Integrating structural geological data into the inverse modelling framework of iTOUGH2

J. Florian Wellmann a,*, Stefan Finsterle b, Adrian Croucher c

a CSIRO Earth Science and Resource Engineering, 26 Dick Perry Ave., Kensington, 6151 WA, Australia
b Earth Sciences Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, USA
c Department of Engineering Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

A R T I C L E   I N F O

Article history:
Received 11 March 2013
Received in revised form
13 September 2013
Accepted 22 October 2013
Available online 31 October 2013

Keywords:
Uncertainty
Implicit geological modelling
Multiphase flow simulation
Sensitivity analysis
Parameter estimation
Monte Carlo

A B S T R A C T

The validity of subsurface flow simulations strongly depends on the accuracy of relevant rock property values and their distribution in space. In realistic simulations, this spatial distribution is based on two geological considerations: (1) the subsurface structural setting, and (2) smaller-scale heterogeneity within a hydrostratigraphic unit. Both aspects are subject to uncertainty, whereas techniques to address heterogeneity are well established, no general method exists to evaluate the influence of structural uncertainties. We present a method to include structural geological data (e.g. observations of geological contacts and faults) directly into an inversion framework, with the aim of enabling the inversion routine to adapt a full 3-D geological model with a set of geological parameters. In order to achieve this aim, we use a set of Python modules to combine several pre-existing codes into one workflow, to facilitate the consideration of a structural model in the typical model evaluation steps of sensitivity analysis, parameter estimation, and uncertainty propagation analysis. In a synthetic study, we then test the application of these three steps to analyse CO2 injection into an anticlinal structure with the potential of leakage through a fault zone. We consider several parts of the structural setting as uncertain, most importantly the position of the fault zone. We then evaluate (1) how sensitive CO2 arriving in several observation wells would be with respect to the geological parameters, (2) if it would be possible to determine the leak location from observations in shallow wells, and (3) how parametric uncertainty affects the expected CO2 leakage amount. In all these cases, our main focus is to consider the influence of the primary geological data on model outputs. We demonstrate that the integration of structural data into the iTOUGH2 framework enables the inversion routines to adapt the geological model, i.e. to re-generate the entire structural model based on changes in several sensitive geological parameters. Our workflow is a step towards a combined analysis of uncertainties not only in local heterogeneities but in the structural setting as well, for a more complete integration of geological knowledge into conceptual and numerical models.

1. Introduction

Structural geological models are commonly used to incorporate information about major geological units and their rock properties into flow simulations. It is well known that these geological models contain uncertainties (e.g. Mann, 1993; Bárdossy and Fodor, 2001; Thore et al., 2002; Turner, 2006; Suzuki et al., 2008; Caumon, 2010; Wellmann et al., 2010; Gaers, 2011; Cherpeau et al., 2012; Lindsay et al., 2012) and it is reasonable to assume that simulated flow fields are sensitive to changes in the structural geological model.

We propose a framework to test sensitivities of simulated flow fields with respect to structural parameters derived from geological data, and to use observed flow field responses to invert for these structural parameters. We establish an automated link between structural geological modelling (using an implicit geological modelling method) and multi-phase flow simulations (using the general-purpose flow simulator TOUGH2). TOUGH2 is used for a wide range of applications, from hydrogeological studies and contaminant transport, to carbon sequestration, geothermal reservoir engineering and nuclear waste disposal (Pruess et al., 2011). The link to TOUGH2 is computationally enabled via PyTOUGH, a set of Python modules offering pre- and postprocessing routines for TOUGH2 simulations (Wellmann et al., 2011). Our forward workflow from structural data to flow simulations is then integrated into an inverse modelling framework, iTOUGH2 (Finsterle, 1999), to use these data as parameters in inversions as well as sensitivity and uncertainty analyses.

The evaluation of sensitivities of simulation results to input parameters is often performed using a manual procedure, for example by testing the influence of minimal and maximal parameter values. Although this procedure can provide insights into the model behaviour, the overall informational value of the analysis is restricted (e.g. Carrera et al., 2005). A systematic analysis of sensitivities based

---

* Corresponding author. Tel.: +61 8 6436 8826.
E-mail addresses: florian.wellmann@graduate.uwa.edu.au, jan.florian.wellmann@gmail.com (J.F. Wellmann).

0098-3004/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2013.10.014
on mathematical principles delivers (in addition to quantitative sensitivity measures) a detailed error evaluation, including information on parameter covariances and correlations (Refsgaard, 1997; Sun and Sun, 2006). For flow simulations with TOUGH2, this functionality is implemented in iTOUGH2. In addition to sensitivity analysis, iTOUGH2 provides methods for parameter estimation (or inversion) from observed data and for uncertainty propagation analyses (e.g. Finsterle, 1999, 2004). With our work, we enable the application of these iTOUGH2 functionalities to the analysis of geological data and structural parameters used to develop and automatically update the underlying geological model.

With structural data, we consider direct measurements or inferred estimates of geological surface contact points and orientations, for example a point defining the position of a stratigraphic interface at depth identified from a well log, or the strike and dip of a fault observed in an outcrop. Integrating structural information into an inverse framework requires a method that automatically constructs a 3-D structural geological model from a relatively small set of parameters representing geological data. In addition, it has to be possible to export the continuous geological model directly into a discrete format, suitable for subsequent flow simulations, without any manual interaction.

These functionalities can be achieved with implicit volumetric modelling methods (Caumon, 2010). Instead of constructing surfaces between geological units, these methods interpolate a 3-D scalar potential field to structural geological data, directly honouring Geological constraints like stratigraphic relationships or fault influence (Lajamie et al., 1997; Calcagno et al., 2008; Caumon, 2010). Geological contacts are defined as equipotential surfaces in the scalar field. We apply a method that was specifically developed to create reliable models for hypothesis testing, even for sparse structural geological input data sets (Putz et al., 2006; Calcagno et al., 2008). An additional important feature of these implicit volumetric methods is that they enable a query of geological units anywhere in space, as they provide completely sealed rock volumes (Caumon, 2010). We will use this feature to facilitate automatic mesh generation for the flow simulation.

As we directly consider structural parameters, such as geological surface contact points and fault dip measurements, our approach is fundamentally different from methods that consider rock property variations on a smaller scale. It is common practice in geological modelling to first generate a structural geological model with a separation of domains with similar geological or petrophysical properties (usually referred to as zonation approach, see Carrera et al., 2005; Oliver and Chen, 2011), and then to populate these zones with rock properties, often with statistical methods to reflect heterogeneities within these zones. The grid-based simulation of multiple realisations for local heterogeneities is computationally straight-forward, widely performed, and implemented in multiple software packages, for example the geostatistical software package GSLIB (Caers, 2011; Deutsch and Journel, 1998). Excellent overviews of the available methods are given in Chilès and Delfiner (1999), Deutsch (2002) and (Caers, 2011).

By contrast, the generation of multiple structural geological models that are consistent with initial structural data and geological constraints is not a trivial task, and no general method is available to date (see chapter 8 of Caers, 2011). The main difficulty lies in the complete automation of the entire modelling procedure from initial structural data for reasonably complex and full 3-D geological models to discretise numerical models used for process simulation. Currently available methods focus on modifying a stratigraphic surface after it has been modelled (Lecour et al., 2001; Thore et al., 2002; Samson et al., 1996; Caers, 2011). An interesting approach has been proposed by Suzuki et al. (2008) where several different initial structural models are first constructed in a manual step, and the simulated stratigraphic surfaces are subsequently adapted using a dynamic data integration procedure. An important aspect of the work by Suzuki et al. (2008) is that they combine the evaluation of different structural starting models with an adaptation of stratigraphic horizons in the simulated models. However, the authors state that they do not attempt to parameterise the entire complex reservoir geometry as it would be, in their words, “extremely difficult”. In a recent publication, Cherpeau et al. (2012) address this issue and describe a technique, based on an implicit structural geological modelling method, for inverting dynamic flow data to determine parameters describing the geometry of a fault network. Our work is conceptually similar to this novel approach, but we attempt a complete parameterisation of the model using initial structural data (surface and fault contact points and orientation measurements), whereas Cherpeau et al. (2012) concentrate on the parameterisation of fault networks. With our work, we aim to add the aspect of directly considering structural data of complex 3-D geological models as parameters in a general multi-physics joint inversion framework.

We focus in the following on the computational implementation of our method for flow simulations with TOUGH2. We will first describe the underlying structural geological modelling and flow simulation methods and our approach to combine both techniques, and finally their integration into the iTOUGH2 framework. We will then apply the workflow to a CO2 injection scenario to evaluate the potential risk of CO2 leakage through a fault. We systematically explore the use of structural geological parameters in three typical model evaluation procedures: (1) sensitivity study, (2) parameter estimation, and (3) analysis of uncertainty propagation.

2. Materials and methods

The aim of this work is to enable the control of structural parameters, and therefore the geometry of the geological model, from an inverse flow simulation framework, iTOUGH2. To achieve this aim, two aspects have to be addressed:

1. A workflow is required that completely automates all steps from geological model construction to input file generation for the flow simulator.
2. This workflow then has to be integrated into the inverse framework of iTOUGH2.

We attempt to resolve both aspects with a combination of existing software and simulation codes using a set of Python and C++ programs. An overview of our approach is presented in Fig. 1, and we describe the relevant aspects below. In addition, we briefly outline relevant features of the functionality of iTOUGH2 for sensitivity analysis, parameter estimation, and uncertainty propagation analysis.

2.1. Combining geological modelling and multi-phase flow simulations

The integration of structural geological data into flow simulations is commonly performed in several steps (e.g. Carrera et al., 2005; Bundschuh and Arriaga, 2010). Starting from the available geological data (for example derived from borehole information, outcrop observations, or interpreted seismic data), a continuous geological model is constructed. This model is then discretised for the subsequent flow simulation, for example in a cartesian or tetrahedral grid. A typical method is to create a mesh that is suitable for the applied simulation procedure, and then to map geological units to each cell in the model (also referred to as model zonation). In the next step, grid cells are populated with properties according to the geological units they represent; these properties may be a stochastic realisation drawn from parameter distributions. With the addition of boundary conditions, sources and sinks,
In the context of our work, the important advantages of this implicit geological modelling approach are that (1) the formulation of geological surfaces and volumes in a potential field directly honours geological constraints, (2) the interpolation can be updated when data are changed, and (3) the potential field $T(p)$ is defined anywhere in 3-D space. The method has been applied successfully to complex geological settings, including multiply deformed mountain ranges (Maxelon et al., 2009) and fault networks (Calcagno et al., 2012). The most important aspect of the method is that a realistic geological model can be defined with a limited number of structural parameters (Putz et al., 2006; Calcagno et al., 2008). This enables us to include the initial geological data as structural parameters in an inverse framework.

The described implicit modelling method is implemented in a commercial software package (GeoModeller, www.geomodeller.com) and its functionality can be assessed through an application programming interface (API). The entire input data set for a geological model, together with geological constraints, is stored in an XML file.

We developed a Python package (pygeomod, see Appendix B for availability) to access the data in the input file and the essential methods in the API required to compute the geological model. Using this package, it is possible, for example, to change the position of a single structural geological observation, such as a surface contact point, using only a few lines of Python code. An example is given in A.3.

2.1.2. From continuous to discrete geological model

The second part of the workflow addresses the generation of a discrete version of the geological model. Discretisation is a requirement of the numerical solver implemented in the flow simulator. A rectilinear cartesian mesh is generated. Once the mesh is defined, an identifier is assigned to each cell according to the geological unit at the position of the cell center in the continuous geological model.

The automation of this step is possible because the geological model, constructed in the first step, is a continuous function that is defined everywhere in space. Once the potential field $T(p)$ and the corresponding isosurface values $t_k$ of the geological surfaces are defined, the geological unit can be determined from the value of the potential function. This functionality can be accessed through the API of the geomodelling software. The API can be accessed directly via Python, but for computational efficiency, we programmed this part in C++ and wrapped it with a Python module for high-level access (see A.1 for an example).

2.1.3. Generation of the simulation input file

The next step involves the population of grid cells with relevant rock properties and the generation of an input file for the flow simulator. The majority of the methods required for this step are already implemented in PyTOUGH (Croucher, 2011; Wellmann et al., 2011), a set of Python modules for pre- and postprocessing of TOUGH2 simulations. We extended the functionality of PyTOUGH with several additional methods, implemented in the Python package geopytough (see Appendix B).

The spatial structure of a TOUGH2 mesh is completely specified by blocks with defined volumes and properties, and the connections between pairs of blocks. PyTOUGH contains methods for working with layered grids. We use these methods to automatically generate a rectangular grid over the extent of the geological model. Relevant rock properties (permeability, porosity, etc.) are then defined on the basis of the discrete geological model (see previous step) and a defined list of rock properties for each geological unit. In addition, it is possible to define boundary conditions, an atmosphere block, and sources and sinks. The results of this step are
Flow simulation and analysis of results

The multiphase flow simulation is performed with TOUGH2, using any of the available equation of state (EOS) modules (Pruess et al., 2011; Pruess and Spycher, 2007a). As the simulation itself is executed with a system call, no further automation is required at this step. Once the simulation is performed, the output file is analysed, again with PyTOUGH. With the grid methods described in the previous step, it is possible to automatically visualize the simulation results, including plots of thermodynamic variables at one block over time, slice plots through the model, or full 3-D visualisations with the Visualization Tool Kit (www.vtk.org).

The flow simulation itself and the analysis of the results complete the workflow from geological data to flow simulation. As all steps are integrated into Python modules, the complete automation of the workflow is straightforward. It is, for example, possible to move a geological surface contact point vertically, which triggers an update of the geological and flow models, the flow simulation and analysis or results to observe what effect this change has on the simulated flow field. A practical example of this possibility is described in Section 3.1.

2.2. Integration into the inverse framework of iTOUGH2

The second part of the methodological development is the integration of the workflow described above into iTOUGH2, which is a general inverse modelling framework to perform sensitivity analysis, parameter estimation, and uncertainty propagation. (Finsterle, 1999, 2004; Finsterle and Zhang, 2011a). It has been used to study inverse problems in a wide range of multiphase flow problems, including geothermal reservoir engineering, carbon sequestration, nuclear waste isolation, and environmental studies (e.g. Finsterle, 2004; Kiryukhin et al., 2008; Zhang et al., 2011; Finsterle and Zhang, 2011b).

The capabilities of iTOUGH2 are accessible to externally provided models, including pre- and postprocessors, through the use of template and instruction files defined by the PEST protocol (Doherty, 1994). Template files are used to control input parameters adjusted by iTOUGH2; instruction files parse output files and extract the values of interest, passing them to iTOUGH2. A brief example of a template file definition is given in A.4. For more details on the interface and its possibilities, see Finsterle and Zhang (2011b).

In this work, we take advantage of the PEST interface implemented in iTOUGH2 to combine geological modelling and TOUGH2 flow simulations, described in Section 2.1. The essential parts are visualised in Fig. 1. iTOUGH2 is the central part: input parameters and output observations are defined, and the flow simulation with TOUGH2 is performed. In addition, iTOUGH2 controls the input file generation through template files in the PEST interface and a preprocessing script that invokes the workflow from structural geological data, through the geological model, to grid and input file generation. A wide range of aspects can be controlled with iTOUGH2: the structural geological data themselves (position and orientation of structural elements), the network of faults, geological modelling parameters, the mesh definition (discretisation) and refinement, the assignment of boundary conditions, geostatistical parameters, and formation properties.

As some of the algorithms implemented in iTOUGH2 require evaluation of derivatives of the model output with respect to selected input parameters, the following issue arises as a result of the proposed workflow. When a continuous geological model is mapped onto the discrete grid used by the flow simulator (see 2.1.2), the objective function is likely discontinuous, i.e., a gradual change in a structural geological parameter may lead to a discrete response. Derivative-based methods are of limited use for parameter estimation in this case. Several derivative-free parameter estimation methods for discontinuous objective functions are available in iTOUGH2 (Finsterle, 1999). In the simulation study in Section 3.2.2, we apply the downhill simplex method.

3. Application: impact of structural uncertainties on CO2 injection

As a test of the integration of implicit geological modelling into the iTOUGH2 framework, we apply the developed methods to a study of CO2 injection into a reservoir in an anticlinal structure. We focus on the low-dimensional parameterisation of the entire geological model and the consideration of uncertainties in these structural geological parameters.

The simulations are arranged according to the description in the materials and methods section above. In the first example, we will show how structural parameters can be changed, and how this change is automatically passed through to the flow simulation using a short Python script. In the second example, we demonstrate how this automated workflow enables iTOUGH2 to use structural geological data for sensitivity analyses, parameter estimation and uncertainty propagation analysis.

3.1. Forward simulation and change of the geological model

The example model represents a carbon sequestration scenario where CO2 is injected into a reservoir in an anticlinal structure below a low permeability sealing layer. The anticlinal structure is offset by a fault, where the fault damage zone can be a potential leakage pathway for CO2 (Fig. 2). The problem is simulated as a 2-D vertical slice with a lateral extent of 5 km, a depth of 2 km, and a thickness of 200 m.

The geological model can be adjusted with only six free structural geological parameters, presented in Fig. 2: three points (RW, RF, and RE) define position and shape of the anticlinal structure, and an additional point (S), representing the position of the top of the seal, which has uniform thickness. The position of the fault zone is parameterised by two points (F7 and F8) near the top and bottom of the fault. The error bars shown in Fig. 2 indicate uncertainties in these parameters: depth and shape of reservoir and seal are uncertain, and so is the lateral position of the fault zone. The following additional geological constraints are considered known and are thus fixed for the base model and all derived
models: thickness of fault zone, position of the reservoir top at the west side of the fault, age constraints between reservoir, seal and cover rock, and the influence of the fault zone on the different geological units. For further details, see the geological model available online (see Appendix B).

The injection simulation is performed with TOUGH2’s ECO2N equation-of-state module (Pruess and Spycher, 2007b). This property module describes mixtures of water, NaCl, and CO2 and heat, enabling the simulation of CO2 injection into deep aquifers, with consideration of salinity. The implemented thermodynamic equations are valid for temperatures of up to approximately 110 °C and pressures of 600 bar.

Rock properties are assigned similarly to those in the CO2 injection example studies described in the ECO2N manual (Pruess, 2005). The two main lithologies are adapted to the CO2 injection scenario defined here: a sand-rich lithology and a low-permeability shale. The reservoir cap rock (i.e. the seal) is simulated as a shale, while all other rocks are simulated as sand-rich lithologies. Relative permeability and capillary pressure functions are defined using the van Genuchten model (Van Genuchten, 1980).

The simulation is performed in two steps. To determine initial pressure and temperature conditions, a steady-state conductive temperature field is simulated for a basal heat flux of 0.06 W/m² and an atmospheric temperature of 15 °C, resulting in a temperature gradient of approximately 30 °C per kilometer. Pressure at the surface is set to 1 atmosphere. In the next step, we simulate the injection of 25 kg/s of CO2 into the anticlinal structure for 30 years. Initial salt mass fraction in the entire model domain is 1%, and no CO2 is present before injection. The simulation of CO2 injection itself is performed in isothermal mode.

The combination of geological modelling and TOUGH2 simulation is an example of the first part of the workflow presented in

![Fig. 3. Discrete geological model, spatial distribution of CO2 after 30 years of injection, and CO2 at observation wells for (a) the base case and (b) the modified version generated by adding 300 m to the lateral position of the fault (parameters $F_T$ and $F_B$ in Fig. 2). Note that geological model construction, discretisation, TOUGH2 input file generation and simulation, and visualisation of results are automatically updated without any further manual interaction (see Appendix A for code). (a) Base case. (b) Modified model.](image-url)
Fig. 1. It involves combining Geomodeller with PyTOUGH to generate a TOUGH2 input file. For the relevant Python code, see A.1, and for the visualisation to generate Fig. 3, see A.2.

The discretised geological model and corresponding simulation results for the base-case injection scenario are visualised in Fig. 3(a). The continuous geological model is discretised into a regular mesh with 80 cells in E-W, and 50 cells in z-direction. The spatial distribution of CO$_2$ in the model domain after a simulation time of 30 years (Fig. 3(a), middle figure) shows the leakage through the fault and the main distribution around observation wells O3–O6, visible also in the plots of CO$_2$ mass fraction in the aqueous phase at the observation wells as a function of time (bottom Fig. 3(a)).

Next, we use the same part of the workflow with the additional step of changing the geological model with modifications in the structural parameters. With only a few lines of code (see A.3), the parameters defining the top and bottom positions of the fault ($F_T$ and $F_B$) are increased by 300 m, resulting in a lateral shift of the fault zone towards the west (Fig. 3(b) top).

Re-running the code for model generation with these updated structural parameters results in a new TOUGH2 input file. The updated distribution of rocktypes, simulation results, and CO$_2$ histories at the observation wells are shown in Fig. 3(b). The difference in geological model, spatial distribution of CO$_2$, and the CO$_2$ mass fraction in the observation wells is evident.

This simple example shows that our approach enables the combination of implicit structural geological modelling with input file generation for TOUGH2 simulations. The steps from changing the initial points of the structural model, recomputing the geological potential field, discretising and exporting the geological data to a template Python script developed in the previous section has to be modified to a template file according to the PEST protocol (Finsterle, 2011). This is easily done by replacing each value defining the relative change of a structural geological parameter (see A.3 for an example) by a PEST-style variable name that can be interpreted by iTOUGH2. The modified Python script and the corresponding iTOUGH2 inversion input files are available from the online repository (see Appendix B).

### 3.2. Control of geological model construction with iTOUGH2

We will now apply the entire workflow presented in Fig. 1, including the integration into iTOUGH2, for three commonly performed analysis steps with respect to simulation and model evaluation (Carrera et al., 2005): sensitivity analysis, parameter estimation, and uncertainty propagation analysis.

In this modelling study, we evaluate the influence of the six structural parameters defining the geological model (Fig. 2), as well as six material properties (the permeability and porosity of the fault, reservoir, and seal). The initial values and assigned standard deviations are summarised in Table 1. Note that we consider changes relative to the initial value for the structural parameters. A positive change to the points defining the fault position ($F_T$ and $F_B$) represent a shift towards the east, and for the other parameters a shift downwards.

In order to use iTOUGH2 as a wrapper for the workflow from geological data to flow simulation for the CO$_2$ injection study, the Python script developed in the previous section has to be modified to a template file according to the PEST protocol (Finsterle, 2011). This is easily done by replacing each value defining the relative change of a structural geological parameter (see A.3 for an example) by a PEST-style variable name that can be interpreted by iTOUGH2. The modified Python script and the corresponding iTOUGH2 inversion input files are available from the online repository (see Appendix B).

#### 3.2.1. Sensitivity study

We first evaluate the local sensitivities of the simulated CO$_2$ values at the observation wells with respect to structural and hydrological input parameters. For increased accuracy, we use a central finite difference scheme for the approximation of the Jacobian, requiring a total of $(2n+1) = 25$ forward simulations, where $n$ is the number of adjustable parameters.

Results from the sensitivity analysis provide insights into data sensitivity, parameter influence, and correlations between parameters. The parameter sensitivities are presented in Table 2. In addition to composite sensitivities, a measure of overall parameter correlation $\Gamma$ ($0 < \Gamma \leq 1$) is obtained, where higher values indicate that the corresponding parameter is relatively independent.

<table>
<thead>
<tr>
<th>Type</th>
<th>Id (unit)</th>
<th>Name</th>
<th>Initial</th>
<th>Stdev</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>$R_W$ (m)</td>
<td>Reservoir left</td>
<td>(0)</td>
<td>50</td>
<td>–100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>$R_F$ (m)</td>
<td>Reservoir right</td>
<td>(0)</td>
<td>50</td>
<td>–100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>$R_T$ (m)</td>
<td>Reservoir top</td>
<td>(0)</td>
<td>50</td>
<td>–100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>$S$ (m)</td>
<td>Seal</td>
<td>(0)</td>
<td>50</td>
<td>–100</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>$F_T$ (m)</td>
<td>Fault position</td>
<td>(0)</td>
<td>500</td>
<td>–1000</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>$F_B$ (m)</td>
<td>Fault position</td>
<td>(0)</td>
<td>500</td>
<td>–1000</td>
<td>1000</td>
</tr>
<tr>
<td>Property</td>
<td>log($k_F$) (m$^2$)</td>
<td>Reservoir perm.</td>
<td>–12.1</td>
<td>0.3</td>
<td>–13</td>
<td>–11</td>
</tr>
<tr>
<td></td>
<td>log($k_S$) (m$^2$)</td>
<td>Seal perm.</td>
<td>–16</td>
<td>1.0</td>
<td>–17</td>
<td>–15</td>
</tr>
<tr>
<td></td>
<td>log($k_R$) (m$^2$)</td>
<td>Fault perm.</td>
<td>–11.5</td>
<td>1.5</td>
<td>–15</td>
<td>–11</td>
</tr>
<tr>
<td></td>
<td>$\phi_F$ (%)</td>
<td>Reservoir por.</td>
<td>15</td>
<td>2.5</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>$\phi_S$ (%)</td>
<td>Seal por.</td>
<td>10.3</td>
<td>5</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>$\phi_R$ (%)</td>
<td>Fault por.</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>35</td>
</tr>
</tbody>
</table>

Table 2 Sensitivity analysis for the parameters defining the geological structure and rock properties. The ratio between conditional and marginal standard deviation $\Gamma$ is a measure of overall parameter correlation (higher values are less correlated).
The parameters defining the geological structure have a strong influence on simulated CO$_2$ concentrations at the observation wells. The most influential parameters are the structural points defining the top position of the fault ($F_T$), and the west and top of the anticlinal structure ($R_W$ and $R_T$). From the parameters of the rock properties, only the reservoir porosity $\Phi_R$ and permeability $k_R$ are of comparable sensitivity. The values of $\Gamma$ indicate that the structural point defining the top position of the fault, and the fault permeability, are the least correlated parameters.

In addition to the sensitivities of the parameters, we also obtain a covariance analysis, which reflects correlations between the different parameters. The matrix of the direct, pairwise parameter correlations is visualised in Fig. 4. As already evaluated with $\Gamma$ in Table 2, it is evident that the structural parameter defining the top of the fault $F_T$ and the permeability of the fault $k_F$ are the least correlated parameters (with corresponding rows and columns showing neutral colours). The top and east points defining the structure of the anticline ($R_E$ and $R_S$) are highly correlated with almost all of the rock properties, and the rock properties have strong correlations among each other. It is interesting to note that the correlations between the structural parameters, i.e., the points defining the eastern ($R_E$) and western ($R_W$) flank of the anticline, are positively correlated. On the other hand, the point defining the top of the reservoir ($R_T$) is positively correlated with the western flank, but negatively with the eastern flank. These correlations follow from the fact that those points define the shape of the anticline, and therefore the overall volume of CO$_2$ that can be stored before breakthrough through the fault occurs. The correlations between these parameters and the reservoir porosity $\Phi_R$ can be interpreted similarly.

In addition to the relevance of sensitivity and correlation analyses for experimental and observational design, the analysis of these measures is important for subsequent parameter estimation. In general, the more sensitive a parameter is, and the less correlated it is with other parameters, the better we will be able to estimate its value from the available observations.

The formal analysis of sensitivities with iTOUGH2 combined with our workflow therefore provides direct insight into sensitivities and correlations for the initial structural geological parameters. In the next step, we will use parameter estimation methods to determine if the structural parameters can be identified from a given set of observations.

3.2.2. Parameter estimation

In our example model, we simulate the observation of CO$_2$ in several shallow observation wells above an area where a leakage zone (in our case parameterised as a fault) would be expected. In a realistic case, it could be of interest to determine the position of this leakage zone, given the measurements of CO$_2$ concentrations.

In the sensitivity study, it was determined that the parameter $F_T$ defining the top position of the fault has a significant influence on the objective function (Table 2), and that we should be able to determine the value of this structural parameter by inverse modelling.

We apply a parameter estimation algorithm to see if it is possible to determine several structural parameters that define the main features of the structural geological model, using CO$_2$ data from observation wells. As this is a synthetic example, the main interest is to see whether the parameter estimation procedure, in combination with our method of re-creating the structural model directly from a small set of geological parameters, is capable of generating a sufficiently wide range of different structural models, and of identifying models close to that used to generate the synthetic data (Fig. 3(a)).

As pointed out in Section 2.2, derivative-based parameter estimation methods are not suitable for the problem considered here; we therefore use the derivative-free downhill simplex minimisation algorithm. As structural parameters, we consider the parameters defining the western side and the top of the reservoir anticline ($R_W$ and $R_T$) in addition to the fault top to allow for wider variability in the structural geological model. In addition to the three structural parameters, we consider the rock properties of the fault zone to be uncertain and assign initial values that are offset from the values used for the forward simulation ($F_T = -25$, $R_W = -500$, $R_T = -25$, log $k_F = -12.5$, $\Phi_R = 0.1$, for comparison with initial values, see Table 1).

Results for the evolution of the structural parameters during successive downhill simplex iterations are presented in Fig. 5. The algorithm explores initially a wide range of different structural models, based on the three structural parameters. After
approximately 15 iterations, the position of the fault is relatively well established. Parameters for the structural points defining the top and west side of the reservoir anticline are reasonably well determined within the accuracy of the model discretisation.

Two of the structural models that were explored during parameter estimation are shown in Fig. 6. The difference in the structural models is clearly visible in the position of the fault. The final structural model obtained as a result of the analysis is similar to the initial model in Fig. 3(a), with an anticlinal structure that is less pronounced than in the two models shown here.

3.2.3. Uncertainty propagation

Finally, we evaluate how uncertainties in the structural geological model propagate through the predictive model of CO2 leakage. Building on the results from the inversion, we wish to evaluate the total amount of CO2 leaked through the fault, together with estimated upper and lower bounds, given the uncertainties in the parameters. Instead of simply using the initial, uncorrelated parameter standard deviations described in Table 1, we propagate the full covariance matrix as determined by the inversion through the prediction model.

In iTOUGH2, two uncertainty propagation methods are implemented: the Monte Carlo method and linear propagation analysis (FOSM). We use here the Monte Carlo method, as the linearity assumptions in the linear propagation analysis are too limiting in our example. We can expect a complex response due to the inherent non-linearity of two-phase flow and the discrete impact of geological structures on leakage.

The analysis of the sensitivity matrix showed that some of the model parameters are strongly correlated (Fig. 4). Consequently, instead of independently sampling from each parameter distribution, we account for parameter correlations using a Latin Hypercube sampling design (Zhang and Pinder, 2003). We generate 100 realisations, using as the means the best estimates of the three structural parameters ($F_T$, $R_W$ and $R_T$) and the rock properties of

![Fig. 6. Examples of structural models explored during parameter estimation with the downhill simplex method, requiring only an adjustment of three structural parameters.](image)

![Fig. 7. Results of uncertainty propagation analysis for total mass of CO2 in the different geological units as a function of time. The green line shows the median value, dashed blue lines 5% and 95% quantiles and the gray dots lower and upper bounds. The numbers indicate the number of the parameter set leading to this result. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)](image)
the reservoir \((k_r \text{ and } \Phi_r)\), and the full covariance matrix (see Table C1) as determined at the minimum of the objective function.

Results of the analysis are shown in Fig. 7. The green lines in the four subfigures represent the median for the mass of CO2 in the respective geologic units, the dashed blue lines the 5% and 95% quantiles, and the grey dots are the minimal and maximal values with numbers indicating the parameter set leading to this result. The uncertainty in structural parameters and reservoir properties considerably affect the predictions of interest. For example, the CO2 arrival time at the fault ranges from 3 to 8 years, and the estimated total amount of CO2 stored in the reservoir after an injection period of 30 years varies from \(4 \times 10^9\) to \(1.2 \times 10^{10}\) kg. Note that the amount of CO2 leaking through the fault to the atmosphere is very large in this synthetic reservoir. This unfavourable outcome may be a result of the 2-D nature of the model along with the chosen location and properties of the fault, which establish an escape pathway for the injected CO2 that cannot be bypassed, leading to the high leakage amount once the cumulative amount injected has reached the limited storage volume beneath the anticlinal structure.

The two structural models corresponding to the parameter sets where CO2 reached the fault and atmosphere first (parameter set 79) and last (parameter set 74) are shown in Fig. 8. Although the differences are not as pronounced as for the models explored during the parameter estimation routine (Fig. 6), the differences in the structural models are clearly visible in the position of the fault and the shape of the anticline.

4. Discussion

The core of our method lies in the application of an implicit geological modelling method that enables the construction of realistic and complex full 3-D structural models with a low-dimensional parameterisation (Calcagno et al., 2008). We examined here a 2-D model for a relatively simple structural setting. However, numerous examples exist where the same implicit modelling method has been applied to complex 3-D geological settings (Martelet et al., 2004; Maxelon and Mancktelow, 2005; Joly et al., 2000; Calcagno et al., 2012). An important consideration is that the structural parameters are closely related to typical geological observations. In our synthetic simulation study, we considered observations of points defining the surface contact between main geological units \((R_{WP}, R_r, R_p \text{ and } S \text{ in Fig. 2})\) and the position of a fault \((F_r, F_b)\). Orientation data, e.g., the dip of a fault, can be incorporated in an analogous manner (see the example in A.3). Because orientation data are considered as gradients of the potential field (Lajaunie et al., 1997), they are not restricted to positions on an observed surface but can be included anywhere in space, for example derived from measurements of sedimentary layering within one unit. The ability to include both aspects, structural points as well as orientation measurements, provides great flexibility in the construction of the structural models. However, special care should be taken in the set-up of the initial model by limiting the number of structural parameters and by making sure they are only weakly correlated.

We described in detail in the introduction the difference between changing the structural model, as we do here, and adjusting spatial distributions of properties, as commonly performed with geostatistical methods. Both aspects usually contain uncertainties which, ideally, should be considered in a realistic evaluation where a flow simulation is based on an initial structural model. Geostatistical methods for the simulation and estimation of property heterogeneity have already been implemented into iTOUGH2 (Finsterle and Kowalsky, 2008). Our work provides the methods to consider uncertainties in the initial structural model. A systematic evaluation of both uncertainties in the structural model and smaller-scale heterogeneities is possible with this combination and the topic of further research.

Representing complex geology in a numerical model may lead to meshes with a very large number of elements and connections, making the simulation computationally demanding. This is especially the case when thin layers or complex structural settings are considered. The methods that we apply for geological modelling and model discretisation are not restricted by the number of elements. However, the computational demands of the TOUGH2 simulation may be a limiting factor. Note that TOUGH2-MP (Zhang et al., 2008), a parallel version of the simulator, has been applied to problems with several million grid blocks. TOUGH2-MP can be integrated using PyTOUGH and executed as an external simulation through the PEST interface.

The fact that distinct geological units of the continuous geological model are mapped onto a discrete mesh may lead to problems when numerically evaluating derivatives with respect to structural parameters, as a small change in a structural parameter might not change the discrete structural model at all. As mentioned before, this may prevent or limit the use of derivative-based parameter estimation methods, specifically the Levenberg–Marquardt algorithm, which is generally very successful for non-linear inverse problems. We thus used the downhill simplex as a derivative-free parameter estimation algorithm in our demonstration of the workflow. Other derivative-free parameter estimation methods that could be applied include grid search, simulated annealing, harmony search, differential evolutionary algorithm and other global minimisation algorithms implemented in iTOUGH2. However, different strategies can also be envisaged on the level of the geological discretisation by ensuring that small changes in geological parameters always lead to continuous,
differentiable changes in the simulation output. Such strategies include adaptive meshing (in a sense of adapting a mesh to a changed geological model realisation), or the inclusion of smoothly varying property averages across stratigraphic unit interfaces as a function of the position of this interface within a computational element.

5. Summary and conclusions

We developed a workflow that enables the adaptation of a geological model through structural parameters within a framework for forward and inverse modelling of multiphase flow in fractured porous media. We developed methods to automate the entire procedure from geological modelling, to model discretisation, to input file generation for flow simulations, to flow simulation and analysis of model results. An important feature of our approach is that we integrate the construction of the geological model through a powerful implicit geological modelling approach: the input mesh for the flow simulation is directly reconstructed from primary structural parameters, such as surface contact points, fault observations, or orientation measurements. The integration enables us to consider this type of geological information directly in typical model evaluation procedures, like sensitivity analysis, parameter estimation, or uncertainty propagation analysis. Our workflow provides a way to evaluate the influence of structural data on flow simulation results, and to estimate primary structural geological parameters from a variety of observations.

We tested the proposed methodology for the three iTOUGH2 application modes with a synthetic example of CO2 injection into an anticlinal structure, with potential leakage through a fault zone. We constructed a structural geological model consisting of four geological units and defined six geological parameters (contact points between the different units) that enabled us to modify the main features of the model that are considered uncertain in our case: the depth and shape of the anticlinal structure, the thickness of the seal, and the lateral position of the fault. We showed that, with only a few lines of code, it is possible to change the parameters defining the position of the fault, and to update the input mesh for the flow simulation given the new geological model. We showed that iTOUGH2 is able to examine the structural model in sensitivity, inverse modelling, and uncertainty propagation analyses. Given the often dominant effect that the geological structure has on flow simulation results, the proposed parameterisation of geological features and the integration into the iTOUGH2 framework adds an important new capability, as it allows the modeller to formally analyse aspects of the conceptual model, estimate the geometry of geological structures, and evaluate the impact of related uncertainties on the simulate system response.

Acknowledgments

We would like to thank the three anonymous reviewers for their constructive comments. The first author is funded by a CSIRO Office of the Chief Executive Post-Doctoral Fellowship scheme within the CSIRO Earth Science and Resource Engineering Division. The work of the second author was supported, in part, by the U.S. Dept. of Energy under Contract no. DE-AC02-05CH11231. Special thanks to Intrepid Geophysics Ltd. for providing an academic version of the software GeoModeller.

Appendix A. Code for model setup and modification

This section contains Python code snippets that illustrate the usage and flexibility of the method presented in this manuscript. The entire code for all examples and the geological model is available on the online repository (see Appendix B).

Python is an open-source interpreted programming language that provides powerful numeric and scientific methods as well as data visualisation techniques (e.g. Langtangen, 2008; Pérez et al., 2011). Python is available for virtually all relevant operating systems. Due to the flexibility of the language, it is ideally suited to combining pre- and post-processing of other simulation codes into a single framework.

A.1. Export of geological model into mesh format for TOUGH2 simulation

The geological model is exported with a Python wrapper for an external C++ program that accesses the API of the geological modelling software (GeoModeller). The geological model is re-computed and exported into a regular grid (rectilinear grids are also possible):

```python
# define mesh
(nx, ny, nz) = (80, 1, 50)
copy_files_export_model(geomodel_dir, new_dir, nx, ny, nz)

For the TOUGH2 simulation, a template TOUGH2 input file is adapted and a new mesh is generated from the exported model:

dat = geopytough.GeoT2Data('CO2_template.dat')
dat.load_geomodel(os.path.join(geomodel_dir, geomodeller_xml_file))
dat.create_regular_mesh_from_geomodel(nx, ny, nz, 
    keep_types = True, save_mesh = True, convention = 0)
dat.update_properties_from_csv_list(csv_file_name)
dat.update_model_from_exported_grid()
dat.write('tmp.dat', meshfilename = "MESH")
```
A.2. Visualisation of rocktypes and CO2 distribution

The visualisation of rocktypes and CO2 distribution, as, for example, presented in Fig. 3 is possible with standard PyTOUGH methods. The discretised geological model (i.e. the distribution of rocktypes) can be visualised with

```python
geo.slice_plot('x', rocktypes = dat2.grid, aspect = 'equal',
              cbar_orientation = 'horizontal',
              plot_limits = ((0,5000),(-2000,0)),
              linewidth = 0.1,
              colormap = 'summer')
```

A slice plot of distribution of CO2 after 30 years of injection (the last time step) is generated with

```python
lst = t2l.t2listing("CO2_template.out")
lst.last()
geo.slice_plot('x', variable = lst.element['XCO2aq'], aspect = 'equal',
              cbar_orientation = 'horizontal',
              plot_limits = ((0,5000),(-2000,0)),
              linewidth = 0.1,
              colormap = 'gray_r')
```

A.3. Modification of geological model

The geological model is adapted with methods of the Python module geopytough:

```python
# initialise geo-logical model object and load model G1 =
G0.GeomodellerClass()
G1.load_geomodeller_file(os.path.join(geomodel_dir, geomodeller_xml_file))

# Get Sections in model and extract points
from sections sect = G1.get_sections()
points = G1.get_formation_point_data(sect[0])

# All points that are assoicated with the fault contain "fault" in
# their ObservationID (in Geomodeller). We now locate all those points
# and shift them by a specified z-value

for point in points:
    name = point.find("+G1xmlns+")Data").get("Name")
    obs_id = point.attrib['ObservationID']
    if "fault" in obs_id:
        G1.change_formation_point_pos(point, add_x_coord = 300)

# Now save the Geomodel xml file and copy it together with all
# relevant Geomodeller files into a new project directory

G1.write_xml(os.path.join(new_dir,'new.xml'))
```
In a similar way, it would be possible to adjust orientation measurements, for example to add 10 degrees to the dip of a fault orientation measurement:

```python
# Change the angle of a fault

foliosations = G1.get_foliation(sect[0])

for foliation in foliations:
    name = point.find("{"+G1 xmlns=""})Data".get("Name")
    obs_id = point.attrib['ObservationID']
    if "fault" in obs_id:
        G1.change_foliation(foliation, add_dip = 10)
```

### A.4. Adjusting the Python script for use with iTOUGH2 through the PEST interface

The Python script for geological model construction and export has to be converted into a PEST template file for use with iTOUGH2. Only some minimal changes are required, basically substituting the values defining the relative changes with a PEST-style variable that can be interpreted by iTOUGH2. Following the example in A.3, an example would be

```python
fault_top = @f_top@
```

```
[...]
```

```python
for point in points:
    name = point.find("{"+G1 xmlns=""})Data".get("Name")
    obs_id = point.attrib['ObservationID']
    if "fault_top" in obs_id:
        G1.change_formation_point_pos(point, add_x_coord = fault_top)
```

With the according parameter block in the iTOUGH2 input file:

```plaintext
> PARAMETER
  >> PEST
    >>> NONE Parameter 1
      >>>> NAME : f_top
      >>>> RANGE : 1000. 3000.
      >>>> DEVIATION : 500.0
      >>>> distribution is NORMAL
      >>>> VALUE
      <<<
      <<<
<<<
```

Complete examples for PEST template and iTOUGH2 input files are available on the online repositories.
Appendix B. Details of online repositories for code and example simulations

The Python modules that were developed for this work are available on the github page of the first author:

→ https://www.github.com/lohoro/geoptyough
and
→ https://www.github.com/lohoro/pygeomod

Python scripts, geological model, TOUGH2 template files, and iTOUGH2 inversion setup files are also available on

Note that all modules are provided without any warranty or liability and published under a Creative Commons share alike license with attribution to the original work. When used in any way, refer this publication.

In addition, the following programs are required:

• A 2.x Python installation, including matplotlib, Numpy and Scipy (available on → http://www.python.org);
• PyTOUGH, freely available on → https://www.github.com/acroucher/PyTOUGH;
• iTOUGH2 with the ECO2N modules (see → http://esd.lbl.gov/ITOUGH2/ for software, source code, and licensing);
• The commercial geological modelling software Geomodeller (See → http://www.geomodeller.com).

Table C1
Matrix with estimated parameter variances and covariances (diagonal and lower triangular matrix) and correlation coefficients (upper triangular matrix) used for LHS sampling.

<table>
<thead>
<tr>
<th></th>
<th>( R_W )</th>
<th>( R_T )</th>
<th>( F_r )</th>
<th>( k_R )</th>
<th>( \Phi_R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_W )</td>
<td>( 0.23 \times 10^2 )</td>
<td>( -0.21 )</td>
<td>( 0.33 )</td>
<td>( 0.40 )</td>
<td>( -0.16 )</td>
</tr>
<tr>
<td>( R_T )</td>
<td>( -0.17 \times 10^2 )</td>
<td>( 0.31 \times 10^3 )</td>
<td>( 0.88 \times 10^3 )</td>
<td>( 0.12 )</td>
<td>( -0.10 )</td>
</tr>
<tr>
<td>( F_r )</td>
<td>( 0.15 \times 10^3 )</td>
<td>( -0.94 \times 10^3 )</td>
<td>( 0.36 \times 10^4 )</td>
<td>( 0.04 )</td>
<td>( 0.002 )</td>
</tr>
<tr>
<td>( k_R )</td>
<td>( 0.29 )</td>
<td>( 0.31 )</td>
<td>( 0.32 )</td>
<td>( 0.23 \times 10^{-1} )</td>
<td>( -0.89 )</td>
</tr>
<tr>
<td>( \Phi_R )</td>
<td>( -0.55 \times 10^{-2} )</td>
<td>( -0.12 \times 10^{-1} )</td>
<td>( 0.64 \times 10^{-1} )</td>
<td>( -0.96 \times 10^{-1} )</td>
<td>( 0.50 \times 10^{-4} )</td>
</tr>
</tbody>
</table>

Fig. C1. Estimated covariances for the parameters in the CO2 injection study, obtained as additional result from the sensitivity study (Section 3.2.1).
The Python modules are independent of the underlying operating system. iTOUGH2 can be compiled on a wide range of operating systems and Geomodeller is currently available for Windows and Linux. The simulations presented in this work were performed on a 64-bit PC running Ubuntu Linux. The workflow has also been tested successfully on Windows PCs.

Appendix C. Additional material related to the simulation results

C.1. Covariance estimation

A figure of estimated covariances, determined for the sensitivity analysis in Section 3.2.1.
The table with covariance and correlation values used for LHS sampling in the Monte Carlo uncertainty propagation analysis is given in Table C1.

C.2. Grid search

In order to obtain a better overall picture of the shape of the objective function, we will first perform a simple grid search. As described above (Section 2.2), a grid search provides a detailed picture of the objective function, albeit for a very high computational cost. We therefore limit it to the two most sensitive parameters (see Table 2): the top position of the fault, and the porosity of the reservoir (Fig. C1).

The objective function is visualised in Fig. C2 for 11 increments for both parameters in the ranges of 0.05 to 0.25 for $\Phi_R$ and $-500$ to $+500$ for $F_T$. The minimum of the objective function, around the values used to generate the synthetic dataset ($F_T=0$, $\Phi_R=0.15$) is clearly visible. In addition, the positive correlation between both parameters can be detected, as identified before by the evaluation of direct correlations (Fig. 4): a position of the fault closer to the injection (increasing values of $F_T$) is balanced by a higher reservoir porosity. It can also be seen that, in the range evaluated here, only one clear minimum exists. The grid search, although an expensive parameter estimation method, reveals the interplay between a structural parameter, the fault position, and a rock property, reservoir porosity.