Nur and Byerlee, 1971; Skempton, 1954)

$$B = 1 - \frac{K}{K_\mathrm{s}} \tag{2}$$

where K and K_s are the bulk moduli of the skeleton and solid grain, respectively.

Neglecting the acceleration for both solid and fluid phases, the linear momentum and mass balance equations under quasi-static condition are written as

$$\nabla\cdot\left(\sigma' - Bp\mathbf{I}\right) + \boldsymbol{f}_\mathrm{b} = \mathbf{0} \tag{3}$$

$$-B\nabla\cdot\dot{\boldsymbol{u}} - \frac{1}{M}\dot{p} + \nabla\cdot\frac{\boldsymbol{k}}{\mu^\mathrm{f}}\left(\nabla p + \rho^\mathrm{f}\boldsymbol{g}\right) = 0 \tag{4}$$

where $\boldsymbol{f}_\mathrm{b}$ is the body force, M is the Biot modulus, $\boldsymbol{k}$ is the intrinsic permeability of the porous media, μ^f is the fluid dynamic viscosity, ρ^f is the density of the pore fluid and $\boldsymbol{g}$ is the gravitational acceleration. Just considering the gravitational body force, the body force of the porous media can be calculated from

$$\boldsymbol{f}_\mathrm{b} = \rho\boldsymbol{g} = (1-\phi)\rho^\mathrm{s}\boldsymbol{g} + \phi\rho^\mathrm{f}\boldsymbol{g} \tag{5}$$

where ϕ is the porosity and ρ^s is the density of the solid. The Biot modulus M, accounting for the compressibility of both solid and pore fluid, is given by (Biot, 1941)

$$M = \frac{1}{(B-\phi)/K_\mathrm{s} + \phi/K_\mathrm{f}} \tag{6}$$

where K_f is the bulk modulus of the pore fluid. In presenting the conservation equations, it is assumed that there is no phase change of the fluid or mass exchange between the fluid and solid constituents, and, therefore, no energy conservation or mass diffusion equation is included in the present model.

Constitutive relations

In this work, a large-deformation material model is adopted to describe the non-linear stress–strain behaviour of the solid. The starting point of deriving the constitutive relations is a stored-energy function W written in terms of the deformation gradient $\boldsymbol{F}$, written as

$$W = \frac{\kappa}{2}\left[\frac{1}{2}\left(J^2 - 1\right) - \log(J)\right] + \frac{\mu}{2}\left[J^{-2/3}\mathrm{tr}(\boldsymbol{b}) - 3\right] \tag{7}$$

where κ and μ are the bulk and shear moduli of the material, respectively; $J = \det(\boldsymbol{F})$ is the determinant of the deformation gradient; $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^\mathrm{T}$ is the left Cauchy–Green tensor; and $\mathrm{tr}(\cdot)$ is the trace operator. This stored-energy function was considered by Simo and Hughes (1998) to illustrate formulations of large-deformation plasticity models based on the notion of an intermediate stress-free configuration.

By taking the derivative of the energy function in Equation 7 with respect to the deformation gradient and pulling back to the reference configuration (Simo, 1988), the (effective) Cauchy stress tensor of the solid can be calculated as

$$\sigma' = \frac{\kappa}{2}\left(J - \frac{1}{J}\right)\mathbf{I} + \mu J^{-5/3}\mathrm{dev}(\boldsymbol{b}) \tag{8}$$

In addition to calculating the effective stress, the consistent tangent moduli, which give the sensitivity of the algorithmic expression for the stresses with respect to the change in total deformation, are needed to achieve an optimal convergence rate in the global solution of the non-linear system of equations (Borja, 2013; Simo and Hughes, 1998). It should be noted that a closed-form analytical tangent can be extremely difficult to obtain, even for this relatively simple material model. In terms of numerical computation of the consistent tangent, previous work typically uses forward difference approximations (Miehe, 1996). In this work, the AD technique available in Albany will be introduced to obtain numerically the exact sensitivity information in linearisation and solution of global boundary value problems, thus freeing a constitutive model developer from deriving and implementing a closed-form consistent tangent. The AD technique will be detailed in a later section.

Boundary value problem and the weak form

Here, the domain of interest of the boundary value problem is denoted as Ω with its boundary Γ. For the deformation field, Γ_u and Γ_t are used herein to denote the displacement and traction boundary conditions, respectively, where Γ_u and Γ_t are complementary sets of Γ, such that $\Gamma = \Gamma_u \cup \Gamma_t$. For the pore pressure field, the two types of boundaries are denoted as Γ_p and Γ_q, which furnish the pressure and flux boundary conditions, respectively. Similarly, Γ_p and Γ_q are also complementary sets, such that $\Gamma = \Gamma_p \cup \Gamma_q$. The boundary conditions are prescribed as

$$\begin{aligned} \boldsymbol{u} &= \bar{\boldsymbol{u}} \text{ on } \Gamma_u \\ \boldsymbol{n}\cdot\sigma &= \bar{\boldsymbol{t}} \text{ on } \Gamma_t \\ p &= \bar{p} \text{ on } \Gamma_p \\ -\boldsymbol{n}\cdot\boldsymbol{q} &= \bar{q} \text{ on } \Gamma_q \end{aligned} \tag{9}$$

where $\bar{\boldsymbol{u}}$, $\bar{\boldsymbol{t}}$, $\bar{p}$ and $\bar{q}$ are the prescribed values at corresponding boundaries – that is, $\bar{\boldsymbol{u}}$ is the prescribed displacement, $\bar{\boldsymbol{t}}$ is the prescribed traction, $\bar{p}$ is the prescribed pore pressure and $\bar{q}$ is the prescribed fluid flux. $\boldsymbol{n}$ is the unit normal vector to the boundaries.

Introducing the weight functions for both the deformation and pore pressure fields, weak forms of governing equations (i.e. the linear momentum and mass balance equations) of the hydromechanical problem are given as

$$\int_\Omega \nabla\eta\colon \left(\sigma' - Bp\mathbf{I}\right)\mathrm{d}\Omega - \int_\Omega \eta\cdot\boldsymbol{b}\,\mathrm{d}\Omega - \int_\Gamma \eta\cdot\bar{\boldsymbol{t}}\,\mathrm{d}\Gamma = 0 \tag{10}$$

$$\int_\Omega \psi\left(B\nabla\cdot\dot{\boldsymbol{u}} + \frac{1}{M}\dot{p}\right)\mathrm{d}\Omega + \int_\Omega \nabla\psi\cdot\frac{\boldsymbol{k}}{\mu^{\mathrm{f}}}\left(\nabla p + \rho^{\mathrm{f}}\boldsymbol{g}\right)\mathrm{d}\Omega - \int_\Gamma \psi\bar{q}\,\mathrm{d}\Gamma = 0 \tag{11}$$

where η and ψ are the weight functions associated with the deformation and pore pressure, respectively, and all the other variables have been introduced previously.

Galerkin approximation and matrix equations

To obtain the matrix form of the hydromechanical problem, the following approximations are introduced to trial solutions $\boldsymbol{u}$ and p such that

$$\boldsymbol{u} = \boldsymbol{N}_u\tilde{\boldsymbol{u}} \tag{12}$$

$$p = \boldsymbol{N}_p\tilde{\boldsymbol{p}} \tag{13}$$

where $\boldsymbol{N}_u$ and $\boldsymbol{N}_p$ are shape functions for displacement and pore pressure, respectively, and $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{p}}$ are the nodal displacement and pore pressure vectors, respectively. The same approximation for the weight functions and the trial solutions is used – that is, the standard Galerkin method.

Substituting Equations 12 and 13 and the approximation of weight functions into the weak form Equations 10 and 11, the global finite-element matrix equations of the hydromechanical problem are written as

$$\mathbf{K}\tilde{\boldsymbol{u}} - \mathbf{M}\tilde{\boldsymbol{p}} - \mathbf{F} = 0 \tag{14}$$

$$\mathbf{M}^T\dot{\tilde{\boldsymbol{u}}} + \mathbf{S}\dot{\tilde{\boldsymbol{p}}} + \mathbf{H}\tilde{\boldsymbol{p}} - \mathbf{Q} = 0 \tag{15}$$

where $\mathbf{K} = \int_\Omega \nabla\boldsymbol{N}_u^T\boldsymbol{D}\nabla\boldsymbol{N}_u\,\mathrm{d}\Omega$, $\mathbf{M} = B\int_\Omega \nabla\boldsymbol{N}_u^T\mathbf{I}\boldsymbol{N}_p\,\mathrm{d}\Omega$, $\mathbf{F} = \int_\Omega \boldsymbol{N}_u^T\boldsymbol{b}\,\mathrm{d}\Omega + \int_\Gamma \boldsymbol{N}_u^T\bar{\boldsymbol{t}}\,\mathrm{d}\Gamma$, $\mathbf{S} = (1/M)\int_\Omega \boldsymbol{N}_p^T\boldsymbol{N}_p\,\mathrm{d}\Omega$, $\mathbf{H} = (1/\mu^{\mathrm{f}})\int_\Omega \nabla\boldsymbol{N}_p^T\boldsymbol{k}\nabla\boldsymbol{N}_p\,\mathrm{d}\Omega$ and $\mathbf{Q} = \int_\Gamma \boldsymbol{N}_p^T\bar{\boldsymbol{q}}\,\mathrm{d}\Gamma - (1/\mu^{\mathrm{f}})\rho^{\mathrm{f}}\boldsymbol{g}\int_\Omega \nabla\boldsymbol{N}_p^T\boldsymbol{k}\,\mathrm{d}\Omega$, where $\boldsymbol{D}$ is the constitutive tangent modulus. Equations 14 and 15 can be written in a compact form, as

$$\begin{bmatrix} 0 & 0 \\ \mathbf{M}^T & \mathbf{S} \end{bmatrix}\begin{bmatrix} \dot{\tilde{\boldsymbol{u}}} \\ \dot{\tilde{\boldsymbol{p}}} \end{bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{M} \\ 0 & \mathbf{H} \end{bmatrix}\begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{p}} \end{bmatrix} - \begin{bmatrix} \mathbf{F} \\ \mathbf{Q} \end{bmatrix} = 0 \tag{16}$$

In Albany, an equal-order displacement–pore pressure mixed finite-element formulation is implemented. A key advantage of such formulation is that it is computationally more efficient. Also, it does not require significant modification to data structures of Albany, except that one needs to add an additional DOF to all nodes for the pore pressure field. The equal-order mixed finite-element method, however, must be stabilised by either adding additional terms or introducing enhanced shape functions to eliminate the spurious mode encountered as a result of failure to satisfy the inf–sup condition (Bathe, 2001). The stabilised mixed finite-element formulation proposed by Sun *et al.* (2013) is adopted in Albany, which stabilises the equal-order poromechanics formulation at finite strain, while simultaneously considering the volumetric locking effect and the compressibility of both solid and fluid constituents.

Time integration of the governing equations

In lieu of the governing equations expressed in Equation 16, the following abstracted time-dependent problem is used to describe the time integration scheme implemented in Albany

$$\mathbf{r}(\dot{\chi}, \chi) = \mathbf{A}\dot{\chi} + \mathbf{B}\chi + \mathbf{C} = 0 \tag{17}$$

where χ is the unknown variable to be solved

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ \mathbf{M}^T & \mathbf{S} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{K} & -\mathbf{M} \\ 0 & \mathbf{H} \end{bmatrix} \quad \mathbf{C} = -\begin{bmatrix} \mathbf{F} \\ \mathbf{Q} \end{bmatrix} \quad \chi = \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\boldsymbol{p}} \end{bmatrix} \tag{18}$$

and $\dot{\chi}$ denotes the time derivative of χ.

The unconditionally stable first-order backward Euler method is adopted for time derivatives of the primary variable, expressed as

19. $$\dot{\chi}^{t+1} = \frac{1}{\Delta t}\left(\chi^{t+1} - \chi^{t}\right)$$

where Δt is the time increment and $[\cdot]^t$ denotes the variable value at time t. Substituting Equation 19 into Equation 17, the following generalised linear system is obtained

20. $$\mathbf{f}\left(\chi^{t+1}\right) = \left(\frac{1}{\Delta t}\mathbf{A} + \mathbf{B}\right)\chi^{t+1} + \mathbf{C} - \frac{1}{\Delta t}\mathbf{A}\chi^{t} = 0$$

The generalised linear system can be solved through a variety of direct and iterative algorithms, linear solvers and preconditioners available in Albany through its extensive use of external software libraries.

Albany: a component-based generic multiphysics analysis code

The previously described governing equations and constitutive relations are implemented within Albany (Salinger *et al.*, 2016) – an open-source project developed at Sandia National Laboratories to demonstrate potentials of the component-based design approach to finite-element code development. The Albany source code is available on GitHub (SNL, 2018). In this section, the component-based generic code design approach is first introduced. Then, a key feature of Albany – that is, its ability to calculate parameter derivatives and assemble Jacobian matrices automatically through AD is discussed in detail.

Component-based generic code design approach

Component-based development is the process by which an application code is built primarily from modular pieces such as independently developed software libraries. Figure 1 shows the component libraries and abstract interfaces used to build Albany. Many of these components come from the Trilinos (Heroux *et al.*, 2005) multiphysics code development framework (Trilinos Project, 2018).

At a high level, the Albany code consists of five main generic components that serve the computation of five main algorithmic domains – namely, global discretisation, problem abstraction and finite-element assembly, non-linear model, linear solver and analysis tools. For each algorithmic domain, Albany provides the abstraction interface that specifies the inputs and outputs required in this algorithmic domain. The abstraction interfaces facilitate the reuse of existing mature libraries for each algorithmic domain in a finite-element analysis, thus, to an extreme extent, mitigating the coding efforts for a finite-element problem. The functionalities of these five main algorithmic domains are briefly described as follows.

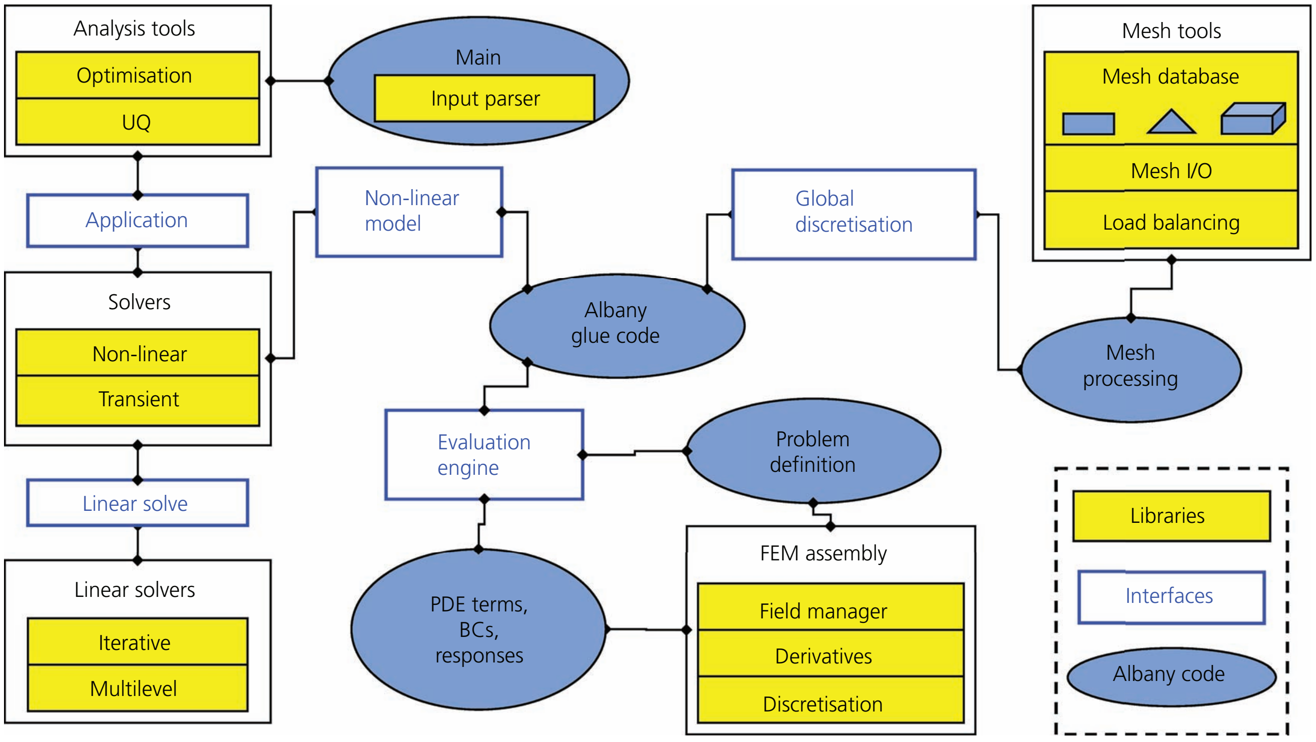


Figure 1. Component libraries and abstract interfaces used to build Albany (adapted from Salinger *et al.* (2016)). BCs, boundary conditions; FEM, finite-element method; I/O, input/output; PDE, partial differential equations; UQ, uncertainty quantification

- Global discretisation: it gives the finite-element assembly process access to all of the data distribution information. It takes a mesh database as inputs and provides quantities in the layout required in a finite-element analysis, such as coordinates, solution vectors, sparse-matrix graphs and DOF numbering/connectivity information. Albany currently supports two discretisation interfaces (libraries): (*a*) the Sierra Toolkit package (Edwards *et al*., 2010) and (*b*) the Parallel Unstructured Mesh Interface (Seol *et al*., 2012).
- Problem abstraction and finite-element assembly: its purpose is to evaluate the discrete finite-element residual and derived quantities such as the Jacobian matrices and parameter derivatives needed for simulation analysis. In this domain, a given multiphysics problem (coded by the developer in the form of residuals) is generalised into physics-independent algebraic problem abstractly represented by Equation 17. A template-based generic programming approach (Pawlowski *et al*., 2012a, 2012b) is leveraged to implement the gather and scatter phases of the finite-element assembly. The AD technique (supported by the Sacado package (Phipps, 2015) in Trilinos) is adopted by Albany to compute automatically the derived quantities such as Jacobian matrices and parameter derivatives.
- Non-linear model: it facilitates the use of the general-purpose solution and analysis algorithms. More specifically, the component aims to satisfy the abstraction of the residuals obtained in the previous domain into inputs (e.g. the solution and time derivatives) and outputs (e.g. residual and Jacobian) required in a non-linear solver. The interface thus allows Albany to access the various solver methods (e.g. NOX, LOCA, Rythmos or Stokhos packages) embedded in Trilinos.
- Linear solver: the linear solver interface is embedded in the non-linear model algorithmic domain. It provides Albany full access to a wide assortment of direct and iterative algorithms, linear solves and preconditioners available in Trilinos.
- Analysis tools: these tools are present at the top level of the software stack and are used to perform a single forward solve, sensitivity analysis, parameter studies, optimisation and so on. The interface allows Albany to access various analysis tools, such as the Dakota toolkit (Adams *et al*., 2009) for uncertainty quantification analysis.

From the perspective of an application developer, a major strength of Albany's component-based code design approach is the ease in which coupled systems of partial differential equations can be implemented. The developer can focus on the physics (e.g. the residual equations and constitutive models) of the problem while having access to algorithm libraries developed by subject matter experts. Moreover, through template-based generic programming methods (Pawlowski *et al*., 2012a, 2012b), another key strength of Albany is that it transforms automatically the code for evaluating the residual into the code for computing Jacobians and parameter derivatives through overloading-based AD, detailed in the following section.

Automatic differentiation

AD is a well-known set of techniques in computational science that transforms a given computation implemented in a programming language into one that computes derivatives of that computation (Pawlowski *et al*., 2012a). For completeness, this section presents major elements of AD and demonstrates how AD is utilised in the implementation of Albany through operator overloading. The interested reader is referred to the book of Griewank and Walther (2008) for an in-depth introduction to AD and to the papers of Pawlowski *et al*. (2012a, 2012b) for details on operator overloading.

Albany utilises operator overloading to implement the AD technique. Briefly, in operator overloading, new data types (special templates) are created to store not only the variable values, but also their derivative values. Overloaded versions of the arithmetic and maths functions are provided that implement the derivatives of those functions operating on these new data types (Pawlowski *et al*., 2012a).

To illustrate the AD technique and operator overloading, consider an example problem with N independent variables $[x_1, x_2,\ldots,x_N]$ and some maths functions defined in Figure 2. For the first function ($y = x_1$), once the templated object representing the function is instantiated, it will be stored as [3, 1, 0,…,0] (a total of $N + 1$ values), where '3' is the scalar value of the function (a value of 3 is used as an example), '1' is the derivative of the function ($y = x_1$) with respect to the first independent variable (x_1) and all the '0s' are the derivatives of the function with respect to other independent variables $[x_2, x_3,\ldots,x_N]$. Other example functions in the figure can be understood the same way. When an arithmetic operator performs on a templated object, the entire array of values is updated based on their specific form. No matter how complex a function may be, the derivatives can always be discretised into a chain of basic arithmetic operations. Therefore, all the derivatives can be calculated and stored in the array. When needed, they can be easily extracted and gathered into the finite-element assembly and used in quantities such as Jacobin matrices (for global Newton solve), parameter derivatives (for optimisation) or polynomial chaos expansions (for uncertainty quantification).

For the multiphysics problem considered in this paper, the independent variables are displacement and pore pressure DOF and the weak forms (Equations 10 and 11) are functions of these independent variables. In the Albany implementation, these variables are defined using the templated new data types. Then, derivatives are automatically computed and stored through the aforementioned AD and operator overloading approach. These derivatives can be used to assemble the global Jacobian matrix when solving the global non-linear system of equations through iterative schemes such as the Newton–Raphson method.

Another important use of the AD technique in Albany is in constitutive model implementation. When integrating the rate

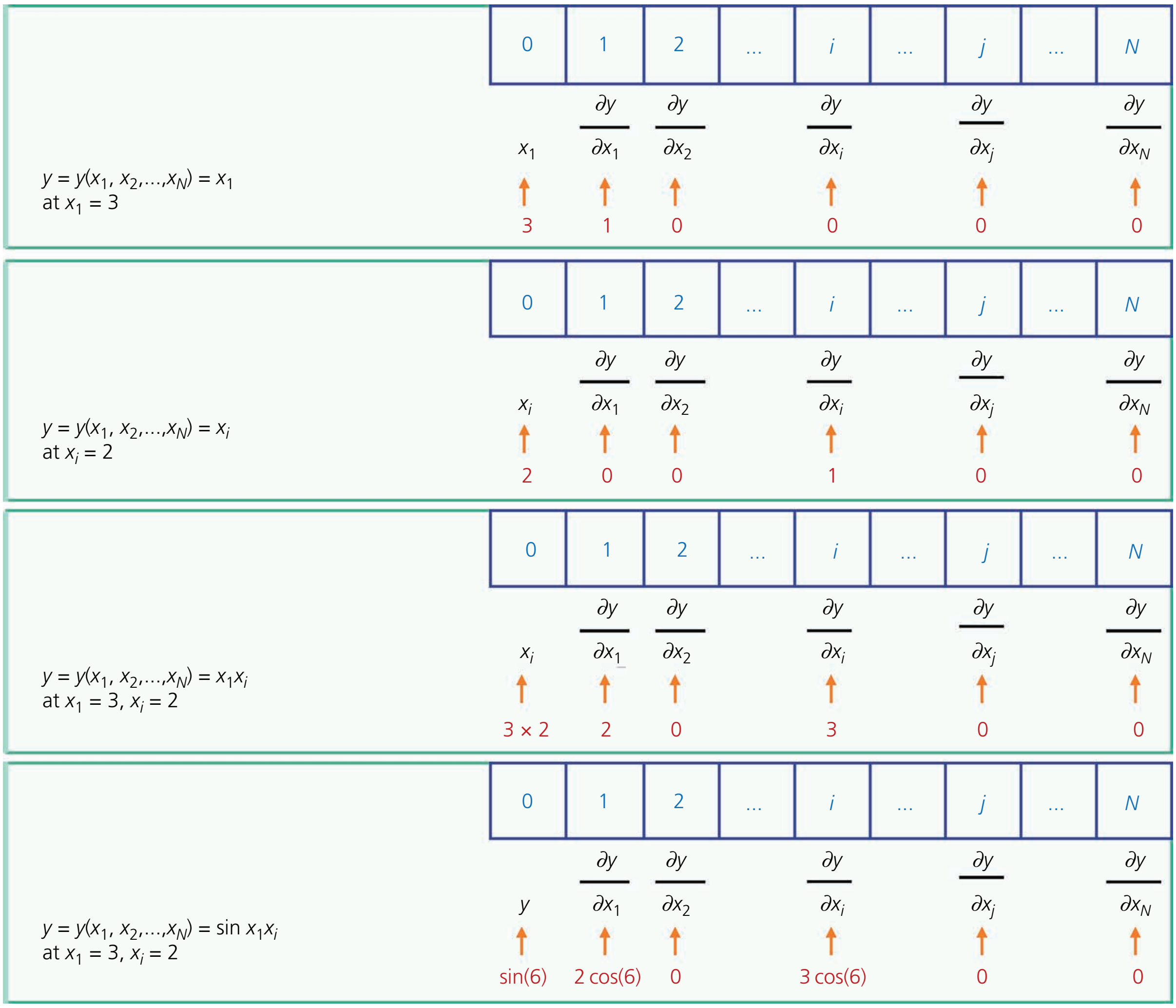


Figure 2. Illustration of templated objects and store of derivative information

form of constitutive equations, one may employ implicit, explicit or semi-implicit methods (e.g. Simo and Hughes, 1998; Sloan *et al.*, 2001; Tu *et al.*, 2009). This often results in a set of discrete equations for updating stresses and state-dependent internal variables. In many cases, the resulting equations are highly non-linear and require iterative solution methods such as Newton's method, where computations of derivatives are necessary to assemble the local Jacobian matrix and to achieve an optimal asymptotic convergence rate. At the constitutive model level, the independent variables are the unknown stress and internal state variables, while the residual equations are the time-discretised constitutive equations (Chen *et al.*, 2014a). Once the residual functions are formulated, the local Jacobian matrix can be easily assembled by extracting derivatives obtained by the AD technique.

In addition to providing sensitivity information for local iterative solve, the AD technique in Albany has also been applied to integrate hyperelastic constitutive relationships, where derivatives of the stored-energy function are calculated by the AD technique to obtain stress and tangent moduli. Specific applications of AD in Albany for the numerical integration of small- and large-deformation constitutive models can be found in the publications by Chen *et al.* (2014a, 2014b) and Sun *et al.* (2014). In the following example problems, constitutive equations of a non-linear material model are derived from a stored-energy function and the AD technique is used to its derivatives and to obtain stress and tangent moduli.

Numerical examples

In this section, two numerical examples are presented to demonstrate the capability of Albany in simulating coupled hydromechanical problems of interest to GCS. The first example is a wellbore injection problem reported by Murdoch *et al.* (2016). The second example considers the carbon dioxide transportation process through a tiny channel surrounded by impermeable formations. As a moderate degree of verification, a commercial finite-element package (Comsol Multiphysics) is used to set-up and simulate both the wellbore injection and the carbon dioxide transportation problems. In the report of Murdoch *et al*. (2016), a number of boundary value problems of interest to GCS, including the wellbore injection problem, have been simulated using Comsol and compared very well with results by Abaqus, which is another robust and commonly used commercial finite-element package. Comsol is best known for its multiphysics modelling capability and is selected in this work to verify results from Albany. Also, it should be noted that all pore pressure results reported in this section refer to excess pore pressure.

Wellbore injection simulation

The wellbore injection simulation is a typical and commonly seen example in the literature as it represents a common situation encountered in a GCS project. The simulation considers the fluid flow and solid deformation in the vicinity of an injecting wellbore that reaches an underground storage formation.

Model domain, boundary conditions and material properties

The domain of interest is a geologic formation consisting of three distinct layers, namely, a 1000 m shale layer on the top, a 100 m sandstone layer in the middle and another 100 m shale layer at the bottom, as shown in Figure 3. A cylindrical domain is considered in the model with the injection well located at the centre. The sandstone layer is of high permeability, which allows the injection and transportation of carbon dioxide, whereas the top and bottom shale layers are much less permeable and confine (sequestrate) the injected carbon dioxide. The radius of the injection well is 0·1 m, and the radius of the formation is 30 000 m. This order of difference in size poses challenges to numerical models and would require an adaptive refinement of the finite-element mesh.

Due to the symmetry of the problem, only one-quarter of the cylindrical domain is modelled. The meshing scheme for the injection problem is shown in Figure 4, where an extremely fine discretisation is used near the injection well and the discretisation gradually coarsens away from the wellbore. An equal-order displacement–pressure eight-node hexahedron finite-element with stabilisation (Sun *et al*., 2013) is used in the Albany model. The boundary conditions are summarised in Table 1. Material properties used in this example are summarised in Table 2. It should be noted that the bulk moduli of the shale and sandstone solid (K_s) are chosen such that they are significantly higher than the drained bulk modulus (K). Carbon dioxide is injected into the sandstone layer through the wellbore, and the duration of the injection is 10^6 s.

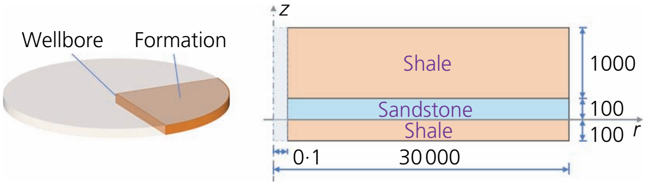


Figure 3. Sketch of the model geometry in the wellbore injection problem (unit: metres). The boundary conditions are summarised in Table 1

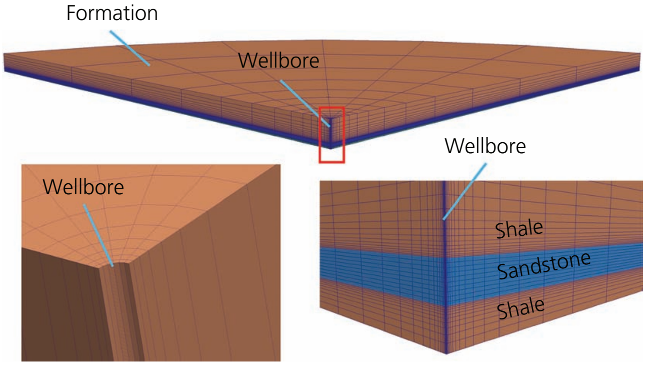


Figure 4. Meshing scheme for the wellbore injection model. The bottom right zoomed-in figure shows the refinement of the mesh around the well and at interfaces between different layers. The bottom left zoomed-in figure shows the mesh around the injection well

Profiles of pore pressure and deformation

The pore pressure and displacement fields are extracted from the simulation results to understand the characteristics of the flow and deformation in the vicinity of the injection well. Figure 5 displays the contour of the pore pressure in the sandstone and shale layers at the end of the injection process (10^6 s). It is observed that the pore pressure remains basically zero in the shale layers due to its very low permeability. In the sandstone layer, the pressure is the highest in the vicinity of the injection well and gradually vanishes away from the injection well (its value at a distance of 3000 m from the injection point is about one-thousandth of that at the injection well). These pore pressure profiles indicate that carbon dioxide is trapped in the sandstone layer by the confining shale formations, a desirable situation for a GCS project.

As the model is axisymmetric, the displacements in the radial direction are used to characterise the horizontal deformation of the formation. Figure 6(a) shows the radial displacement of the formation at the end of injection. In the sandstone layer, the radial displacement first increases and then decreases as the distance to the injection point increases. The largest radial displacement is about 0·49 mm located at 500 m away from the injection point. The top and bottom shale layers also exhibit slight, but noticeable, radial expansions.

The vertical displacement contour at the end of injection is shown in Figure 6(b). The top of sandstone layer exhibits a

Table 1. Boundary conditions of the wellbore injection problem

Boundary	Solid phase	Fluid phase
Top $z = 1100$ m	Free $\sigma_{zz} = 0$, $\sigma_{rz} = 0$	Pore pressure $p = 0$
Bottom $z = -100$ m	Roller $u_z = 0$, $\sigma_{rz} = 0$	Impermeable
Outer $r = 30\,000$ m	Roller $u_r = 0$, $\sigma_{rz} = 0$	Pore pressure $p = 0$
Inner $r = 0{\cdot}1$ m		
Shale -100 m $< z < 0$ m	Roller $u_r = 0$, $\sigma_{rz} = 0$	Impermeable
Sandstone 0 m $< z < 100$ m	Pressure $\sigma_{rr} = 1$ MPa, $\sigma_{rz} = 0$	Pore pressure $p = 1$ MPa
Shale 100 m $< z < 1100$ m	Free $\sigma_{rr} = 0$, $\sigma_{rz} = 0$	Impermeable

Table 2. Material properties of the sandstone, shale and fluid used in the wellbore injection problem

Parameter	Description	Value	Unit	Reference
Shale				
K_s	Bulk modulus of shale solid	$5{\cdot}0 \times 10^{13}$	Pa	Sun *et al.* (2013)
k	Intrinsic permeability of shale	$1{\cdot}0 \times 10^{-20}$	m^2	Fu *et al.* (2017)
ϕ	Porosity of shale	0·05	—	Fu *et al.* (2017)
K, κ	Bulk moduli of shale	$6{\cdot}7 \times 10^{9}$	Pa	Fu *et al.* (2017)
μ	Shear modulus of shale	$4{\cdot}0 \times 10^{9}$	Pa	Fu *et al.* (2017)
Sandstone				
K_s	Bulk modulus of sandstone solid	$5{\cdot}0 \times 10^{13}$	Pa	Sun *et al.* (2013)
k	Intrinsic permeability of sandstone	$1{\cdot}5 \times 10^{-14}$	m^2	Fu *et al.* (2017)
ϕ	Porosity of sandstone	0·15	—	Fu *et al.* (2017)
K, κ	Bulk moduli of sandstone	$6{\cdot}7 \times 10^{9}$	Pa	Fu *et al.* (2017)
μ	Shear modulus of sandstone	$4{\cdot}0 \times 10^{9}$	Pa	Fu *et al.* (2017)
Fluid				
K_f	Bulk modulus of pore fluid	$1{\cdot}0 \times 10^{9}$	Pa	Vilarrasa *et al.* (2010)
μ^f	Dynamic viscosity	$6{\cdot}0 \times 10^{-5}$	Pa s	Pensado *et al.* (2008)
ρ^f	Density	725·7	kg/m^3	Pensado *et al.* (2008)

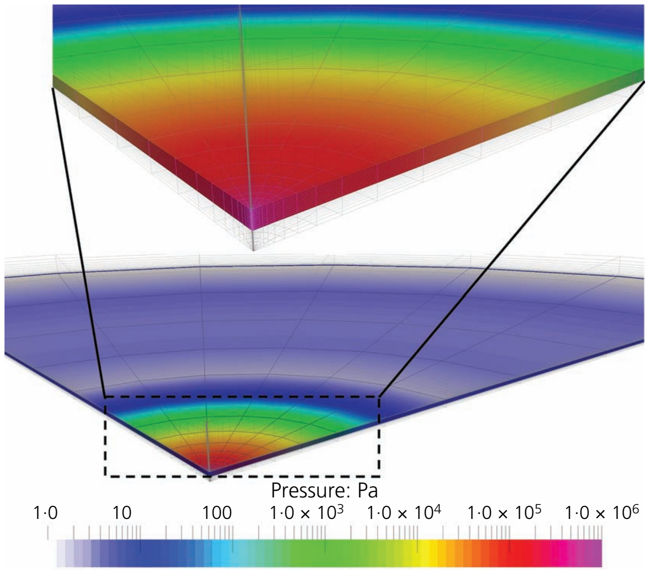


Figure 5. Contours of the pore pressure at the end of injection

positive vertical displacement, whereas the bottom exhibits a negative vertical displacement. The results indicate that there is a significant vertical expansion at the wellbore vicinity in the sandstone layer due to the injection pressure. The ground surface near the wellbore exhibits a positive vertical displacement, indicating that the surface also expands due to the injection pressure.

Verification of Albany accuracy

As previously mentioned, results obtained from Albany are compared and verified against those using the Comsol Multiphysics finite-element package. It should be noted that in Comsol an axisymmetric model is used to take advantage of the radial symmetry of the problem. Figure 7 shows results of pore pressure distribution along a radial line at the middle of the sandstone layer by both Albany and Comsol. It can be seen that the results from the two programs are in perfect agreement.

The vertical and radial displacements of two radial lines are extracted to characterise the deformation of the whole formation. The first radial line is in the middle of the sandstone layer, and the other one is at the top ground surface. The radial and vertical displacements along these two lines are shown in Figure 8, where, again, the results from Albany match very well with those from Comsol.

The vertical and radial displacements at the line along the wellbore are also extracted to gain insight into the deformation of

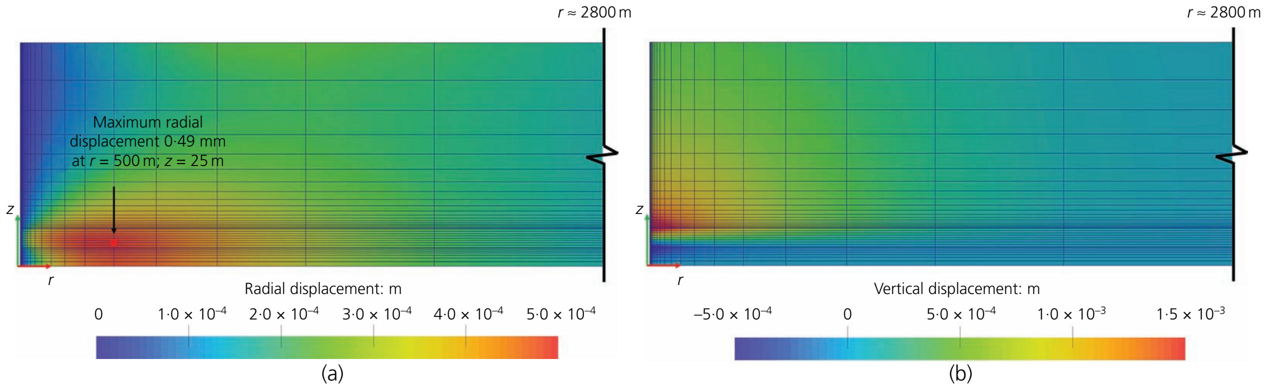


Figure 6. Contours of the (a) radial and (b) vertical displacements at the end of injection

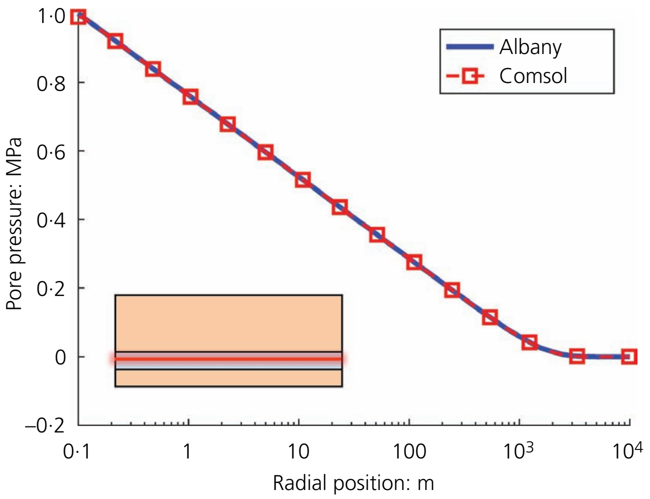


Figure 7. Pore pressure plotted against radial distance from the injection well along the centreline of the sandstone layer at the end of injection

the wellbore and its vicinity, and the results are presented in Figure 9. There are some slight differences in the results of vertical displacement (Figure 9(a)), particularly at the interfaces between the sandstone layer and the shale layers (i.e. vertical positions of 0 and 100 m). The model in Comsol is two-dimensional axisymmetric, while Albany uses a full three-dimensional model. Also, another possible cause could be the different discretisations, particularly near the injection well, between Albany and Comsol. This discretisation or mesh sensitivity issue will be investigated in detail in the next section.

Mesh sensitivity analysis

To gain insight into the mesh sensitivity issue of this particular example problem, different levels of mesh refinement (all with the same meshing scheme) are adopted to discretise the domain. Specifically, different numbers of intervals are applied to discretise the top shale layer in the vertical direction – that is, the number of the intervals n = 10, 15 and 20 in this case, as shown in Figure 10. It results in meshes with 6300, 7350 and 8400 total elements, respectively.

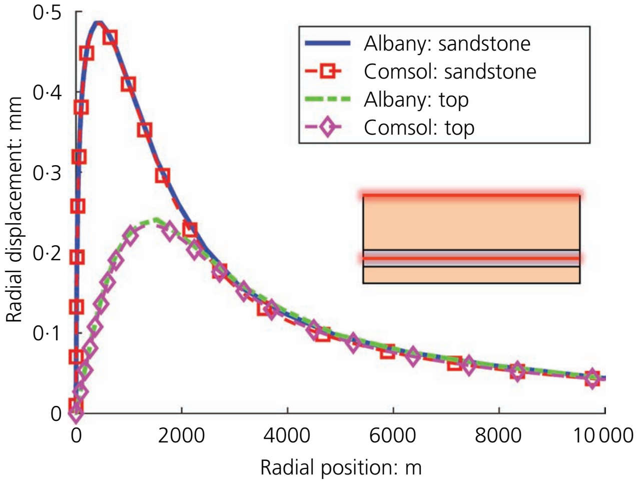
The vertical displacements along a vertical line in the wellbore and along a radial line at the surface are selected to show the effects of different discretisations, as plotted in Figure 11. Notice that the coarsest mesh results in a noticeable deviation from the benchmark results (i.e. results by Comsol). As the mesh density increases (i.e. n increases), the results from Albany gradually approach and converge to the benchmark results. It should be pointed out that a mesh sensitivity study using the same numbers of intervals for the shale layer is also performed in Comsol, where it is found that Comsol is much less sensitive to mesh sizes (at least for the number of intervals chosen). The Comsol results shown in Figure 11 are obtained using the finest mesh (i.e. n = 20).

Carbon dioxide transportation in a deep-buried permeable channel

In this example, the carbon dioxide transportation process in an underground permeable channel surrounded by impermeable formations is simulated. The permeable channel was meant to represent an old, buried riverbed. Because of the way the flow of the river sorts the sand grains, these are often quite permeable and might later become buried under a material that lithifies into a much less permeable materials, such as a shale formation. This long, permeable region might contain oil or natural gas, or, in the carbon dioxide sequestration application, it might be a good place for carbon dioxide injection and storage. In the analysis, in addition to pore pressure and deformation fields, the parallel performance of Albany will also be presented.

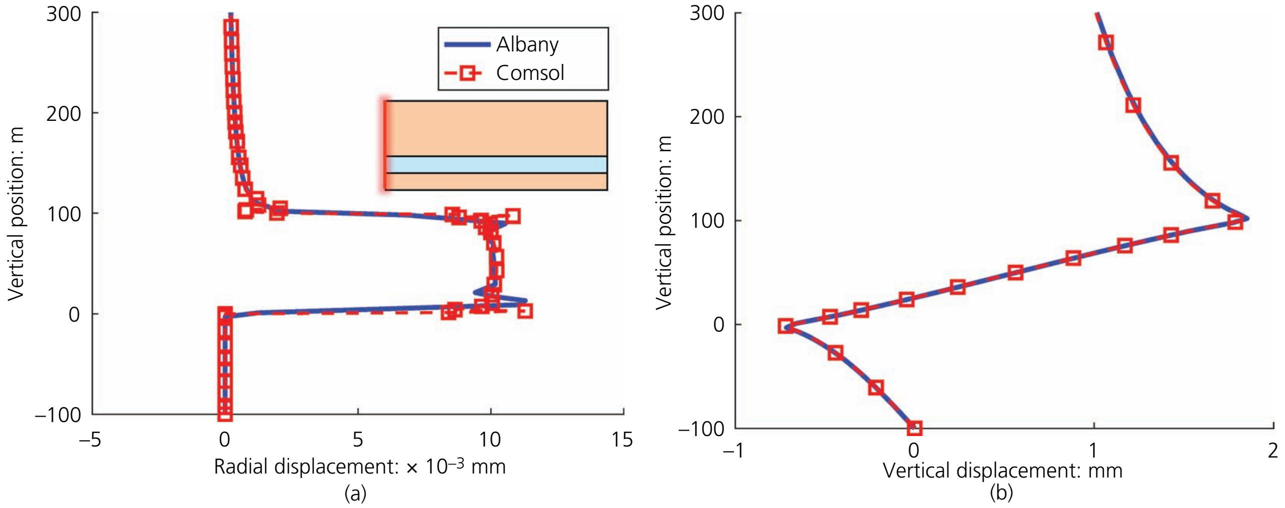
Model domain, boundary conditions and material properties

The model domain, its geometry and the meshing scheme are presented in Figure 12. In the simulation, carbon dioxide is moving through a permeable underground channel due to pressure differences between two ends of the channel. The boundary conditions are summarised in Table 3, and the same sets of



(a) (b)

Figure 8. (a) Radial and (b) vertical displacements along the radial lines in the middle of the sandstone layer and at the surface at the end of injection



(a) (b)

Figure 9. (a) Radial and (b) vertical displacements along the vertical line at the wellbore

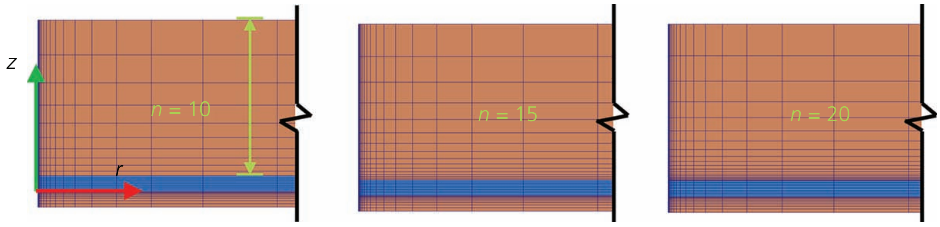
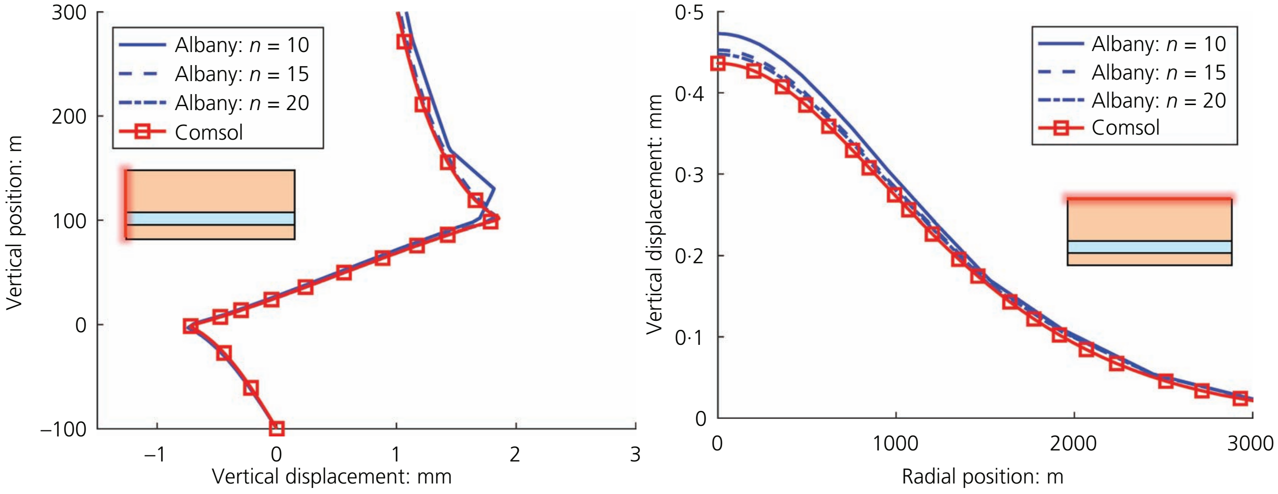


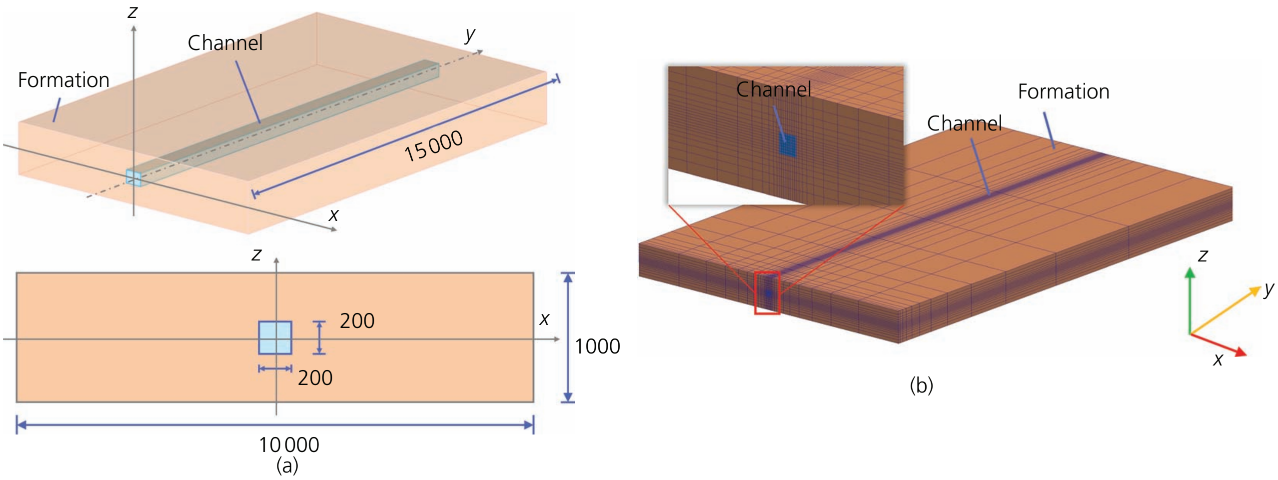
Figure 10. Three meshes used in the mesh sensitivity study



(a)

(b)

Figure 11. The vertical displacements (a) along the vertical line in the wellbore and (b) along the radial line at the surface with different mesh discretisations



(a)

(b)

Figure 12. (a) Model geometry and (b) mesh scheme for the carbon dioxide transportation problem (unit: metres)

Table 3. Boundary conditions in the carbon dioxide transportation problem

Boundary	Solid phase	Fluid phase
Top $z = 500$ m	Free $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$	Pore pressure $p = 0$
Left and right $x = \pm5000$ m	Roller $u_x = 0$	Pore pressure $p = 0$
Back $y = 15\,000$ m	Roller $u_y = 0$	Pore pressure $p = 0$
Front $y = 0$ m		
Channel	Roller $u_y = 0$	Pore pressure $p = 1$ MPa
Formation	Roller $u_y = 0$	Impermeable
Bottom $z = -500$ m	Roller $u_z = 0$	Impermeable

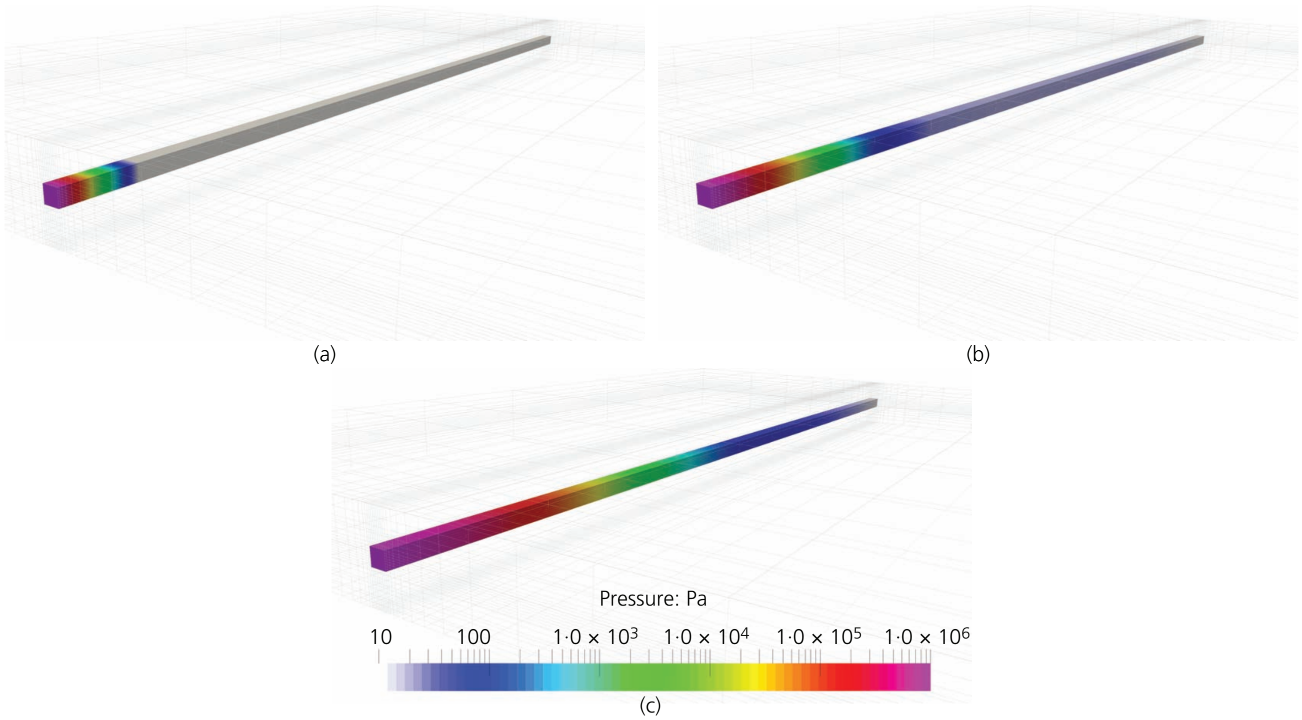


Figure 13. Contours of pore pressure distributions in the channel at different times: (a) $t = 10^4$ s; (b) $t = 10^5$ s; (c) $t = 10^6$ s

material properties in Table 2 are used in this example. The simulation lasts for 10^6 se.

Results and discussion

The distributions of pore pressure at different time are shown in Figure 13. As expected, the pore pressure is the highest (1 MPa) at the left end of the channel, and it gradually propagates towards the far end of the channel, driven by pressure differences. Also, except in the vicinity of the channel, the pore pressure remains zero in much of the confining formations due to their very low permeability. The plots of pore pressure (along the centreline of the channel) against distance at different times are shown in Figure 14, and all plots show the same behaviour – that is, the gradual decrease in pore pressure further away from the left end of the tunnel.

Figure 15 compares the pore pressure evolutions over time at different distances away from the left end of the tunnel. In

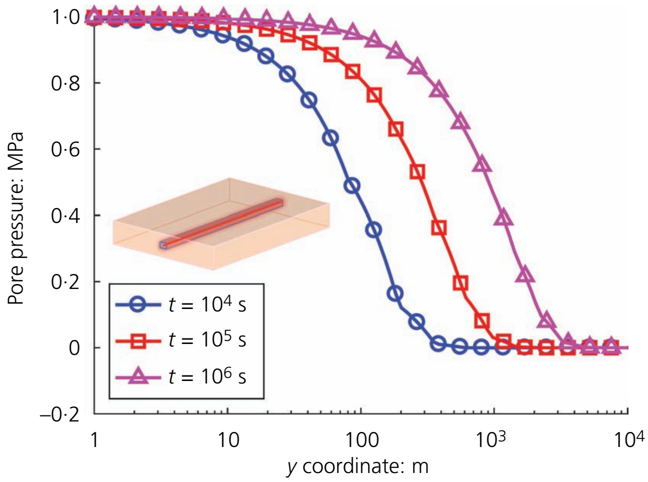


Figure 14. Pore pressure along the centreline of the channel plotted against distance at different times

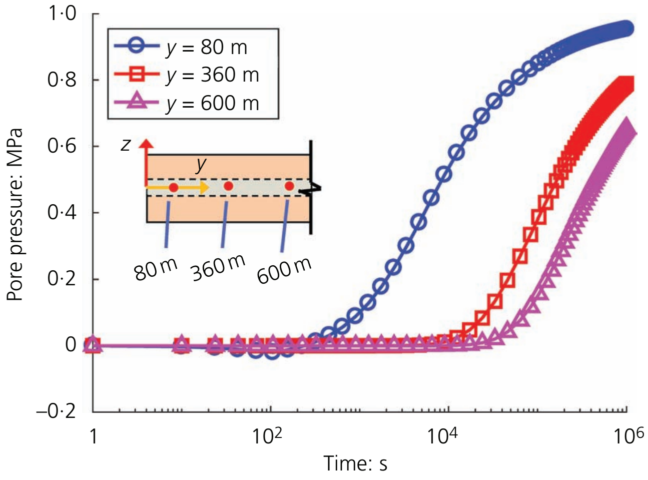


Figure 15. Pore pressure evolutions at three locations along the centreline of the channel in the y direction

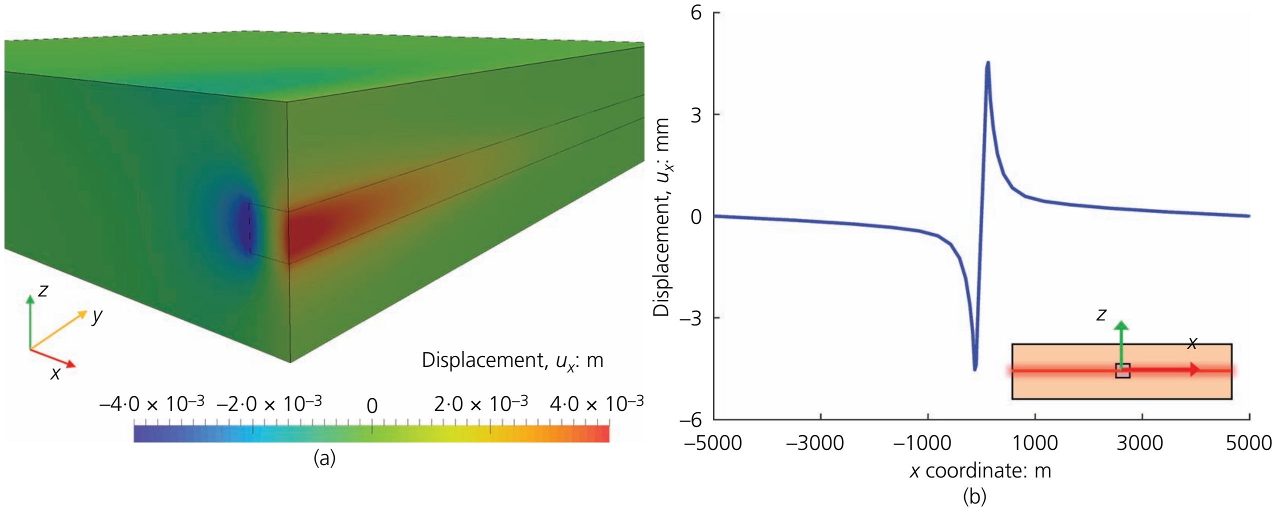


Figure 16. The displacement in the x direction at cross-section $y = 0$: (a) displacement contour; (b) displacement plotted against x coordinate along the centreline

general, the pore pressure at any position remains zero until carbon dioxide arrives, after which point the pore pressure starts to increase. The exception is the position very close to the left end (y = 80 m curve in Figure 15). For this position, there is a small negative pore pressure preceding the arrival of carbon dioxide. A similar behaviour was also observed in Comsol simulations. This negative pore pressure in the vicinity of the left end may be attributed to the coupling effect of fluid flow and solid deformation – that is, the volumetric expansion of the low-permeability formation would attract fluid flow and result in a negative pore pressure.

Figure 16 shows the contour of the x component of the displacement field, denoted as u_x, and the value of u_x along the centreline in the x direction, at the end of the simulation ($t = 10^6$ s). The left side of the channel possesses a negative x displacement, and the right side possesses a positive x displacement, indicating that the formation expanded in the x direction due to the pressure from carbon dioxide within the channel. The largest expansion is about 4·5 mm (Figure 16(b)) and is located at the channel–formation interface. The expansion vanishes rapidly at distances far away from the channel.

Figure 17 shows the contour of the y component of the displacement field, denoted as u_y, and the value of u_y along the centreline in the y direction, at the end of the simulation ($t = 10^6$ s). Much of the formation has a noticeable positive y displacement. The displacement first increases and then decreases

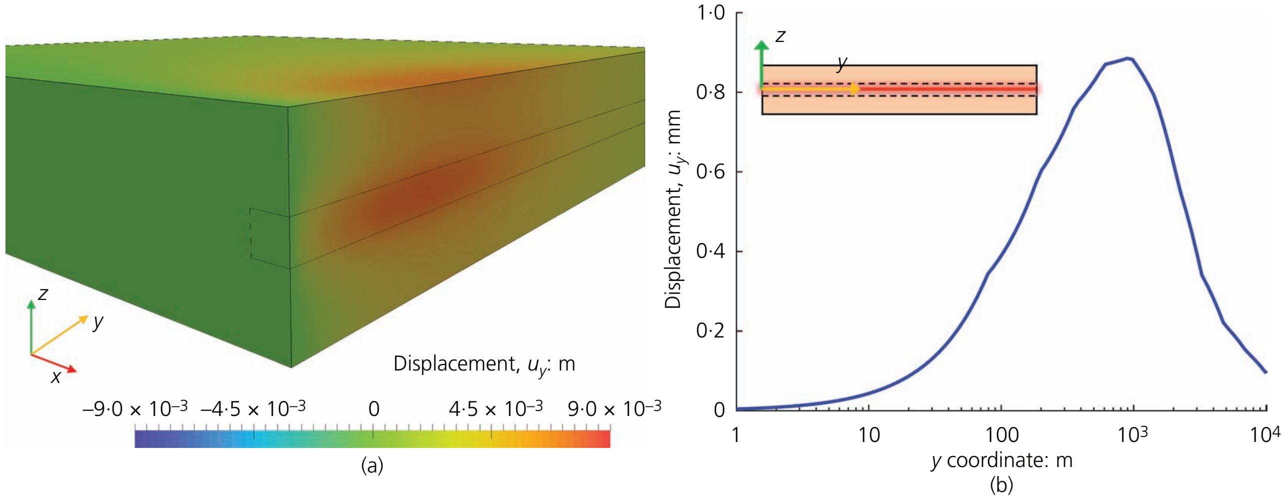


Figure 17. The displacement in the y direction at cross-section $x = 0$: (a) displacement contour; (b) displacement plotted against y coordinate along the centreline

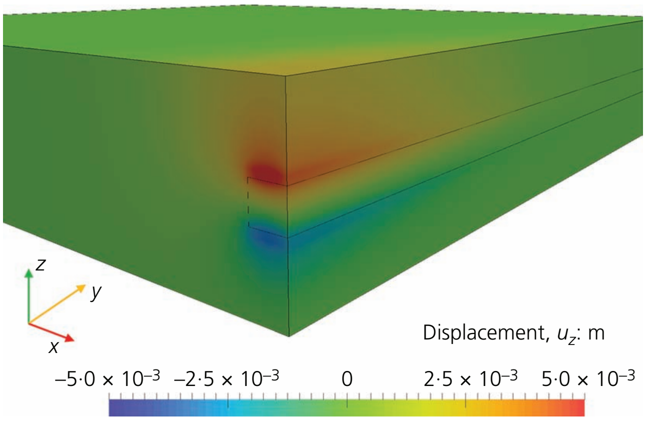


Figure 18. Contour of the vertical (*z* component) displacement field

with increasing distance away from the left end of the channel. The largest *y* displacement in the channel is about 0·88 mm, located about 900 m away from the left end.

Figure 18 shows the contour of the *z* component of the displacement (i.e. the vertical displacement) field, at the end of the simulation ($t = 10^6$ s). The largest vertical displacement is located right at the top and bottom channel–formation interfaces. The vertical deformation decreases further away from the channel. The ground surface right above the channel exhibits a noticeable positive vertical displacement.

To gain more insights into the vertical deformation at the ground surface, the vertical displacement at the ground surface along the *x* and *y* directions are extracted and plotted in Figure 19. Figure 19(a) shows that the vertical displacements at the surface along the *y* direction ($x = 0$) are the largest near the left end ($y = 0$) and gradually decrease but always remain positive away from the left end. From Figure 19(b), it is seen that the vertical displacements at the immediate vicinity of the channel (-500 m $< x <$ 500 m) are positive and rapidly decrease to zero away from the channel. The results simulated by Comsol are also presented as a comparison. Again, the two sets of results match very well. Some minor differences are observed in the immediate vicinity of the channel, where distinct material properties exist at the channel–formation interface and may pose challenges to numerical models.

Finally, to demonstrate the parallel performance of Albany, carbon dioxide transport simulations are run on Clemson University's Palmetto cluster using different numbers of central processing units CPUs (from 1 to 12). The computing time and memory usage of the simulation with one CPU are used to normalise the results, which are plotted in Figure 20. As expected, the total runtime decreases with increasing number of CPUs used in the simulation, while, at the same time, the memory usage increases. This parallel capability is particularly important for large-scale simulations and has been demonstrated for other applications using Albany (Tezaur *et al.*, 2015).

Conclusions

In this work, a generic component-based multiphysics analysis code – Albany – is introduced for the hydromechanical modelling and analysis of GCS problems. The component-based programming approach makes extensive use of algorithm libraries (in particular the Trilinos libraries) developed by subject matter experts. A key feature of Albany is the exploitation of the AD technique, providing numerical exact computations of derivatives required in assembling Jacobian matrices or consistent tangent moduli. Two numerical problems, namely, wellbore injection and carbon dioxide transport in a confined underground channel, are

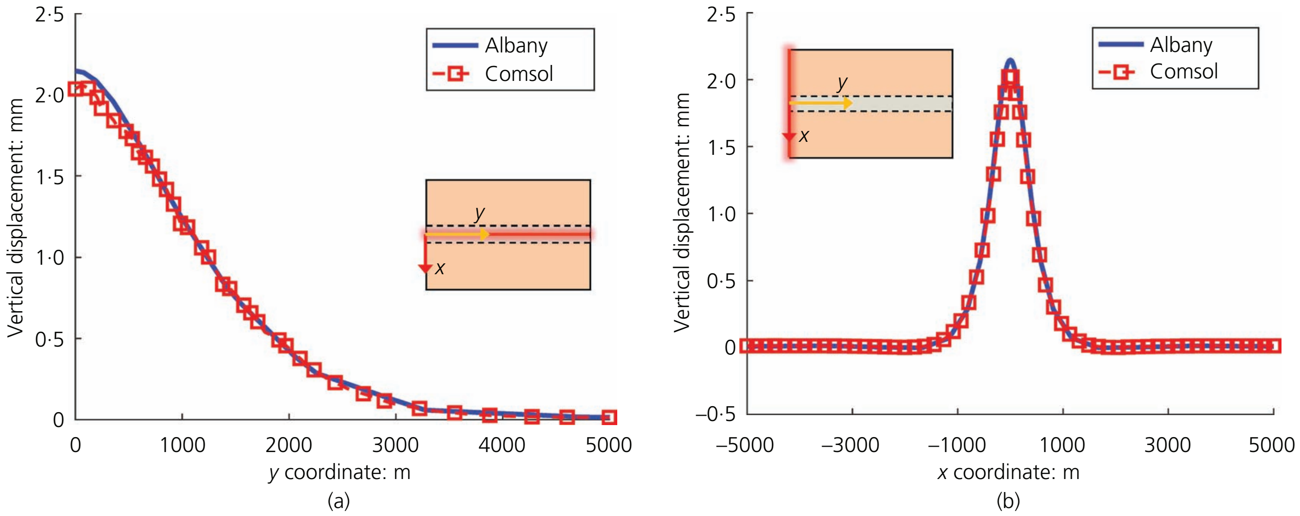


Figure 19. The vertical deformation at the ground surface along two lines in the *x* and *y* directions: (a) line $x = 0$, $z = 500$ m; (b) line $y = 0$, $z = 500$ m

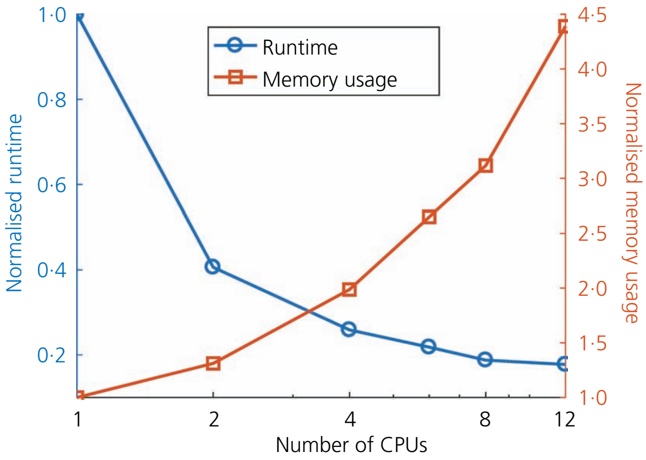


Figure 20. Parallel performance in terms of runtime and memory usage. The runtime and memory are normalised respect to their values obtained from the run with one CPU

presented to demonstrate the performance of the Albany analysis framework. Results from Albany are also verified against those obtained from Comsol – a robust commercial multiphysics finite-element package. A summary of the technical findings is presented in the following.

From the wellbore injection simulations, it is found that excess pore pressure is the highest at the injection well and gradually vanishes away from the well. Carbon dioxide is found to be trapped in the sandstone layer, a desirable situation for a GCS project. Analysis of displacement fields shows that the maximum radial displacement happens at some distance (500 m) away from the injection point, while the vertical displacement fields indicate expansion in the shale layer. From the carbon dioxide transport in a confined underground channel, it is found that the excess pore pressure gradually decreases away from the high-pressure boundary. A slight, but noticeable, negative pressure is observed to precede the arrival of carbon dioxide in the vicinity of the high-pressure boundary (80 m from the boundary). The vertical displacements at the immediate vicinity of the channel are positive and rapidly decrease to zero, while the radial displacements perpendicular to the channel exhibit a more gradual decreasing trend. In both examples, results from Albany match very well with those from Comsol, indicating the effectiveness and accuracy of the Albany framework. The parallel performance is also presented to demonstrate the scalability of Albany models for large-scale simulations commonly encountered in GCS problems.

REFERENCES

Adams BM, Bohnhoff WJ, Dalbey KR *et al.* (2009) *Dakota, a Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual*. Sandia National Laboratories, Albuquerque, NM, USA, Technical Report SAND2010-2183.

Azadi M, Aminossadati SM and Chen Z (2016) Large-scale study of the effect of wellbore geometry on integrated reservoir-wellbore flow. *Journal of Natural Gas Science and Engineering* **35**: 320–330, https://doi.org/10.1016/j.jngse.2016.08.032.

Bao J, Xu Z and Fang Y (2014) A finite-element model for simulation of carbon dioxide sequestration. *Environmental Geotechnics* **1(3)**: 152–160, https://doi.org/10.1680/envgeo.13.00024.

Bao J, Xu Z and Fang Y (2016) A coupled thermal-hydro-mechanical simulation for carbon dioxide sequestration. *Environmental Geotechnics* **3(5)**: 312–324, https://doi.org/10.1680/envgeo.14.00002.

Bathe KJ (2001) The inf–sup condition and its evaluation for mixed finite element methods. *Computers & Structures* **79(2)**: 243–252, https://doi.org/10.1016/S0045-7949(00)00123-1.

Biot MA (1941) General theory of three–dimensional consolidation. *Journal of Applied Physics* **12(2)**: 155–164, https://doi.org/10.1063/1.1712886.

Borja RI (2013) *Plasticity: Modeling & Computation*. Springer, Berlin, Germany.

Celia MA and Nordbotten JM (2009) Practical modeling approaches for geological storage of carbon dioxide. *Groundwater* **47(5)**: 627–638, https://doi.org/10.1111/j.1745-6584.2009.00590.x.

Chen Q, Ostien JT and Hansen G (2014a) Automatic differentiation for numerically exact computation of tangent operators in small-and large-deformation computational inelasticity. In *TMS 2014: 143rd Annual Meeting & Exhibition* (The Minerals, Metals & Materials Society (ed.)). Springer, Cham, Switzerland, pp. 289–296.

Chen Q, Sun WC and Ostien JT (2014b) Finite element analysis of hydro-mechanical coupling effects on shear failures of fully saturated collapsible geomaterials. In *Soil Behavior and Geomechanics* (Zhang X, Chu J and Bulut R (eds)). American Society of Civil Engineers, Reston, VA, USA, GSP 236, pp. 688–698.

Cholewinski A and Leonenko Y (2018) Modeling of ex-situ dissolution for CO_2 sequestration. *Environmental Geotechnics*, https://doi.org/10.1680/jenge.18.00012.

Damen K, Faaij A and Turkenburg W (2006) Health, safety and environmental risks of underground CO_2 storage – overview of mechanisms and current knowledge. *Climatic Change* **74(1–3)**: 289–318, https://doi.org/10.1007/s10584-005-0425-9.

Douglas C, Furtado F, Ginting V *et al.* (2010) On the development of a high-performance tool for the simulation of CO_2 injection into deep saline aquifers. *Rocky Mountain Geology* **45(2)**: 151–161, https://doi.org/10.2113/gsrocky.45.2.151.

Edwards HC, Williams AB, Sjaardema GD, Baur DG and Cochran WK (2010) *SIERRA Toolkit Computational Mesh Conceptual Model*. Sandia National Laboratories, Albuquerque, NM, USA, Technical Report SAND2010-1192.

Fossum AF and Fredrich JT (2000) Cap plasticity models and compactive and dilatant pre-failure deformation. In *Pacific Rocks 2000: Rock around the Rim* (Girard J, Liebman M, Breeds C and Doe T (eds)). A.A. Balkema, Rotterdam, the Netherlands, pp. 1169–1176.

Foster CD, Regueiro RA, Fossum AF and Borja RI (2005) Implicit numerical integration of a three-invariant, isotropic/kinematic hardening cap plasticity model for geomaterials. *Computer Methods in Applied Mechanics and Engineering* **194(50–52)**: 5109–5138, https://doi.org/10.1016/j.cma.2005.01.001.

Fu P, Settgast RR, Hao Y, Morris JP and Ryerson FJ (2017) The influence of hydraulic fracturing on carbon storage performance. *Journal of Geophysical Research: Solid Earth* **122(12)**: 9931–9949, https://doi.org/10.1002/2017JB014942.

Griewank A and Walther A (2008) *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd edn. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, vol. 105.

Harp DR, Pawar R and Gable CW (2014) Numerical modeling of cemented wellbore leakage from storage reservoirs with secondary capture due to thief zones. *Energy Procedia* **63**: 3532–3543, https://doi.org/10.1016/j.egypro.2014.11.382.

Haszeldine RS (2009) Carbon capture and storage: how green can black be? *Science* **325(5948)**: 1647–1652, https://doi.org/10.1126/science.1172246.

Heroux MA, Bartlett RA, Howle VE *et al.* (2005) An overview of the Trilinos project. *ACM Transactions on Mathematical Software (TOMS)* **31(3)**: 397–423, https://doi.org/10.1145/1089014.1089021.

Lackner KS (2003) A guide to CO_2 sequestration. *Science* **300(5626)**: 1677–1678, https://doi.org/10.1126/science.1079033.

Miehe C (1996) Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Computer Methods in Applied Mechanics and Engineering* **134(3–4)**: 223–240, https://doi.org/10.1016/0045-7825(96)01019-5.

Murdoch L, Moysey S, Germanovich L and Hughes B (2016) *Proof-of-feasibility of Using Well Bore Deformation as a Diagnostic Tool to Improve CO_2 Sequestration*. Clemson University, Clemson, SC, USA, Report DE FE0004542.

Nur A and Byerlee J (1971) An exact effective stress law for elastic deformation of rock with fluids. *Journal of Geophysical Research* **76(26)**: 6414–6419, https://doi.org/10.1029/JB076i026p06414.

Orr FM (2009) Onshore geologic storage of CO_2. *Science* **325(5948)**: 1656–1658, https://doi.org/10.1126/science.1175677.

Pan L, Oldenburg CM, Pruess K and Wu YS (2011) Transient CO_2 leakage and injection in wellbore–reservoir systems for geologic carbon sequestration. *Greenhouse Gases: Science and Technology* **1(4)**: 335–350, https://doi.org/10.1002/ghg.41.

Pawlowski RP, Phipps ET and Salinger AG (2012a) Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, part I: template-based generic programming. *Scientific Programming* **20(2)**: 197–219, https://doi.org/10.3233/SPR-2012-0350.

Pawlowski RP, Phipps ET, Salinger AG *et al.* (2012b) Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, part II: application to partial differential equations. *Scientific Programming* **20(3)**: 327–345, http://dx.doi.org/10.3233/SPR-2012-0351.

Pensado AS, Pádua AA, Comuñas MJP and Fernández J (2008) Viscosity and density measurements for carbon dioxide + pentaerythritol ester lubricant mixtures at low lubricant concentration. *Journal of Supercritical Fluids* **44(2)**: 172–185, https://doi.org/10.1016/j.supflu.2007.10.004.

Phipps ET (2015) http://trilinos.sandia.gov/packages/sacado/ (accessed 20/12/2018).

Roy S, Juha M, Shephard MS and Maniatty AM (2018) Heat transfer model and finite element formulation for simulation of selective laser melting. *Computational Mechanics* **62(3)**: 273–284, https://doi.org/10.1007/s00466-017-1496-y.

Salinger AG, Bartlett RA, Bradley AM *et al.* (2016) Albany: using component-based design to develop a flexible, generic multiphysics analysis code. *International Journal for Multiscale Computational Engineering* **14(4)**: 415–438, https://doi.org/10.1615/IntJMultCompEng.2016017040.

Seol S, Smith CW, Ibanez DA and Shephard MS (2012) A parallel unstructured mesh infrastructure. *Proceedings of High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion, Salt Lake City, UT, USA*, pp. 1124–1132.

Simo JC (1988) A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition: part I. continuum formulation. *Computer Methods in Applied Mechanics and Engineering* **66(2)**: 199–219, https://doi.org/10.1016/0045-7825(88)90076-X.

Simo JC and Hughes TJ (1998) *Computational Inelasticity*. Springer, New York, NY, USA.

Skempton AW (1954) The pore-pressure coefficients A and B. *Géotechnique* **4(4)**: 143–147, https://doi.org/10.1680/geot.1954.4.4.143.

Sloan SW, Abbo AJ and Sheng D (2001) Refined explicit integration of elastoplastic models with automatic error control. *Engineering Computations* **18(1/2)**: 121–194, https://doi.org/10.1108/02644400110365842.

SNL (Sandia National Laboratories) (2018) https://github.com/gahansen/Albany (accessed 12/12/2018).

Sun W, Ostien JT and Salinger AG (2013) A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain. *International Journal for Numerical and Analytical Methods in Geomechanics* **37(16)**: 2755–3D58, https://doi.org/10.1002/nag.2161.

Sun W, Chen Q and Ostien JT (2014) Modeling the hydro-mechanical responses of strip and circular punch loadings on water-saturated collapsible geomaterials. *Acta Geotechnica* **9(5)**: 903–934, https://doi.org/10.1007/s11440-013-0276-x.

Tezaur IK, Perego M, Salinger AG, Tuminaro, RS and Price SF (2015) Albany/FELIX: a parallel, scalable and robust, finite element, first-order Stokes approximation ice sheet solver built for advanced analysis. *Geoscientific Model Development* **8(4)**: 1197–1220, https://doi.org/10.5194/gmd-8-1197-2015.

Trilinos Project (2018) http://trilinos.org (accessed 12/12/2018).

Tu X, Andrade JE and Chen Q (2009) Return mapping for nonsmooth and multiscale elastoplasticity. *Computer Methods in Applied Mechanics and Engineering* **198(30–32)**: 2286–2296, https://doi.org/10.1016/j.cma.2009.02.014.

Vilarrasa V, Bolster D, Dentz M, Olivella S and Carrera J (2010) Effects of CO_2 compressibility on CO_2 storage in deep saline aquifers. *Transport in Porous Media* **85(2)**: 619–639, https://doi.org/10.1007/s11242-010-9582-z.

Xu T, Apps JA and Pruess K (2004) Numerical simulation of CO_2 disposal by mineral trapping in deep aquifers. *Applied Geochemistry* **19(6)**: 917–936, https://doi.org/10.1016/j.apgeochem.2003.11.003.

Xu T, Apps JA, Pruess K and Yamamoto H (2007) Numerical modeling of injection and mineral trapping of CO_2 with H_2S and SO_2 in a sandstone formation. *Chemical Geology* **242(3–4)**: 319–346, https://doi.org/10.1016/j.chemgeo.2007.03.022.