NRAP-Open-IAM User's Guide

Release alpha 2.7.1-23.06.30

NRAP-Open-IAM Development Team

Jun 30, 2023

CONTENTS:

1 Obtaining NRAP-Open-IAM **1**

1.1 Introduction . . . 1
1.2 Downloading NRAP-Open-IAM . . . 1
1.3 Installing NRAP-Open-IAM . . . 2
1.4 Testing installation . . . 3
1.5 Contributors . . . 3

2 Getting Started **5**

2.1 Conceptual model and overview . . . 5
2.2 Visualization options . . . 9
2.3 GUI Operation . . . 9
2.4 Control Files . . . 14
2.5 Output . . . 18
2.6 Analysis Options in Control File . . . 19
2.7 Setup of visualization options . . . 20
2.8 Units . . . 40

3 Components Description **41**

3.1 Stratigraphy Component . . . 41
3.2 Simple Reservoir Component . . . 42
3.3 Analytical Reservoir Component . . . 43
3.4 Lookup Table Reservoir Component . . . 44
3.5 Generic Reservoir Component . . . 45
3.6 Multisegmented Wellbore Component . . . 46
3.7 Cemented Wellbore Component . . . 47
3.8 Cemented Wellbore (WR) Component . . . 48
3.9 Open Wellbore Component . . . 49
3.10 Generalized Flow Rate Component . . . 50
3.11 Hydrocarbon Leakage Component . . . 51
3.12 Seal Horizon Component . . . 52
3.13 Fault Flow Component . . . 55
3.14 Fault Leakage Component . . . 58
3.15 Carbonate Aquifer Component . . . 59
3.16 Deep Alluvium Aquifer Component . . . 60
3.17 FutureGen2 Aquifer Component . . . 62
3.18 FutureGen2 AZMI Component . . . 63
3.19 Generic Aquifer Component . . . 64
3.20 Atmospheric Model Component . . . 65
3.21 Plume Stability Component . . . 66
3.22 Chemical Well Sealing Component . . . 67

4 Component Comparison 69

5 Use Cases 73

Bibliography 83

Python Module Index 87

CHAPTER

ONE

OBTAINING NRAP-OPEN-IAM

1.1 Introduction

NRAP-Open-IAM is an open-source Integrated Assessment Model (IAM) for Phases II and III of the National Risk Assessment Partnership (NRAP). The goal of this software is to go beyond risk assessment into risk management and containment assurance. NRAP-Open-IAM is currently in active development and is available for testing and feedback only.

As this is a prototype of software being actively developed, we are seeking any feedback and/or issue reports. An online feedback form can be found here: https://docs.google.com/forms/d/e/1FAIpQLSed5mcX0OBx1dLNmYGbmS4Vfc0mdOLapIzFqw-6vHoho9B19A/viewform?usp=sf_link

Issue reports and feedback can be left at the forum on NETL's Energy Data eXchange webpage for NRAP-Open-IAM: https://edx.netl.doe.gov/workspace/forum/nrap-tools/topic?t=nrap-tools-nrap-open-iam or on GitLab issues page for NRAP-Open-IAM: https://gitlab.com/NRAP/OpenIAM/-/issues?sort=created_date&state=opened

If you have been given access to the code indirectly and would like to be notified when updates are available for testing, please contact the development team to be added to our email list.

1.2 Downloading NRAP-Open-IAM

NRAP-Open-IAM tool and examples can be downloaded from a public GitLab repository located at https://gitlab.com/NRAP/OpenIAM. If the NRAP-Open-IAM was downloaded from the GitLab repository, the folder name may have the repository's current hash appended to it. Feel free to rename the folder something simple like *NRAPOpenIAM* to simplify the navigation.

1.3 Installing NRAP-Open-IAM

The NRAP-Open-IAM requires Python version 3.9 or greater to operate. If you need to install Python, we describe all steps of the installation process below.

General Installation Guide:

- Extract the tool files from the provided/downloaded zip.
- Navigate to the *installers* folder within the recently unzipped directory.
- Navigate to the folder corresponding to the operating system that you are utilizing.
- Follow instructions file located in the folder for your operating system.

For Windows: The file *Installation_Instructions_Windows.txt* describes steps required to install needed Python packages for the proper work of NRAP-Open-IAM.

For macOS: The file *Installation_Instructions_macOS.txt* describes steps user needs to follow in order to install required Python packages.

For Linux OS: Linux users are assumed to know the installation commands for their specific version of Linux needed to install required tools. The file *Installation_Instructions_Linux.txt* specifies the needed software and package dependencies.

For alternative installation of Python the following packages are needed: `NumPy`, `SciPy`, `PyYAML`, `Matplotlib`, `Pandas`, `TensorFlow` (of version 2.6), `Keras`, `scikit-learn`, `Pmw`, `pip`, and `six`. In most cases (mainly dependent on the platform and Python distribition) the required libraries can be installed using `pip` or `conda` package managers. Additional libraries recommended to run Jupyter notebooks and scripts illustrating work of NRAP-Open-IAM are `IPython` and `Jupyter`.

On macOS and Linux machines the gfortran compiler needs to be present/installed to compile some of the NRAP-Open-IAM code (macOS users can find gfortran here: (https://gcc.gnu.org/wiki/GFortranBinariesMacOS)).

After the proper version of Python is installed, the NRAP-Open-IAM can be set up and tested. **Note: If Python was installed through Anaconda please use Anaconda prompt instead of command prompt for setup and tests.** In the NRAP-Open-IAM distribution folder find and open the sub-folder *setup*. Next, open a command prompt/Anaconda prompt in the *setup* folder (on Windows, this can be done by holding Shift and right clicking inside the folder when no file is selected, then selecting `Open command window here`; alternatively, one can navigate to the folder *setup*, type `cmd` in the address bar of the file browser and hit Enter to open the command prompt there). (On Windows, Anaconda prompt can be found in the programs menu under submenu `Anaconda3 (64-bit)`.) Run the setup script by entering the command:

```
python openiam_setup_tests.py
```

in the command prompt/Anaconda prompt. This will test the version of Python installed on the system. Next the setup script will test the versions of several Python libraries that the NRAP-Open-IAM depends on. The setup script will compile several Fortran libraries needed for some component models on Mac and Linux. Users of Windows OS will be provided with the compiled libraries. Finally, the setup script will run the test suite to see if the NRAP-Open-IAM has been installed correctly. If the results printed to the console indicate errors during the testing the errors have to be resolved before the NRAP-Open-IAM can be used. When contacting the developers to resolve problems please include all output from the setup script or test suite runs.

1.4 Testing installation

After setup the test suite can be run again by entering the NRAP-Open-IAM *test* directory in a terminal and typing:

```
python iam_test.py
```

Test results will be printed to the terminal. The setup script run during the installation process uses the same test suite after testing whether the necessary Python libraries are installed, and compiling the NRAP-Open-IAM libraries.

1.5 Contributors

During the Phase II and/or Phase III of the NRAP the following researchers contributed to the development of NRAP-Open-IAM (listed in alphabetical order with affiliation at the time of active contribution):

- Diana Bacon (Pacific Northwest National Laboratory)
- Seunghwan Baek (Pacific Northwest National Laboratory)
- Pramod Bhuvankar (Lawrence Berkeley National Laboratory)
- Suzanne (Michelle) Bourret (Los Alamos National Laboratory)
- Julia De Toledo Camargo (Pacific Northwest National Laboratory)
- Bailian Chen (Los Alamos National Laboratory)
- Abdullah Cihan (Lawrence Berkeley National Laboratory)
- Dylan Harp (Los Alamos National Laboratory)
- Paul Holcomb (National Energy Technology Laboratory)
- Jaisree Iyer (Lawrence Livermore National Laboratory)
- Elizabeth Keating (Los Alamos National Laboratory)
- Seth King (National Energy Technology Laboratory)
- Greg Lackey (National Energy Technology Laboratory)
- Ernest Lindner (National Energy Technology Laboratory)
- Kayyum Mansoor (Lawrence Livermore National Laboratory)
- Mohamed Mehana (Los Alamos National Laboratory)
- Saro Meguerdijian (Los Alamos National Laboratory)
- Nathaniel Mitchell (National Energy Technology Laboratory)
- Omotayo Omosebi (Lawrence Berkeley National Laboratory)
- Veronika Vasylkivska (National Energy Technology Laboratory)
- Ya-Mei Yang (National Energy Technology Laboratory)
- Yingqi Zhang (Lawrence Berkeley National Laboratory)

CHAPTER

TWO

GETTING STARTED

The NRAP-Open-IAM has several ways for a user to build and run simulations, including graphical user interface (GUI), text based control files and python scripts. The simplest way to build and run simulations for NRAP-Open-IAM is the GUI. This guide will primarily focus on using the GUI to interact with the NRAP-Open-IAM. To launch the GUI open a command prompt in the *source/GUI* directory and type:

```
python NRAP_OPENIAM.py
```

2.1 Conceptual model and overview

NRAP-Open-IAM is designed for modeling and simulating the behaviour of geologic carbon storage (GCS) system models. These system models can include representations of reservoir response to CO_2 injection, the resulting impacts on potential leakage pathways like wellbores, and any contaminant plumes that form in aquifers or the atmosphere due to brine or CO_2 leakage. Each part of this process (reservoir response, leakage pathway behavior, and impacts on the receptor of leakage) is represented by a particular component within the system model. Additionally, NRAP-Open-IAM can automate specific types of analysis, for example, determining the area of review for a GCS site. The intended purpose of NRAP-Open-IAM is to provide quantitative metrics that aid in GCS project planning, permitting, and operational decision support. NRAP-Open-IAM can be run through the graphical use interface (GUI; section *GUI Operation*), the control file interface (section *Control Files*), or a script-based approach in Python.

2.1.1 System model design

Within NRAP-Open-IAM, the system model contains components that are connected to each other. For example, a reservoir component can provide pressures and CO_2 saturations to a wellbore component. The wellbore component then uses these inputs to calculate CO_2 and brine leakage rates to a specific aquifer. In its turn an aquifer component can use these leakage rates to model the evolution of pH and TDS plumes within the aquifer. An aquifer component is a receptor type of component which can also be represented by an atmosphere component. For example, a wellbore component can provide leakage rates to the atmosphere, and an atmosphere component can then be used to model the evolution of atmospheric contaminant plumes. Fig. 2.1 is a conceptual model demonstrating the connection of reservoir components, wellbore components, and a receptor (aquifer or atmosphere components). Almost all NRAP-Open-IAM system models include a stratigraphy component. A stratigraphy component can be used to define the thicknesses and depths of the storage reservoir, aquifers, and aquitards in the model domain. Many NRAP-Open-IAM components are designed to link with the stratigraphy component in a way that automatically sets component parameters related to the stratigraphy without additional user efforts.

The setup demonstrated in Fig. 2.1 is the most common design of an NRAP-Open-IAM system model. Other designs not fitting the illustrated scheme are possible as well. For example, some of the components producing leakage rates are not designed to be linked to an aquifer component that accepts leakage from a point source. The `SealHorizon` component can use pressures and CO_2 saturations from a reservoir component to model brine and CO_2 leakage rates through a fractured caprock overlying the storage reservoir, but the predicted leakage rates occur over an area of the

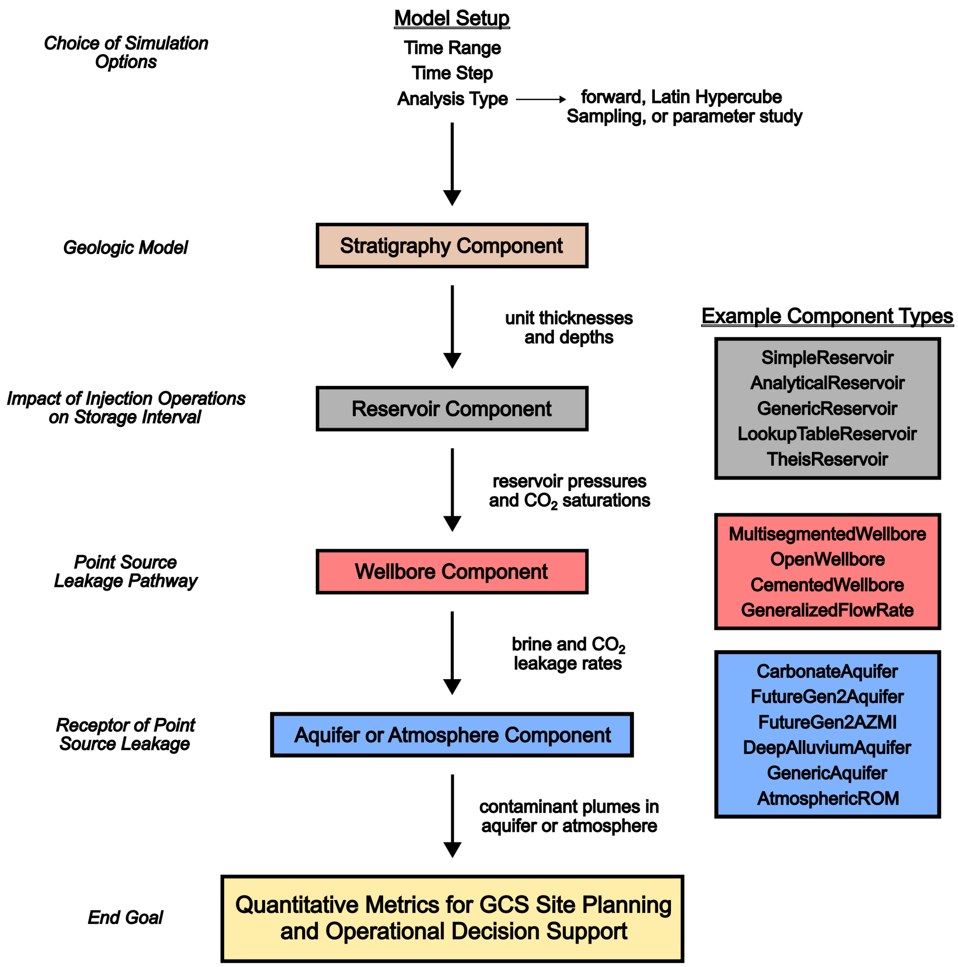


Fig. 2.1: Standard System Model Design

caprock/reservoir interface. The aquifer components shown in Fig. 2.1 are designed to receive leakage rates from a point source, like a wellbore, and, therefore, are not appropriate to use with rates that apply over an area or a linear element (e.g., faults in the `FaultFlow` component). Such components can still be used to evaluate leakage risks, however, and Fig. 2.2 shows a conceptual model demonstrating the connection of a reservoir component with a non-local leakage pathway component (i.e., not a point source leakage pathway). Aquifer components accepting leakage rates that apply over areas or linear elements may be developed in the future. The system model shown in Fig. 2.2 represents how these non-local leakage pathway components can be used with the currently available aquifer components.

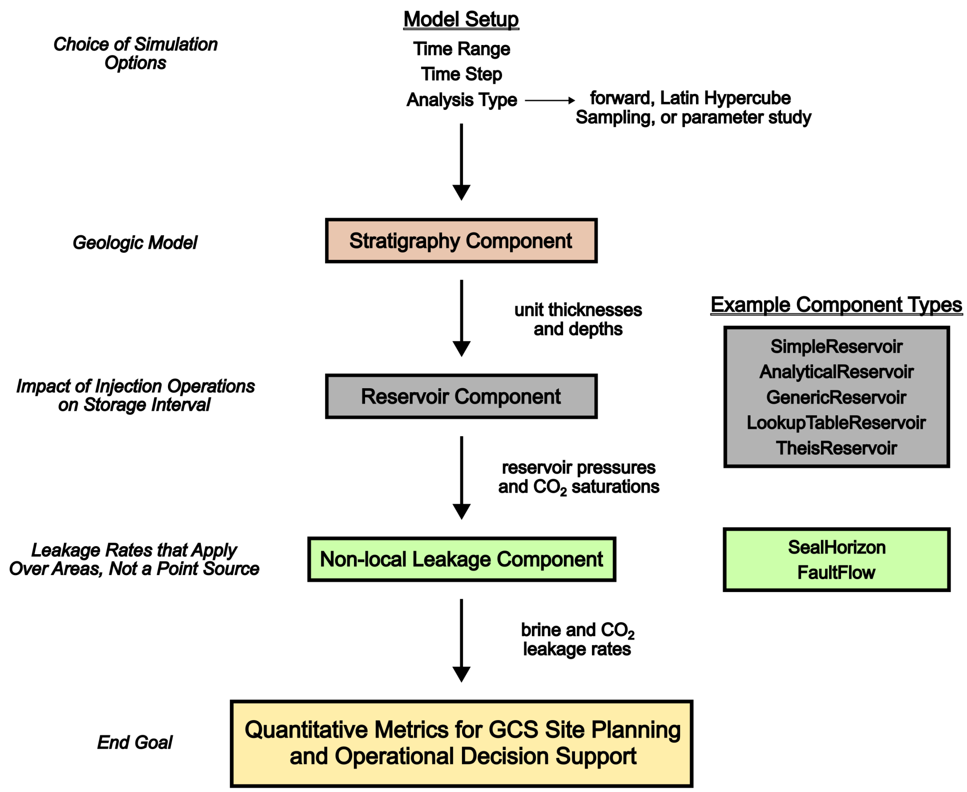


Fig. 2.2: Non-local Leakage System Model Design

NRAP-Open-IAM has several components that are designed to not be connected to other components. For example, the `PlumeStability` component does not connect to the stratigraphy or leakage pathway components. Instead, it uses .csv files containing reservoir conditions over time to evaluate the development and movement of plumes within the reservoir (e.g., plume areas, velocity of the plume centroid, and the direction in which the dispersion of the plume occurs). The `ChemicalWellSealing` component, which evaluates if and when a fracture will seal due to calcite precipitation, also does not connect with other components. This type of system model design is demonstrated in Fig. 2.3.

NRAP-Open-IAM is designed to accept a variety of component designs, so it can be used to create many different system model designs. Note that the same type of .csv files used by `PlumeStability` components can also be used to read reservoir conditions into a system model with `LookupTableReservoir` components (e.g., driving wellbore leakage components with output from high-fidelity reservoir simulations).

The support of one GCS site may require the use of multiple system models. For example, the `ChemicalWellSealing` component (Fig. 2.3) can help inform which wells are likely to self-seal and, therefore, might be excluded from being considered as a possible leakage pathway in a larger GCS system model (Fig. 2.1). The `ChemicalWellSealing` component has a parameter related to reservoir pressure, so one should first constrain this parameter by running a reservoir simulation (either through a separate program or through an NRAP-Open-IAM simulation where the reservoir

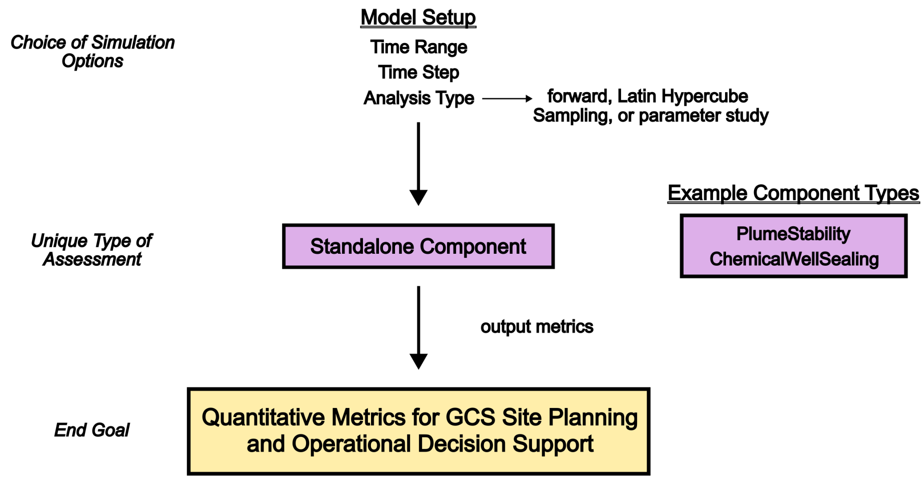


Fig. 2.3: Standalone Component System Model Design

component is the final component in the chain). In this case, one might use three GCS system models (1) to constrain reservoir pressures, (2) to determine whether particular wells will self-seal (Fig. 2.3), and (3) to model the behavior of the whole GCS site (excluding wells that will self-seal; Fig. 2.1).

2.1.2 Component parameters and applications

Each component model has a variety of parameters, and these parameters have different limits or ranges. When setting up a simulation, any component parameter that is not given a value by the user will automatically be set to the parameter's default value.

Different components are meant to be used in different applications. For example, the `TheisReservoir` component was designed to handle multiple (brine) injection and/or extraction wells over a large area, with the pressures produced reflecting the interaction of multiple wells. In contrast, the `SimpleReservoir` component can simulate only one injection well. The `FutureGen2Aquifer` and `FutureGen2AZMI` components were designed to represent the specific geochemistry of the Future Gen site but may be applicable to similar aquifers with depths between 100 m and 700 m (`FutureGen2Aquifer`) and 700 m and 1600 m (`FutureGen2AZMI`). The `OpenWellbore` component is useful for worst-case comparison scenarios where leakage through an unplugged wellbore enters an aquifer. In contrast, a `MultisegmentedWellbore` component simulates flow through impaired cement and can account for leakage into multiple aquifers overlying the injection zone. More details regarding the distinctions between different wellbore components and their intended applications are available in chapter *Component Comparison*. Descriptions of NRAP-Open-IAM components and their parameters are provided in chapter *Components Description* below.

2.1.3 Analysis types

NRAP-Open-IAM simulations can use one of three analysis types: `Forward`, Latin Hypercube Sampling (`LHS`), or parameter study (`Parstudy`) (Fig. 2.1 - Fig. 2.3). A `Forward` analysis (`forward` in the control file interface) is deterministic. Each parameter is fixed at a certain value, so the simulation has the same results each time it is run.

`LHS` simulations (`lhs` in the control file interface) are stochastic: the parameters are varied between minimum and maximum values according to their dsitribution. An `LHS` simulation consists of many separate realizations, where each realization has different parameter values. This approach helps to address parameter uncertainty. For example, by constraining parameter values within reasonable (and random) bounds, one can focus on the proportion of `LHS` realizations in which leakage occurs. This proportion can be used to estimate the likelihood of such a leakage event.

Leakage occurring in 10% of realizations could be considered to correspond to a 10% probability of leakage, given the user's current understanding of the parameter values. Latin Hypercube Sampling is different from a purely random approach in that it evenly samples the range of values for each stochastic parameter, and runs the model for different combinations of parameter values. A purely random sampling of parameter values could fail to explore a large section of the parameter space. `LHS` simulations divide each parameter's range into subranges; within each subrange, a parameter value is randomly selected. Because `LHS` simulations need to evenly sample the parameter space through the assessment of multiple subranges, each `LHS` simulation requires a minimum number of realizations. If one attempts to run an `LHS` simulation without a sufficient number of realizations, the code will raise a `LinAlgError` saying "Matrix is not positive definite." If this error occurs, increase the total number of realizations.

The `Parstudy` analysis type (`parstudy` in the control file interface) also evaluates different realizations of each simulation, where each realization has different parameter values. The `Parstudy` analysis type also divides a variable parameter's specified range into separate subranges and then selects a value from each subrange. While an `LHS` simulation has the user to specify how many realizations to evaluate, a `Parstudy` simulation has the user to specify how many parameter values to use for each stochastic parameter (i.e., number of subranges within the overall range). As a result, the number of realizations for `Parstudy` simulations increases exponentially with the number of stochastic parameters.

Overall, the `Forward` analysis type is intended to be used for decision-support of GCS system models that are deterministic and conceptually appropriate. The `LHS` analysis type is intended to be used for decision-support of GCS system models that address the uncertainty in parameter values. Finally, the `Parstudy` analysis type is intended to be used to study the effects of certain parameters on model outputs (i.e., sensitivity analysis). Studying the effects of certain parameters is important for decision support. Understanding which parameters have the most significant impact can help inform the user about the parameters they should focus on constraining in their study area. Control file interface can help to automate sensitivity analysis within `LHS` simulations; see *ControlFile_ex8a-ControlFile_ex8c*.

2.2 Visualization options

NRAP-Open-IAM includes a variety of visualization options. For example, there are multiple options for plotting a model domain's stratigraphy with 2-dimensional and 3-dimensional plots. The 3-dimensional `Stratigraphy` plot type can display features like injection sites and wellbores. The user can specify spatial variations in unit thicknesses and depths, and the stratigraphy visualization options are provided to help users ensure that the model domain conforms to the intended design. Other plot types available with NRAP-Open-IAM include: time series of outputs; map-view figures showing the extent of reservoir and aquifers impacts (meant to inform an operation's area of review (`AoR`)); map-view figures showing when monitoring wells at specified locations can detect contaminant plumes in aquifers (time to first detection, `TTFD`); and map-view figures showing the extent of atmospheric CO_2 plumes. The plotting options in NRAP-Open-IAM through the GUI interface (section *GUI Operation*) are somewhat limited. Control file and script inteface have access to all visualization options in NRAP-Open-IAM (see, e.g., section *Setup of visualization options*). Simulation results are saved in .csv files, however, and these files can be used to create figures in separate programs.

2.3 GUI Operation

When the GUI is first opened a disclaimer screen will be shown followed by the main interface.

To begin building a model click on the **Enter Parameters** button. The process of building a model consists of entering basic model parameters, defining the geologic stratigraphy of the site, then adding a component model for each component of the system to be modeled. Therefore, the first tab that user would see after clicking the **Enter Parameters** button is the model parameters view.

Start by defining a `Simulation name` for the model: the name also will be used as the name of the file containing details of NRAP-Open-IAM simulation. Time should be entered in years. The `End time` is the number of years during which the simulation will be run. A uniform time step will be taken during the simulation specified by the `Time step` entry (typically 1 year time steps are used). The NRAP-Open-IAM can perform three types of simulations and/or analysis:

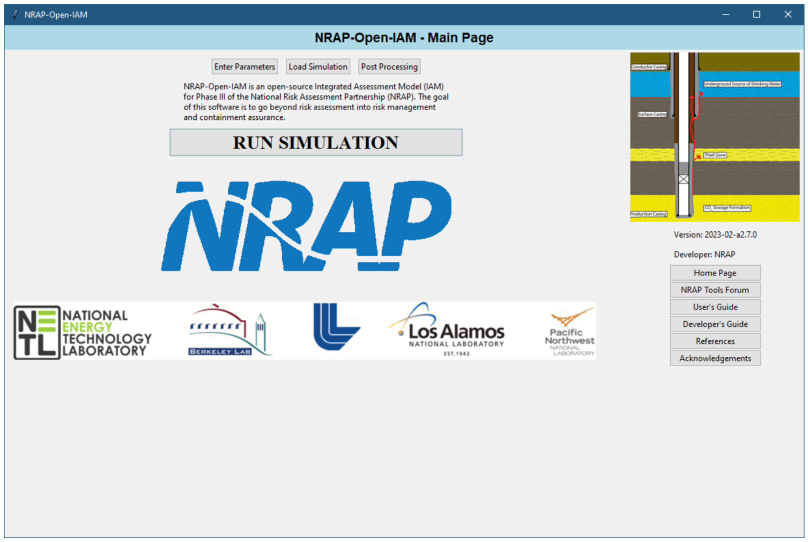


Fig. 2.4: Main NRAP-Open-IAM Interface

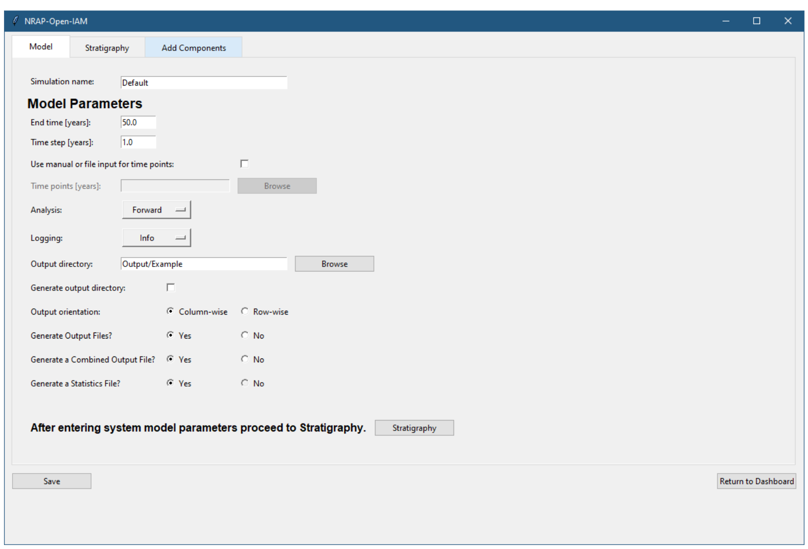


Fig. 2.5: Model Parameters View

Forward, LHS, and Parstudy. Forward analysis runs a single deterministic scenario. LHS (abbreviation for Latin Hypercube Sampling) is the type of random parameters sampling used to run stochastic simulations. Parstudy (short for parameter study analysis) divides user-defined range for each stochastic variable into equally spaced subdomains and selects parameter value from each subdomain. Parstudy analysis is useful for studying the effects of several variables on the components outputs but the number of realizations grows exponentially with the number of variables.

The NRAP-Open-IAM creates a log file with each simulation run: the level of information being logged in can be set by changing value of Logging entry. In general, the default level of Info would contain the most useful messages. A Debug (debugging) level of Logging will contain more information about component model connections, setup and calls, but will produce very large files and should be avoided for large simulations. Warn (warning) and Error levels can be used if log file sizes become an issue.

The NRAP-Open-IAM will save all the simulation results to the specified Output directory. In text field corresponding to Output directory user needs to enter a path to the folder where the output will be saved. In the case the entered path does not exist the empty directory will be created if box Generate output directory is checked. Additionally, if the provided path is not absolute, it is assumed that it starts in the NRAP-Open-IAM root folder. A {datetime} stamp can be added to the folder name so that each run of a particular simulation will be saved separately, otherwise results from a previous run will be overwritten by subsequent runs until the output folder is changed. After setting up the model parameters proceed to the Stratigraphy tab.

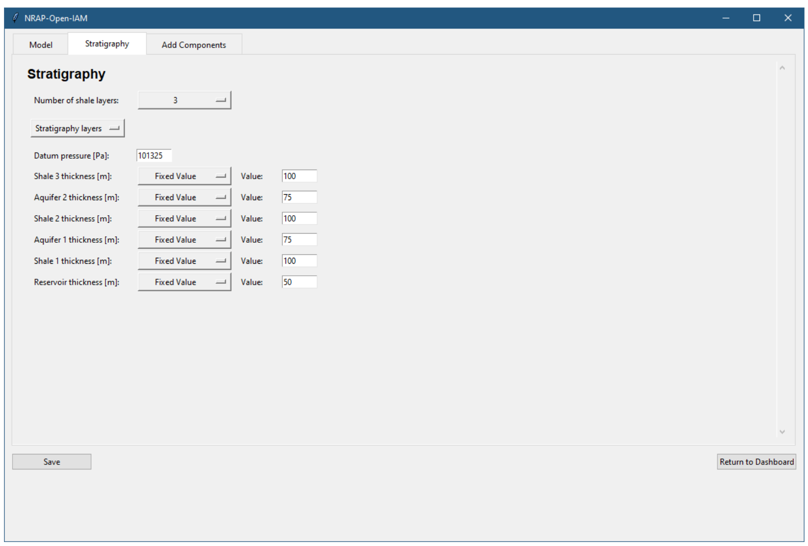


Fig. 2.6: Stratigraphy View

In the Statigraphy tab model parameters related to the stratigraphy of the CO_2 storage site are defined. All coordinate systems are assumed to have units of meters and are defined by the reservoir component used. Model parameters for the stratigraphy and appropriate components are defined by either assigning a fixed value or random distribution to vary over. For the LHS analysis parameters defined with a distribution will be sampled from that distribution. For forward simulation all parameters should be specified with a fixed value. See the *Stratigraphy Component* section of this document for a list of all available parameters and their definitions.

2.3.1 Adding Component Models

The NRAP-Open-IAM is designed in a way so that only the components of interest need to be modeled in the system. Generally, a simulation will be built from the deepest component upward (reservoir, wellbore, aquifer, etc.). To add a component, first give it a name (each component must have a unique name). Next select the type of component model to be used. When adding subsequent components, a connection to existing components can be specified.

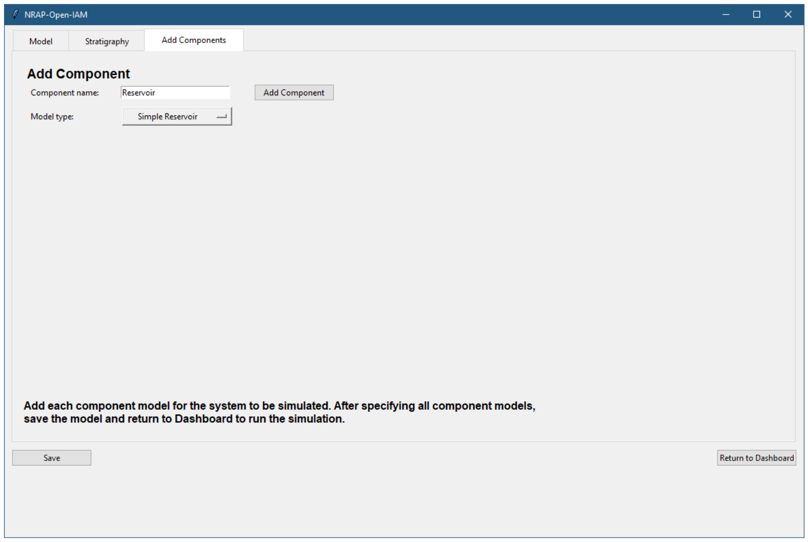


Fig. 2.7: Adding a Component Model

Each component model has component-specific input parameters and outputs. Parameters can be specified to be sampled from different distributions, or take on default values. When running a forward model parameters should only be specified as fixed values. When running a parameter study the parameters to vary should be specified as having a uniform distribution and minimum and maximum values. For stochastic simulations, any distributions can be specified. Parameter and output definitions can be found in the specific component model parameter section.

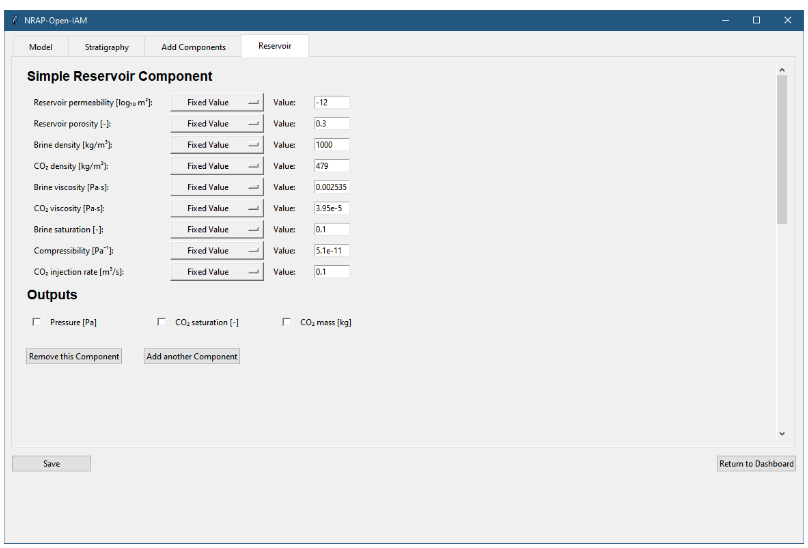


Fig. 2.8: Setup of Reservoir Component

If a component is specified that needs input from another component but the deeper component is not to be part of the model (i.e. specifying a wellbore model without a reservoir model), dynamic parameters can be used for the component model input. For dynamic parameters a value must be specified for each time step in the simulation. Values can be entered manually separated by a comma, or entered by providing path to the file containing the data. Some components require specification of which layer in the stratigraphy they represent (such as an aquifer model).

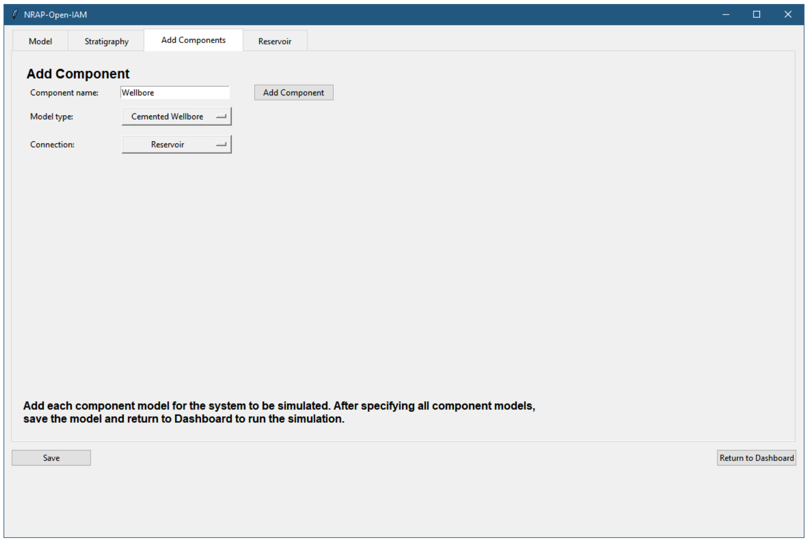


Fig. 2.9: Adding Second Component

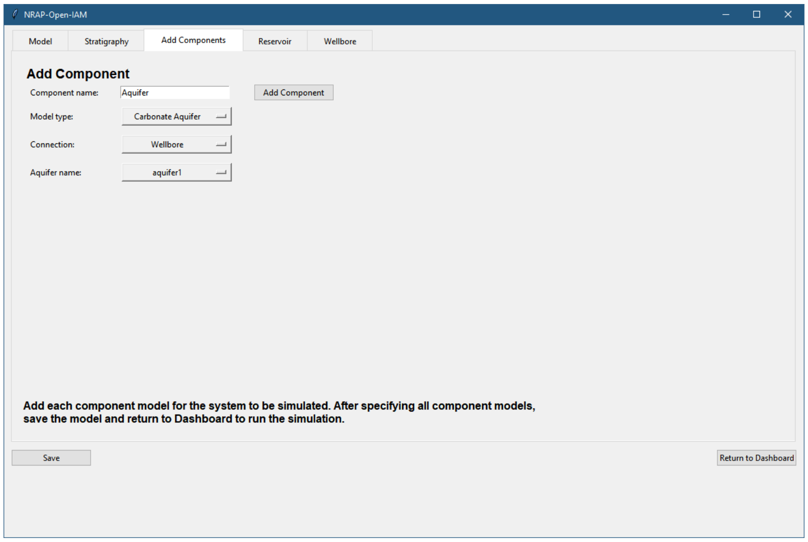


Fig. 2.10: Adding a Component Model with Connection and a Stratigraphy Selection

After a given component is specified, subsequent component can be added to the system model. When all required components have been added, save the model and return to the dashboard. The system model can then be run using the **RUN SIMULATION** button on the main dashboard.

2.3.2 GUI Setup Examples

In the folder *examples*, there is a subfolder *GUI_Files* with example simulation files that can be loaded into the GUI and run by the NRAP-Open-IAM. To run one of the provided examples select **Load Simulation** on the main dashboard of the GUI. In the file browser that appears, navigate to the *GUI_Files* subfolder of the *examples* folder and select the first example file *01_Forward_AR_CW.OpenIAM*. This example runs a simple forward model with a `AnalyticalReservoir` component providing an input to a `CementedWellbore` component. When the file is loaded into the GUI, the parameters of the simulation can be investigated. After the simulation is complete the user can proceed to the post-processing step (by clicking **Post Processing** on the main dashboard of the GUI) to visualize and, for some scenarios, analyze the obtained results. Post Processing tab has a folder selection button which allows user to select (output) folder containing results of simulation. Note that the selection of the folder (and loading of results) might fail if the simulation did not finish successfully. In this case it is recommended to check file *IAM_log.txt* within the output folder containing useful (debug, info, warning or error) messages produced during the simulation. File names of the GUI setup examples distributed with the tool contain shortcuts that would help the user to figure out the featured components and type of analysis.

The second example file *02_LHS_AR_MSW.OpenIAM* is a stochastic simulation of system model containing a `AnalyticalReservoir` and a `MultisegmentedWellbore` components. Example illustrates Latin hypercube sampling approach applied to the parameters of the setup model. The number of realizations run is 30.

The third example file *03_LHS_LUT_MSW.OpenIAM* illustrates use of a `LookupTableReservoir` and `MultisegmentedWellbore` components. The lookup tables data set utilized in the example for the `LookupTableReservoir` component is based on the simulation for Kimberlina oil field ([20]).

The fourth example file *04_LHS_DP_MSW.OpenIAM* illustrates Latin hypercube sampling approach applied to a `MultisegmentedWellbore` component. The pressure and CO_2 saturation required as inputs of the component are provided in the form of arrays. This form of input arguments is called dynamic parameters, i.e. parameters that change in time.

The system model setup in the fifth example file *05_LHS_AR_OW_CA.OpenIAM* illustrates application of three component models: `AnalyticalReservoir`, `OpenWellbore` and `CarbonateAquifer`. It estimates the impact the leakage of fluids through the wellbore has on the aquifer overlying the storage reservoir.

2.4 Control Files

Control files are a method of getting user input into the NRAP-Open-IAM for setting up a simulation scenario. Control files use a YAML format (extension *.yaml*). Any line in the control file starting with a pound sign (#) is a comment and is ignored by the program. The basic format of the control file is a parameter name followed by a colon, space, and a value. For objects with several parameters the object name is followed by a colon and the underlying parameters are listed on the consecutive lines tabbed in. For example, consider this partial file

```
#-------------------------------------------------
# NRAP-Open-IAM Control File example
#-------------------------------------------------
ModelParams:
    EndTime: 50
    TimeStep: 1.0
    Analysis: forward
    Components: [SimpleReservoir1,
                 CementedWellbore1]
    OutputDirectory: ../../output/output_ex1a_{datetime}
    Logging: Debug
Stratigraphy:
    numberOfShaleLayers:
```

(continues on next page)

(continued from previous page)

```
        vary: False
        value: 3
    # Thickness is in meters
    shale1Thickness:
        min: 500.0
        max: 550.0
        value: 525.0
    shale2Thickness:
        min: 450.0
        max: 500.0
        value: 475.0
    shale3Thickness:
        vary: False
        value: 11.2
    aquifer1Thickness:
        vary: False
        value: 22.4
    aquifer2Thickness:
        vary: False
        value: 19.2
    reservoirThickness:
        vary: False
        value: 51.2
```

Here, the first three lines are comments that are not read by the code. The fourth line defines the keyword `ModelParams` which describes parameters of the system model. The subsequent lines contain parameters of `ModelParams`. A `ModelParams` section is required in all NRAP-Open-IAM control files. The `EndTime` keyword defines the ending time for the simulation in years (50 years in the example). The `TimeStep` parameter defines the length of a time step (1 year in the example). The type of analysis being run is a `forward` (deterministic) simulation. Other possible options for `Analysis` parameter are `lhs` for Latin Hypercube Sampling analysis and `parstudy` for a parameter study. The `Components` parameter is a required entry that contains a list of component model names that are defined later in the file. The component list will always begin with a square bracket '[' followed by each of the component names that make up the system separated by a comma ',' and closed by a square bracket ']'. The names of the components are listed in the order they are supposed to be run. The next keyword `OutputDirectory` defines a directory for the output to be written into. The output directory can be appended with a keyword `{datetime}`. When a simulation is run, the `{datetime}` keyword will be replaced with the date and time of the simulation run. Note that the `{datetime}` keyword is optional: if it is omitted subsequent runs of the simulation will overwrite past results. That is, if there is a need to keep all results from re-running an NRAP-Open-IAM case, the `{datetime}` keyword will easily facilitate this; if re-running an NRAP-Open-IAM case should overwrite previous results, the `{datetime}` keyword should be omitted. In the output folder the NRAP-Open-IAM places a copy of the input file, all outputs from the component models written to text files, and *.png* images for all graphics. The last keyword `Logging` defines what level of logging information is written out to the logging files. Options for `Logging` levels are `Debug`, `Info`, `Warning`, and `Error`. `Info` is the default level (if no `Logging` keyword is given the logging will be set to `Info`) and will give you a valuable information about when parameters go outside of permitted ranges and when there are problems in the program. Debug is a good option if you have problems with your IAM model and want more information to explore the causes.

The next keyword section of the file is the required `Stratigraphy` section. In this section any model parameters related to the stratigraphy of the CO_2 storage site is defined. Any parameters for the stratigraphy are defined here with either a deterministic value or a range to vary over. A fixed value of any given parameter can be specified with the `vary: False` and `value: ###` specification shown here or simply `parameterName: ###`. The `min` and `max` specification gives a range for the parameter to vary over if an analysis is done over multiple realizations. See the *Stratigraphy Component* section of this document for a list of all available parameters.

The next sections of the input file defines every component model in the component model list specified earlier in the control file. The first component listed in the example is SimpleReservoir1 defined as follows

```
#-------------------------------------------------
# SimpleReservoir1 is a user defined name for component;
# the type SimpleReservoir is the ROM model name
#-------------------------------------------------
SimpleReservoir1:
    Type: SimpleReservoir
    Parameters:
        injRate: 0.1
    Outputs: [pressure,
              CO2saturation]
```

This section of the file defines a `SimpleReservoir` Component model named *SimpleReservoir1* to be part of the system model. The name *SimpleReservoir1* can be replaced with any other name defined by user, but will not be a part of the system model unless it is an element of the components list described in the previous section `ModelParams`. The `Type` is a keyword that defines the component model to be used and must match up with one of the component models currently available in the NRAP-Open-IAM. The `Parameters` section defines parameters of the component model. Description of parameters available for the user to specify can be found in the *Components Description* chapter of the current documentation. The component model parameters are specified in the same fashion as the `Stratigraphy` parameters. The `Outputs` specifies the observations of the component model that will be output from the simulation. Please refer to the *Components Description* chapter of this document to see which parameters and outputs are available for user specification in the control file.

Generally, dynamic (time-varying) input to component models comes from the output of other connected component models (e.g., the pressure and saturation as an input to a wellbore leakage model comes from a reservoir model). In some instances there may be a need to study a component model without the other attached component models feeding the input. In this case dynamic input can be specified with the `DynamicParameters` keyword. Under the `DynamicParameters` section each input name is specified followed by a list of values (enclosed in square brackets []) of the same length as the number of time points (a value for each time point, including an initial value). See files *ControlFile_ex7a.yaml* and *ControlFile_ex7b.yaml* for example of control files utilizing dynamic input for some components.

The next section of the input file is similar to the previous section and defines the next component model *CementedWellbore1*.

```
#-------------------------------------------------
CementedWellbore1:
    Type: CementedWellbore
    Connection: SimpleReservoir1
    Number: 4
    Locations:
        coordx: [100, 540]
        coordy: [100, 630]
    RandomLocDomain:
        xmin: 150
        xmax: 250
        ymin: 200
        ymax: 300
    Parameters:
        logWellPerm:
            min: -14.0
            max: -12.0
            value: -13.0
    Outputs: [CO2_aquifer1,
```

(continues on next page)

(continued from previous page)

```
                CO2_aquifer2,
                CO2_atm,
                brine_aquifer1,
                brine_aquifer2]
```

In this part of the example, `CementedWellbore` type component model is specified. There are four wellbores of this type being added with `Number: 4`: two of the locations are given in the `Locations` part and other two are generated randomly within the domain specified in `RandomLocDomain` part.

Unknown wellbore locations can be generated by specifying more wellbores (with `Number: ``) than the number of known wellbore locations. To control the location of the random well placement, a ``RandomLocDomain` section need to be used as

```
    RandomLocDomain:
        xmin: 150
        xmax: 250
        ymin: 200
        ymax: 300
```

This specification will limit the x-coordinate of random wells to be between 150 and 250, and the y-coordinate to be between 200 and 300. Sampling will be from a uniform distribution on the domain defined by `xmin` (`ymin`) and `xmax` (`ymax`).

All coordinate systems are assumed to have units of meters and are defined by the reservoir component used. Known wells will be placed first; after all known well coordinates are used wells will be placed within the random wells domain.

Known wellbore coordinates are entered as a comma separated list. There must be a comma between each coordinate. Random wellbores generated in an area when more wells are specified than number of known coordinates. After completing the model parameters proceed to the Stratigraphy tab.

For the SimpleReservoir component model the default injection location is at [0, 0]. For the Lookup Table based reservoir component the wellbore locations should fall within the domain of the reservoir simulations.

See *ControlFile_ex3.yaml* for additional example using random well placement and *ControlFile_ex4a.yaml* for example using only known well locations.

The last section of the input file is used to specify a graphical output

```
#-------------------------------------------------
# Plot setup part of the control file
#-------------------------------------------------
Plots:
    CO2_Leakage1:
        TimeSeries: [CO2_aquifer1]
        subplot:
            ncols: 2
            use: True
    CO2_Leakage2:
        TimeSeries: [CO2_aquifer2]
        subplot:
            ncols: 2
            use: True
    Pressure_plot:
        TimeSeries: [pressure]
        subplot:
```

(continues on next page)

(continued from previous page)

```
            ncols: 2
            use: True
            SimpleReservoir1_000.pressure: 'Pressure at well #1'
            SimpleReservoir1_001.pressure: 'Pressure at well #2'
            SimpleReservoir1_002.pressure: 'Pressure at well #3'
            SimpleReservoir1_003.pressure: 'Pressure at well #4'
        Title: Reservoir Pressure at Wellbore Location
```

Here, three plots are being requested. The firsts two plots will illustrate the CO_2 leakage to the shallow aquifer and the thief zone aquifer; the third plot will illustrate the pressures in the reservoir for the four wellbore locations specified earlier in the control file. *CO2_Leakage1*, *CO2_Leakage2* and *Pressure_plot* are the user defined names of the three plots to be created: these will also be used as the filenames of the figures saved in the output directory. `TimeSeries` is a keyword that instructs the program to plot the observation data as a time series plot. The values to be plotted (**CO2_aquifer1**, **CO2_aquifer2** and **pressure** above) have to be defined in the control file as outputs from one of the specified component models. Each plot will have a title corresponding to the values plotted. A user defined title can be specified with the `Title` keyword (as illustrated for the *Pressure_plot*) in the given plot section. For each aquifer the CO_2 leakage rates for all wells will be plotted on the same figure but on different subplot. If each observation is to be plotted on a separate subplot, the `subplot` keyword with `use` set to `True` must be specified, as illustrated in the example setup. Additionally, the `ncols` keyword (under `subplot` section) can be used to set the number of subplot columns to use. The number of rows is controlled by the number of different values (observations) to plot over the number of columns. Each subplot will be given a (default) title of the variable plotted unless specified by user. The default title names can be replaced with the user defined ones by using the full observation name as a key and the desired title as the value under `subplot` section as shown in the setup of *Pressure_plot*.

The example file described here can be found in the *examples/Control_Files* directory with the filename *ControlFile_ex1a.yaml*. To run this example, open a command prompt in the *examples/Control_Files* directory and run the command:

```
python ../../source/openiam/openiam_cf.py --file ControlFile_ex1a.yaml
```

Note: use \ on Windows and / on Mac and Linux.

Other example control files can be found in the same directory. They can be run by replacing the file name in the above command with the user specified one.

2.5 Output

Output is written to the folder specified in the model definition with the `Output directory`. If the path to the output directory is not absolute (i.e., does not containt the drive letter) it is assumed to start from the NRAP-Open-IAM root folder containing the tool distribution. For each component model of the system `Outputs` can be specified. When an output is specified for a forward model the values of that output are written to a file (*output_name.txt*) in the `Output Directory`. For a stochastic model (LHS or parametric study analysis) the outputs are included in the results file (*LHS_results.txt* or *parstudy_results.txt*) as well as a file for a statistical summary of the input parameters and output observations (*LHS_statistics.txt* or *parstudy_statistics.txt*). A copy of the input control file is also written to the `Output Directory` folder. Through the GUI, this input file can be loaded back in to rerun the simulation.

After a simulation is run, the post processing section of the GUI can be used to generate plots of the results and run sensitivity analysis. When the post processing page is first opened it asks for a folder specification. This folder is the output folder of the results you want to analyze. Navigate to that output folder using the **Browse** button.

2.5.1 Plotting

User can access post-processing capabilities of GUI by clicking on the **Post Processing** button on the main page. The **Post Processor** window will appear. After a folder containing the results of the simulation is selected different options for the simulation results appear that are already set for plotting. There are several types of plots that can be created depending on what type of simulation was run and what components were specified. The simplest plot is a `Time Series` plot where the output is plotted against time, multiple realizations will be plotted as separate lines. Specify a title to give the plot and a file name along with making a selection of what output to plot. Pressing the plot button will generate the plot, it will be saved with the filename given to the output directory for the results. If a simulation was run with multiple realizations a `Time Series Stats` plot or a `Time Series and Stats` plot will be options in the `Plot Type` menu. A `Time Series Stats` plot will shade the quadrants of the results along with plotting the mean and median results, but will not plot the individual realizations. A `Time Series and Stats` plot will overlay the realizations on the stats plots of the shaded quadrants. If AtmosphericROM component was included in the simulation, map-view plots of the plume for a single realization or probabilistic ensemble can be generated.

2.5.2 Sensitivity Analysis

If the simulation results are from a LHS simulation the `Processing` menu will have options for several types of sensitivity analysis. Note that while a sensitivity analysis can be run on simulations with a small number of realizations, the results will most likely be inaccurate. If the sensitivity coefficients do not sum to one, or if they vary largely through time, the number of realizations might need to be increased. Generally, 500 to 1000 realizations are needed for a sensitivity analysis. However, this might change depending on the complexity of the simulation. Each type of sensitivity analysis will produce plots and/or text file output in the output directory.

`Correlation Coefficients` option produces a plot matrix of either `Pearson` or `Spearman` correlation coefficients. Any system model observation can be excluded from the analysis if needed, although no exclusions need to be made.

`Sensitivity Coefficients` option calculates the sensitivity coefficients for each selected output to all inputs. Selecting multiple outputs will run the sensitivity coefficient calculation multiple times. The capture point is the index for point in time at which the sensitivity coefficient are to be calculated. The analysis produces a bar chart.

`Multiple Sensitivity Coefficients` option calculates the impact of input parameters on multiple outputs. Multiple outputs should be selected here. The capture point is the index for point in time at which the sensitivity coefficient are to be calculated. The analysis will produce a bar chart.

`Time Series Sensitivity` option will produce a line graph illustrating how the impact from input parameters changes over time with respect to an output value. Selecting multiple output values will run the analysis multiple times. The capture point determines the time at which the sensitivity coefficients are compared and then ordered based on the comparison.

2.6 Analysis Options in Control File

The NRAP-Open-IAM uses the Model Analysis ToolKit (MATK) [9] for the basis of its probabilistic framework. More information about MATK can be found here: http://dharp.github.io/matk/. The MATK code repository can be found here: https://github.com/dharp/matk.

Parameter input and output interactions can be explored using the *Analysis* section of the control file. Correlation coefficients can be calculated using the `CorrelationCoeff` keyword. Parameter sensitivity coefficients for any output simulation value can be calculated using a Random-Balanced-Design Fourier Amplitude Sensitivity Test (RBD-Fast) technique. The control file keywords `SensitivityCoeff`, `MultiSensitivities`, `TimeSeriesSensitivity` can be used to access different sensitivity coefficient outputs. See *ControlFile_ex8.yaml* for details on using the analysis section.

The Sensitivity Analysis is done with the SALib package [12]. For more information on the RBD-Fast technique see [31] and [28]. While not accessible through the control files, a scripting interface is provided to a Sobol sensitivity analysis [30].

This section will be expanded in the future.

2.7 Setup of visualization options

The plot types available within NRAP-Open-IAM are: TimeSeries, TimeSeriesStats, TimeSeriesAndStats, StratigraphicColumn, Stratigraphy, AoR, TTFD, GriddedMetric, GriddedRadialMetric, AtmPlumeSingle, and AtmPlumeEnsemble. We review the process of creating these plots and then discuss each plot type separately.

To create a figure using a simulation run with a *.yaml* control file, the the figure must be set up in the *Plots* section. Within the *Plots* section, the user can have multiple entries that each represent a different plot to be created. The names of these entries are defined by the user, and the names are generally used as the file name for the corresponding figure. An exception occurs with some plots. For example, with the TTFD plot type generates a large number of plots, and the final file names will depend on the input used (e.g., *TDS_Plume_Timings_Realization10.png*). The plot name provided can have extensions appended that specify the file type of the resulting figure file (e.g., *.png*, *.tiff*, or *.eps*). Below, we show two examples of TimeSeries plot entries in a *.yaml* control file.

```
Plots:
    Pressure_Figure:
        TimeSeries: [pressure]
    CO2_Sat_Figure.tiff:
        TimeSeries: [CO2saturation]
```

Since the first plot entry (*Pressure_Figure*) does not have an extension (e.g., *.png* or *.tiff*) appended at the end (e.g., *Pressure_Figure.tiff*), the produced figure will be of the default *.png* type. The second plot entry (*CO2_Sat_Figure.tiff*), however, includes *.tiff* at the end of the name; the resulting figure file will be of *.tiff* type. Note that if an extension is provided when using a plot type that generates names (e.g., the TTFD plot type), the generated names will still use the extension provided.

When we refer to an entry as being indented beneath another entry, we mean that in a *.yaml* file the indented entry is on a lower line and preceeded by additional four spaces. The indentation of one entry carries over to all entries contained within that entry. In the example above, for example, *Pressure_Figure:* is indented beneath `Plots:`, and `TimeSeries: [pressure]` is indented beneath *Pressure_Figure:*. `TimeSeries` is preceeded by eight spaces, while *Pressure_Figure* is preceeded by four spaces. Additionally, the name of each entry is followed by a colon (:). When entries have other entries indented beneath them, the colon will be the last character on that line (e.g., `Plots:` or `Pressure_Figure:`). When entries do not have other entries indented beneath them and instead take an input value or list, then the colon is followed by that input (e.g., `TimeSeries:  [pressure]`). Note that for the rest of this section, we will not include the colon when discussing an entry; it is assumed that each entry is followed by a colon.

All plot types have certain entries that are either required or optional. Some of these optional entries are used by multiple plot types. To avoid repeating the definitions of these entries, we first present the optional entries used by multiple plot types. Then, we review each plot type.

Each plot type has the optional entries FigureDPI and FigureSize.

- FigureDPI - the dots-per-inch (DPI) of the resulting figure(s) (default is 100). Most figure types produce only one figure file, but plot types like Stratigraphy and TTFD can produce multiple figures from one entry. Larger DPIs will create high-resolution, high-quality figures, but the file sizes are also larger. File size and quality is also influenced by the extension used (e.g., *.png* or *.tiff*). Recommended FigureDPI values are between 100 and 300.
- FigureSize - the width and height of the figure in inches. The width and height must be entered as a list of length two (e.g., `FigureSize:  [15, 8]`), with the width as the first number and the height as the second. The default

values for the plot types are: [13, 8] for `TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats`; [6, 10] for `StratigraphicColumn` plots; [12, 10] for `Stratigraphy` plots; [10, 8] for `AoR`, `TTFD`, `GriddedMetric`, and `GriddedRadialMetric` plots; and [15, 10] for `AtmPlumeSingle` and `AtmPlumeEnsemble` plots. This entry can also be provided as `figsize` instead of `FigureSize`.

The `TimeSeries`, `TimeSeriesStats`, `TimeSeriesAndStats`, `Aor`, `StratigraphicColumn`, and `Stratigraphy` plot types have the optional entry `Title`.

- `Title` - the text placed in bold at the top of the figure. If the `Title` entry is given for `StratigraphicColumn` or `Stratigraphy` plots, it replaces the default titles ('Stratigraphy for the Study Area' and 'Stratigraphy for the Study Area, Top of Each Unit Shown' for each plot type, respectively). For the remaining plot types, by default, there is no title for the overall figure, only titles for each individual subplot generated based on the output shown (with `AoR` plots having one subplot). Note that if one wants to have CO_2 in a title, one should enter it as 'CO$_2$' (the $$ symbols aid in the proper formatting).

Below , we show examples of `FigureDPI`, `FigureSize`, and `Title` entries used in a *.yaml* control file.

```yaml
Plots:
    Strat_Col:
        StratigraphicColumn:
            Title: Stratigraphy
            FigureDPI: 200
            FigureSize: [8, 12]
    Stratigraphy_Figure.tiff:
        Stratigraphy:
            Title: Stratigraphy
            FigureDPI: 300
            FigureSize: [13, 11]
    Pressure_Figure:
        TimeSeriesStats: [pressure]
        FigureDPI: 100
        FigureSize: [12, 8]
        Title: Pressure Results
    TDS_Volume_AoR.tiff:
        AoR: [Dissolved_salt_volume]
        FigureDPI: 200
        FigureSize: [12, 12]
        Title: AoR Results, Dissolved Salt Impact
    TTFD_Figures:
        TTFD:
            PlumeType: Dissolved_salt
            ComponentNameList: [GenericAquifer1, GenericAquifer2]
            FigureDPI: 200
            FigureSize: [12, 11]
    Gridded_Brine_Aquifer:
        GriddedMetric:
            ComponentNameList: [FaultFlow1]
            MetricName: brine_aquifer
            FigureDPI: 200
            FigureSize: [9, 9]
    Gridded_Salt_Mass_Frac:
        GriddedRadialMetric:
            ComponentNameList: [GenericAquifer1]
            MetricName: Dissolved_salt_mass_fraction
            FigureDPI: 200
```

(continues on next page)

(continued from previous page)

```
            FigureSize: [11, 9]
    Atmospheric_Plume_Single:
        AtmPlumeSingle:
            FigureDPI: 200
            FigureSize: [16, 12]
    Atmospheric_Plume_Probability.tiff:
        AtmPlumeEnsemble:
            FigureDPI: 300
            FigureSize: [14, 11]
```

Notice that the `FigureDPI` and `FigureSize` entries for the `StratigraphicColumn`, `Stratigraphy`, `TTFD`, `GriddedMetric`, `GriddedRadialMetric`, `AtmPlumeSingle`, and `AtmPlumeEnsemble` plots are indented under the plot type. In contrast, the `FigureDPI`,

`FigureSize`, and `Title` entries for the `TimeSeriesStats` and `AoR` plots are not

indented beneath the plot type. This discrepancy occurs because the `TimeSeriesStats` and `AoR` entries are followed by a metric (e.g., [**pressure**]), while the other plot type entries are not.

The `StratigraphicColumn` and `Stratigraphy` plot types both have the optional entries `ReservoirColor`, `ShaleColor`, `AquiferColor`, `ReservoirAlpha`, `ShaleAlpha`, `AquiferAlpha`, `ReservoirLabel`, `Shale#Label`, and `Aquifer#Label` (where # is a particular unit number). Note that the color and alpha entries containing `Shale` and `Aquifer` can be used with a number specifying a certain shale or aquifer (e.g., `Shale2Color` or `Aquifer1Alpha`). Without a specific number, these entries will apply to all shales or aquifers (excluding units that have their own, separate entry of the same type). Note that the color entries can be a string (e.g., `ReservoirColor: orange` or `Aquifer2Color: g`) or a list of length three representing fractions of red, green, and blue (`Aquifer2Color: [0.25, 0.25, 1]`). The entries containing `Alpha` set the alpha values used in the plot. Alpha values range from 0 to 1 and control transparency, with 1 being fully opaque and values approaching 0 becoming more transparent. For examples showing the use of these entries, see *ControlFile_ex33b*. To prevent a label from being shown, a label entry can be given as ‘’ (e.g., `Shale2Label: ''`). One might not want a label if the setup causes overlap between different labels.

- `ReservoirColor` - the color used when plotting the reservoir. The default is [0.33, 0.33, 0.33].
- `ShaleColor` - the color used when plotting all shales (or a specific shale, if given as `Shale#Color`, where # is an appropriate unit number). The default is red.
- `AquiferColor` - the color used when plotting all shales (or a specific shale, if given as `Aquifer#Color`, where # is an appropriate unit number). The default is blue.
- `ReservoirAlpha` - the alpha value used when plotting the reservoir. The default value is 0.5.
- `ShaleAlpha` - the alpha value used when plotting all shales (or a specific shale, if given as `Shale#Alpha`, where # is an appropriate unit number). The default value is 0.25.
- `AquiferAlpha` - the alpha value used when plotting all aquifers (or a specific aquifer, if given as `Aquifer#Alpha`, where # is an appropriate unit number). The default value is 0.25.
- `ReservoirLabel` - the label displayed for the reservoir. In `StratigraphicColumn` plots, the default label is “Reservoir Thickness: H m,” where H is the unit thickness. In `Stratigraphy` plots, the default label is “Reservoir.”
- `Shale#Label` - the label displayed a specific shale, where # is an appropriate unit number. In `StratigraphicColumn` plots, the default label is “Shale # Thickness: H m,” where H is the unit thickness. In `Stratigraphy` plots, the default label is “Shale #.”
- `Aquifer#Label` - the label displayed a specific aquifer, where # is an appropriate unit number. In `StratigraphicColumn` plots, the default label is “Aquifer # Thickness: H m,” where H is the unit thickness. In `Stratigraphy` plots, the default label is “Aquifer #.”

The `Stratigraphy`, `TTFD`, `GriddedMetric`, `GriddedRadialMetric`, `AtmPlumeSingle`, and `AtmPlumeEnsemble` plot types all have the optional entries `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SpecifyXandYLims`, and `SaveCSVFiles`.

- `PlotInjectionSites` - an option to plot injection sites (default is `False`). The only acceptable values are `True` or `False`.
- `InjectionCoordx` - value or list of values for the x coordinate(s) of injection site(s) (default is None). The value(s) are in meters. This entry must be provided when using a `LookupTableReservoir`, as that component type does not have an .injX attribute. Other reservoir types like `SimpleReservoir` or `AnalyticalReservoir` can be displayed without an InjectionCoordx entry.
- `InjectionCoordy` - value or list of values for the y coordinate(s) of injection site(s) (default is None). The value(s) are in meters. This entry must be provided when using a `LookupTableReservoir`, as that component type does not have an .injY attribute. Other reservoir types like `SimpleReservoir` or `AnalyticalReservoir` can be displayed without an InjectionCoordy entry.
- `SaveCSVFiles` - an option to save results in *.csv* files. The only acceptable values are `True` or `False`. The default value for `AoR`, `TTFD`, `GriddedMetric`, `GriddedRadialMetric`, `AtmPlumeSingle`, and `AtmPlumeEnsemble` plots is `True`, while the default value for `Stratigraphy` plots is `False`. For `Stratigraphy` plots, the *.csv* files contain unit thicknesses and depths across the domain. The *.csv* files are not saved by `Stratigraphy` plots when the simulation uses the `LookupTableStratigraphy` option.

If set up, `SpecifyXandYLims` is a dictionary containing two entries: `xLims` and `yLims` (i.e., `xLims` and `yLims` are indented beneath``SpecifyXandYLims`` in a *.yaml* file).

- `SpecifyXandYLims` - a dictionary containing two optional entries related to the limits of the figure's x and y axes (default is None). Within this dictionary are the entries `xLims` and `yLims`.
- `xLims` - an entry under `SpecifyXandYLims` containing a list of length two that represents the x-axis limits (e.g., `xLims: [0, 1000]`; default is None). The values are in meters. The first and second values in the list are the lower and upper limits, respectively. If `xLims` is not provided or provided incorrectly, the figure will use the default approach for setting the x-axis limits.
- `yLims` - an entry under `SpecifyXandYLims` containing a list of length two that represents the y-axis limits (e.g., `yLims: [0, 1000]`; default is None). The values are in meters. The first and second values in the list are the lower and upper limits, respectively. If `yLims` is not provided or provided incorrectly, the figure will use the default approach for setting the y-axis limits.

The `Stratigraphy`, `TTFD`, and `AtmPlumeEnsemble` plots also have the optional entry `SpecifyXandYGridLims`, which is a dictionary containing the `gridXLims` and `gridYLims` entries. `AoR` plots do not have grid entries because the x and y values used are those of the `OpenWellbore` components. `GriddedRadialMetric` plots use the radial grids produced by a component (e.g., a `GenericAquifer` component), while `GriddedMetric` plots use only the locations corresponding with the output.

- `SpecifyXandYGridLims` - a dictionary containing two optional entries related to the x and y limits for the gridded data evaluated (default is None). In `Stratigraphy` plots, the gridded data are the three-dimensional planes depicting the the top of each unit. For `TTFD` and `AtmPlumeEnsemble` plots, the gridded data are the color-labelled values. Within this dictionary are the entries `gridXLims` and `gridYLims`.
- `gridXLims` - an entry under `SpecifyXandYGridLims` containing a list of length two that represents the x-axis limits for the grid used to evaluate results (e.g., `gridXLims: [100, 900]`; default is None). The values for `gridXLims` are in meters. The first and second values in the list are the lower and upper limits, respectively. If `gridXLims` is not provided or provided incorrectly, the figure will use the default approach for creating the gridded values.
- `gridYLims` - n entry under `SpecifyXandYGridLims` containing a list of length two that represents the y-axis limits for the grid used to evaluate results (e.g., `gridYLims: [100, 900]`; default is None). The values for `gridYLims` are in meters. The first and second values in the list are the lower and upper limits, respectively.

If `gridYLims` is not provided or provided incorrectly, the figure will use the default approach for creating the gridded values.

The `Stratigraphy`, `TTFD`, and `AtmPlumeEnsemble` plot types can all use the optional entries `xGridSpacing` and `yGridSpacing`:

- `xGridSpacing` - a horizontal distance (m) used as the interval between the grid points in the x-direction (default is None). If this entry is not setup, the x-coordinates of the grid points are defined using a default approach (1/100th of the range in x-values).
- `yGridSpacing` - a horizontal distance (m) used as the interval between the grid points in the y-direction (default is None). If this entry is not setup, the y-coordinates of the grid points are defined using a default approach (1/100th of the range in y-values).

The `AoR`, `GriddedMetric`, and `GriddedRadialMetric` plot types have the optional entry `TimeList`:

- `TimeList` - a list specifying the times (in years) for which to create separate figures (e.g., `TimeList: [1, 5, 10]). Otherwise, one figure can be created for each timestep by having ``TimeList: All`. If TimeList is not entered for an `AoR` plot, the figures created will show the maximum values for all locations across all model times. If `TimeList` is not entered for a `GriddedMetric` or `GriddedRadialMetric` plot, the default setting is `TimeList: All`.

The `TTFD`, `GriddedMetric`, and `GriddedRadialMetric` plot types all have the required entry `ComponentNameList`:

- `ComponentNameList` - a list containing the names provided for each of the components producing output to be used for the creation of the figures (e.g., `ComponentNameList: [FutureGen2AZMI1, FutureGen2AZMI2]` in *ControlFile_ex40.yaml*). Below, we show a section of the *.yaml* file for *ControlFile_ex40.yaml*. This section demonstrates where the name is provided for the *FutureGen2AZMI2* component. Below the excerpt is an example of how component names are set when using NRAP-Open-IAM in a script application.

Excerpt from *ControlFile_ex40* demonstrating how an aquifer component is given the name FutureGen2AZMI2:

```
FutureGen2AZMI2:
    Type: FutureGen2AZMI
    Connection: MultisegmentedWellbore1
    AquiferName: aquifer3
    Parameters:
        por: 0.132
        log_permh: -12.48
        log_aniso: 0.3
        rel_vol_frac_calcite: 0.1
    Outputs: [pH_volume, TDS_volume, Dissolved_CO2_volume,
              Dissolved_CO2_dx, Dissolved_CO2_dy, Dissolved_CO2_dz]
```

Example of setting the component name (*FutureGen2AZMI2*) in a script application:

```
fga = sm.add_component_model_object(FutureGen2AZMI(name='FutureGen2AZMI2', parent=sm))
```

The `GriddedMetric` and `GriddedRadialMetric` plot types both have the required entry `MetricName`:

- `MetricName` - the name of the metric to plot. For a `GriddedMetric` plot, the `SealHorizon` and `FaultFlow` outputs **CO2_aquifer**, **brine_aquifer**, **mass_CO2_aquifer**, or **mass_brine_aquifer** can be provided for `MetricName`. For a `GriddedRadialMetric` plot, the `GenericAquifer` outputs **Dissolved_CO2_mass_fraction** or **Dissolved_salt_mass_fraction** can be provided for `MetricName`. When plotting these ``GenericAquifer` metrics, the component used must also produce **r_coordinate** and **z_coordinate** outputs.

The `GriddedMetric`, `GriddedRadialMetric`, and `AtmPlumeSingle` plot types have the optional entry `Realization`:

- `Realization` - the realization number for which to display results (default is 0). Note that this optional input is only used in simulations using Latin Hypercube Sampling (`lhs`) and Parameter Study (`parstudy`) analysis types. This input uses the indexing rules in Python, where 0 represents the first realization and (N - 1) represents the last (where N is the number of realizations).

The `GriddedMetric` and `GriddedRadialMetric` plot types both have the optional entry `EqualAxes`:

- `EqualAxes` - the option to force the x and y axes to cover the same distances (for an equal aspect ratio). The acceptable values are `True` or `False`, and the default value is `True`. If set to `True`, the axes limits given with `xLims` and `ylims` will not be used.

Examples of setting up each plot type in a *.yaml* file are shown in the sections below.

2.7.1 TimeSeries, TimeSeriesStats, and TimeSeriesAndStats

The `TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats` plot types are used to display results varying over time. Although this section covers three plot types, these plot types are different variations of the same type of plot.

`TimeSeries` plots are line plots of results varying over time. The number of lines in the resulting figure depends on the setup of the scenario. For example, components and associated locations entered in the *.yaml* file can define the number of curves shown in the figure but only the components that produce the metric being plotted (e.g., **pressure** or **brine_aquifer1**) influence the number of lines created for that particular metric.

`TimeSeriesStats` and `TimeSeriesAndStats` plots can only be produced for simulations using the Latin Hypercube Sampling (LHS, `lhs` in a control file setup) or Parameter Study (`parstudy` in a control file setup) analysis types (not the `forward` analysis type). Simulations using the `lhs` and `parstudy` analysis types create separate simulations (i.e., different realizations) that explore the parameter space. The parameters varied are those entered with minimum and maximum values, which are meant to model uniform distribution. Consider, for example, a `TimeSeriesStats` plot set up for an LHS run with 30 realizations. The `ModelParams` section of the *.yaml* file would be similar to this excerpt from *ControlFile_ex4a.yaml*:

```
ModelParams:
    EndTime: 10
    TimeStep: 1.0
    Analysis:
        Type: lhs
        siz: 30
    Components: [SimpleReservoir1,
                 OpenWellbore1,
                 CarbonateAquifer1]
    OutputDirectory: output/output_ex4a_{datetime}
    Logging: Debug
```

The entries `Type:  lhs` and `siz:  30` under `Analysis` specify the run as an LHS simulation with 30 realizations. Each realization will use different values for the parameters that are setup to vary. In a `TimeSeries` plot, the outputs for each realization will be represented by separate lines.

If an LHS or parstudy simulation uses many realizations and many component locations, the `TimeSeries` plot could become visually unclear. To avoid a lack of visual clarity, `TimeSeriesStats` plots show the basic information about the distribution of results. The plot produces lines representing mean and median values as well as shaded regions showing the four quartiles of the distribution varying over time (0th to 25th, 25th to 50th, 50th to 75th and 75th to 100th percentiles).

`TimeSeriesAndStats` plots combine the approaches of `TimeSeries` and `TimeSeriesStats` plots. The mean, median, and quartiles are shown along with line graphs for each realization.

`TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats` plots can have the following optional entries: `UseMarkers`, `VaryLineStyles`, `UseLines`, `Subplot`, `Title`, `FigureDPI`, and `FigureSize` (the latter three are described above). Note that `Subplot` is a dictionary containing the optional entries `Use` and `NumCols` (i.e., the `Use` and `NumCols` options are on a line beneath `Subplots` and preceeded by four more spaces).

- `UseMarkers` - an option to show results with values annotated with markers like circles and squares (default is `False`). The only acceptable values are `True` or `False`. If markers are used, the colors of markers and lines will vary in the normal manner (i.e., a rotation through the default matplotlib color order).
- `VaryLineStyles` - an option to vary the line styles used (default is `False`). The only acceptable values are `True` or `False`. The matplotlib line styles used are 'solid', 'dotted', 'dashed', and 'dashdot'. Line colors will still vary in the normal manner.
- `UseLines` - an option to show results with lines (default is `True`). The only acceptable values are `True` or `False`. If neither markers nor lines are used, the plot will not show any results. One should only set `UseLines` to `False` if `UseMarkers` is set to `True`. If `UseLines` is set to `False`, `VaryLineStyles` will automatically be set to `False`, regardless of the entry provided in the *.yaml* file.
- `Subplot` - a dictionary containing the optional entries `Use` and `NumCols`. This entry can also be provided as `subplot`.
- `Use` - the option to use multiple subplots (`True`) or not (`False`). The defalt value is `True`. The different subplots can show the results for different locations and/or the results for different metrics (e.g., **pressure** in one subplot, **CO2saturation** in another). If different types of output (e.g., **pressure** and **CO2saturation**) are included in a `TimeSeries` plot but `Use` is set to `False`, the y-axis label will only reflect one of the two output types. This entry can also be provided as `use`.
- `NumCols` - the number of columns used to set up the subplots, if `Use` is set to `True`. If the plot includes 3 or fewer metrics (influenced by the output type(s) and locations used), the default value is 1. If the plot includes 4 or more metrics, the default value is 2. This entry can also be given as `ncols`. The number of rows is taken as the number required to plot the metrics used by the plot (given the number of columns). The number of metrics is set by the number of output types given (e.g., two for `TimeSeries: [pressure, CO2saturation]`) and the number of locations for those output types. For example, if `NumCols` is two, then the number of rows will be half the number of metrics (rounding up to the next integer). Note that `TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats` plots will automatically adjust the font sizes used to seek to avoid text overlapping with other features. Accomplishing a specific subplot configuration without the text becoming too small, for example, can be aided by also changing the `FigureSize` entry (see above).

These optional entries are not indented under `TimeSeries` or `TimeSeriesAndStats` in a *.yaml* file, but are instead indented under the figure name. If `UseMarkers`, `VaryLineStyles`, or `UseLines` are provided for a `TimeSeriesStats` plot, the entries will have no effect (i.e., they do not influence the mean and median lines or the shaded quartiles).

The titles for individual subplots are generated based on the metric shown in the subplot (i.e., output type and location the output was produced for). One can specify the subplot title that will correspond to a specific output, however, by entering the output name under `Subplot`. The output name depends on the corresponding component, the output type, and the location index. For example, a `SimpleReservoir` component named SimpleReservoir1 producing pressure at location 0 will result in an output name of `SimpleReservoir1_000.pressure`. The component name is followed by an underscore, then the location index (starting at 0 and expressed with three digits), then a period, and finally the output name (e.g., **pressure**). The text used for the title is given after the output name as `ComponentName_000.OutputName: Text Used for Title`. For an example of this approach, see *ControlFile_ex1a*.

Below, we show examples of `TimeSeries` and `TimeSeriesAndStats` plots in a *.yaml* control file.

```yaml
Plots:
    Pressure_and_Sat:
        TimeSeries: [pressure, CO2saturation]
        UseMarkers: False
        UseLines: True
```

(continues on next page)

(continued from previous page)

```
        VaryLineStyles: True
        FigureDPI: 150
        Subplot:
            Use: True
            NumCols: 2
            SimpleReservoir1_000.pressure: 'Pressure at Well 0'
            SimpleReservoir1_001.pressure: 'Pressure at Well 1'
            SimpleReservoir1_000.CO2saturation: 'CO$_2$ Saturation at Well 0'
            SimpleReservoir1_001.CO2saturation: 'CO$_2$ Saturation at Well 1'
    Pressure_Stats:
        TimeSeriesAndStats: [pressure]
        UseMarkers: True
        UseLines: False
        VaryLineStyles: False
        FigureDPI: 400
```

For examples of `TimeSeries` plots, see control file examples 1a, 1b, 2, 3, 7a, 7b, and 14. For examples of `TimeSeriesStats` plots, see control file examples 4a, 4b, 6, 8, 15, and 39. For examples of `TimeSeriesAndStats` plots, see control file examples 4a, 14, and 40.

2.7.2 StratigraphicColumn

`StratigraphicColumn` plots show unit thicknesses at one location. If the simulation does not use spatially variable stratigraphy, then the unit thicknesses at the one location used are representative of the entire domain. For more details regarding the use of spatially variable stratigraphy, see the section for the `Stratigraphy` plot type below. When using spatially variable stratigraphy with the `LookupTableStratigraphy` approach, the `StratigraphicColumn` plot may be more visually clear than the `Stratigraphy` plot type.

The `StratigraphicColumn` plot type has the following optional entries: `XValue`, `YValue`, `DepthText`, `ReservoirColor`, `ShaleColor`, `AquiferColor`, `ReservoirAlpha`, `ShaleAlpha`, `AquiferAlpha`, `ReservoirLabel`, `ShaleLabel`, `AquiferLabel`, `FigureDPI`, `FigureSize`, and `Title`. All of these entries but `XValue`, `YValue`, and `DepthText` are described above.

- `XValue` - the x-coordinate (m) of the location used for the plot. The default value is is 0 m.
- `YValue` - the y-coordinate (m) of the location used for the plot. The default value is is 0 m.
- `DepthText` - option specifying whether to show depths at each unit interface (`True`) or not (`False`). The default value is `True`. One may want to disable the depth text if, for example, certain units are so thin that the text for different units plot on top of each other.

Two examples of `StratigraphicColumn` plots in a *.yaml* control file are shown below. One plot does not include any optional entries and, therefore, uses the default options. The other plot includes a variety of optional entries.

```
Plots:
    Strat_Col_Default:
        StratigraphicColumn:
    Strat_Col_With_Options.tiff:
        StratigraphicColumn:
            Title: Stratigraphy
            Shale1Label: Lower Aquitard
            Shale2Label: Middle Aquitard
            Shale3Label: Upper Aquitard
            Aquifer1Label: AZMI
```

(continues on next page)

(continued from previous page)

```
            Aquifer2Label: Freshwater Aquifer
            ReservoirLabel: Storage Reservoir
            ReservoirColor: darkmagenta
            Shale1Color: [0.33, 0.33, 0.33]
            Shale2Color: olive
            Shale3Color: rosybrown
            Aquifer1Color: deeppink
            Aquifer2Color: mediumturquoise
            Shale1Alpha: 0.3
            Shale2Alpha: 0.3
            Shale3Alpha: 0.45
            Aquifer1Alpha: 0.3
            Aquifer2Alpha: 0.45
            ReservoirAlpha: 0.45
            FigureDPI: 300
            XValue: 2000
            YValue: 2000
            DepthText: False
```

For more examples of `StratigraphicColumn` plots, see control file examples *ControlFile_ex33a-ControlFile_ex38*.

2.7.3 Stratigraphy

`Stratigraphy` plots are three-dimensional plots showing the specified stratigraphy as well as features like wellbores and injection sites. These plots can vary with the approach used for the stratigraphy. For example, a `strike` and `dip` can be assigned in the `Stratigraphy` section of a *.yaml* control file. Alternatively, the `LookupTableStratigraphy` option allows one to create the domain's stratigraphy with a *.csv* file containing unit thicknesses. `Stratigraphy` plots also work for simulations with spatially uniform unit thicknesses.

First, we discuss the use of a `strike` and `dip` options. The `Stratigraphy` section from *ControlFile_ex33a.yaml* is shown below:

```
Stratigraphy:
    spatiallyVariable:
        strikeAndDip:
            strike: 315
            dip: 5
            dipDirection: NE
            coordxRefPoint: 1200
            coordyRefPoint: 1200
    numberOfShaleLayers:
        vary: False
        value: 3
    shale1Thickness:
        value: 750.0
        vary: False
    shale2Thickness:
        value: 950.0
        vary: False
    shale3Thickness:
        value: 200
        vary: False
```

(continues on next page)

(continued from previous page)

```
    aquifer1Thickness:
        vary: False
        value: 200
    aquifer2Thickness:
        vary: False
        value: 200
    reservoirThickness:
        vary: False
        value: 150
```

To set up spatially variable stratigraphy, one can use `spatiallyVariable` keyword indented under `Stratigraphy`. To use strike and dip values, the `strikeAndDip` keyword needs to be indented under `spatiallyVariable`. The entries indented under `strikeAndDip` are as follows:

- `strike` - the strike of the units in degrees clockwise from north in a map view presentation. For example, strike values of 0 or 180 make the units strike north/south; strike values of 90 or 270 make the units strike east/west, and strike values of 30 or 210 make the units strike northeast/southwest. Acceptable values are in a range between 0 to 360.
- `dip` - the dip of the units in degrees, where a positive value corresponds with unit depths increasing in the `dipDirection` provided. Acceptable values range from 0 to less than 90.
- `dipDirection` - the dip direction provided in a cardinal direction - N, E, S, W, NE, SE, SW, or NW. Note that this entry must be compatible with the `strike` entry. For example, units cannot strike north/south and dip to the north, but they could strike north/south and dip to the east or west.
- `coordxRefPoint` - the x-coordinate (m) of the reference point. The unit thicknesses provided for the reference point are used to calculate unit thicknesses across the domain.
- `coordyRefPoint` - the y-coordinate (m) of the reference point. The unit thicknesses provided for the reference point are used to calculate unit thicknesses across the domain.

Note that the unit thicknesses indented under `Stratigraphy` are those at the reference point (x = `coordxRefPoint`, y = `coordyRefPoint`). When using the `strikeAndDip` option, unit thicknesses in other parts of the domain are calculated in relation to this reference point. Other `Stratigraphy` component parameters like *numberOfShaleLayers* and *datumPressure* cannot vary across the domain. Note that units can effectively pinch out, although the thicknesses will only be reduced to the minimum value of 1 m. Additionally, while the `strike` and `dip` option will make some units thicker (e.g., increasing the thickness of the the top shale so that the units beneath it have greater depths), each unit thickness cannot exceed the maximum value of 1600 m.

To use the `LookupTableStratigraphy` approach, one can use `spatiallyVariable` indented under `Stratigraphy` and then `LookupTableStratigraphy` keyword indented under `spatiallyVariable`. This approach is demonstrated in *ControlFile_ex38.yaml*:

```
Stratigraphy:
    spatiallyVariable:
        LookupTableStratigraphy:
            FileName: 'stratigraphy.csv'
            FileDirectory: 'examples/Control_Files/input_data/ex38'
            MaxPointDistance: 100
```

The entries indented under `LookupTableStratigraphy` are as follows:

- `FileName` - the name of the *.csv* file containing unit thicknesses and other `Stratigraphy` component parameters (*numberOfShaleLayers*, *datumPressure*, and *depth*).
- `FileDirectory` - the directory containing the *.csv* file referenced by *FileName*. The directory is given in relation

to the main directory used for the NRAP-Open-IAM installation being used but `FileDirectory` can also provide an entire path name like

C:UsersUserNameDocumentsNRAPOpenIAMexamplesControl_Filesinput_dataex38.

- `MaxPointDistance` - to set unit thicknesses at each location evaluated in the domain, each location must be within a certain distance of a point in the *.csv* file referenced with `FileName`. `MaxPointDistance` is that maximum distance (m) (default is 100 m). If a location in the domain is not close enough to a point in the *.csv* file, the simulation will return an error. Users can avoid this error by setting `MaxPointDistance` to a higher value, but using too high a value could lead to inaccurate depictions of the domain's stratigraphy. `MaxPointDistance` is intended to help ensure that `LookupTableStratigraphy` *.csv* files include sufficient information. It is the user's responsibility to make sure that the *.csv* file contains sufficient information and the `MaxPointDistance` is not too high.

The first two columns of a `LookupTableStratigraphy` *.csv* file are x and y coordinates (m) with the columns named 'x' and 'y', respectively. Any unit thicknesses (m) that vary with x and y values should be listed in columns with the same number of rows as the x and y columns. The thicknesses specified in a particular row of the *.csv* file correspond to the x and y values from the same row. If a unit thickness does not vary with x and y values, that unit thickness can be displayed in a column with a single row. A location in the domain will be assigned the unit thicknesses from the closest location in the `LookupTableStratigraphy` *.csv* file - if the closest location is within `MaxPointDistance` of the location. For an example, see the *stratigraphy.csv* file in the directory *examples/Control_Files/input_data/ex38*.

Note that `Stratigraphy` plots created for simulations using `LookupTableStratigraphy` will not have three-dimensional planes. Instead, the tops of each unit are plotted as squares along each wellbore.

`Stratigraphy` plots can have the following optional entries: `PlotWellbores`, `PlotWellLabels`, `WellLabel`, `PlotInjectionSites`, `PlotInjectionSiteLabels`, `InjectionCoordx`, `InjectionCoordy`, `PlotStratComponents`, `StrikeAndDipSymbol`, `SpecifyXandYLims`, `SpecifyXandYGridLims`, `xGridSpacing`, `yGridSpacing`, `View`, `SaveCSVFiles`, `ReservoirColor`, `ShaleColor`, `AquiferColor`, `ReservoirAlpha`, ShaleAlpha``, AquiferAlpha``, `ReservoirLabel`, `Shale#Label`, `Aquifer#Label`, `FigureDPI`, `FigureSize`, and `Title`. Four of these entries (`StrikeAndDipSymbol`, `SpecifyXandYLims`, `SpecifyXandYGridLims`, and `View`) are dictionaries containing additional entries (i.e., more entries indented beneath them in a *.yaml* file). The entries `SpecifyXandYLims`, `SpecifyXandYGridLims`, `xGridSpacing`, `yGridSpacing`, `SaveCSVFiles`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `ReservoirColor`, `ShaleColor`, `AquiferColor`, `ReservoirAlpha`, ShaleAlpha``, AquiferAlpha``, `ReservoirLabel`, `Shale#Label`, `Aquifer#Label`, `FigureDPI`, and `FigureSize` are described above.

- `PlotWellbores` - an option to plot wellbores as vertical lines (default is `True`). The only acceptable values are `True` or `False`.
- `PlotWellLabels` - an option to show text labels specifying wellbore types and numbers (default is `True`). If `WellLabel` is not entered, labels will be set according to the wellbore component type. For example, the labels could be "Open Wellbore 1" for an Open Wellbore, "M.S. Wellbore 1" for a MultiSegmented Wellbore, or "Cemented Wellbore 1" for a Cemented Wellbore. If `WellLabel` is entered, the text provided will be used. The only acceptable values are `True` or `False`.
- `WellLabel` - the label used for wellbores if `` PlotWellLabels``is set to ``True ``. If the text given includes empty brackets (*{}*), then the location index will be inserted in that position. If this entry was given as `WellLabel:  Legacy Well {}`, for example, then the labels would range from "Legacy Well 0" to "Legacy Well (N - 1)," where N is the maximum location index for the wellbore components (location indices use the python indexing). If `WellLabel` is given without brackets, then the same text will be displayed for each wellbore component (e.g., `WellLabel:  Well`). if `` PlotWellLabels``is set to ``True `` but `WellLabel` is not entered, labels will be set using the default approach.
- `PlotInjectionSiteLabels` - an option to show a text label for the injection site(s) (default is `False`).
- `PlotStratComponents` - the option to plot squares along each wellbore at the depths at which the wellbore intersects the top of a unit (default is `False`). The tops of shales are shown with red squares, while the tops of aquifers are shown with blue squares. The only acceptable values are `True` or `False`.

- `StrikeAndDipSymbol` - a dictionary containing four optional entries related to the strike and dip symbol shown in the figure (default is None). Within this dictionary are the entries `PlotSymbol`, ``coordx, coordy, and length.
- `PlotSymbol` - an entry under `StrikeAndDipSymbol` that specifies whether to show the strike and dip symbol (default is `True`). The only acceptable values are `True` or `False`.
- `coordx` - an entry under `StrikeAndDipSymbol` that specifies the x-coordinate at which to plot the strike and dip symbol (default is None). If `coordx` is not setup, the graph will use a default location (which depends on the domain).
- `coordy` - an entry under `StrikeAndDipSymbol` that specifies the y-coordinate at which to plot the strike and dip symbol (default is None). If `coordy` is not setup, the graph will use a default location (which depends on the domain).
- `length` - an entry under `StrikeAndDipSymbol` that specifies the length scale (m) of the strike and dip symbol (default is None). For flat-lying units, the length is the diameter of the circular symbol used. For dipping units, the length applies to the line going in direction of strike (not the line in the dip direction). If `length` is not provided, the graph will use a calculated length (which depends on the domain).
- `View` - a dictionary containing two optional entries related to the perspective of the three-dimensional graph (default is None). Within this dictionary are the entries `ViewAngleElevation` and `ViewAngleAzimuth`. A separate version of the figure is created for each combination of the `ViewAngleElevation` and `ViewAngleElevation` entries, where the first values in the keywords list are used for the same graph and so on.
- `ViewAngleElevation` - an entry under `View` containing a list of the elevation angles (in degrees) to use in the `Stratigraphy` plot(s) (default is [10, 30]). Values must be between -90 and 90. See the matplotlib documentation regarding view angles. This list must have the same length as the `ViewAngleAzimuth` list.
- `ViewAngleAzimuth` - an entry under `View` containing a list of the azimuth angles (in degrees) to use in the `Stratigraphy` plot(s) (default is [10, 30]). Values must be between 0 and 360. See the matplotlib documentation regarding view angles. This list must have the same length as the `ViewAngleElevation` list.

Two examples of *.yaml* entries for `Stratigraphy` plots are shown below. The first entry uses the default settings, while the second entry specifies each option. Since the simulation uses a `LookupTableReservoir`, the entry has to include `InjectionCoordx` and `InjectionCoordy`. `InjectionCoordx` and `InjectionCoordy` are not required when using another type of reservoir component with option `PlotInjectionSites: True`.

```
Plots:
    Strat_Plot_Default_Settings:
        Stratigraphy:
    Strat_Plot.tiff:
        Stratigraphy:
            Title: Proposed GCS Site
            FigureDPI: 500
            PlotInjectionSites: True
            PlotInjectionSiteLabels: True
            InjectionCoordx: 200
            InjectionCoordy: 200
            PlotWellbores: True
            PlotWellLabels: True
            PlotStratComponents: True
            SaveCSVFiles: False
            SpecifyXandYLims:
                xLims: [0, 400]
                yLims: [0, 400]
            SpecifyXandYGridLims:
                gridXLims: [25, 375]
```

(continues on next page)

(continued from previous page)

```
            gridYLims: [25, 375]
        StrikeAndDipSymbol:
            PlotSymbol: True
            coordx: 100
            coordy: 300
            length: 75
        View:
            ViewAngleElevation: [5, 10, 5, 10]
            ViewAngleAzimuth: [300, 300, 310, 310]
```

For examples of `Stratigraphy` plots, see examples *ControlFile_ex33a.yaml-ControlFile_ex38.yaml*. For examples of using `Stratigraphy` plots in a script application, see files *iam_sys_reservoir_mswell_stratplot_dipping_strata.py* and *iam_sys_reservoir_mswell_stratplot_no_dip.py*.

2.7.4 AoR

Area of Review (`AoR`) plots are developed to estimate the AoR needed for a geologic carbon storage project based on the spatial extent of reservoir impacts (pressure and CO_2 saturation) and potential aquifer impacts (dissolved salt and dissolved CO_2 plume volumes). The potential extent is found by distributing `OpenWellbore` components across the domain. We recommend setting `OpenWellbore` locations using the grid placement option (see examples *ControlFile_ex31a.yaml* to *ControlFile_ex31d.yaml*). The `OpenWellbore` (components) are hypothetical and used to consider the aquifer impacts that could occur if a leakage pathway (extending from the reservoir to the aquifer being considered) was available at each `OpenWellbore` location. The approach used for `AoR` plots is based on the work [1].

Note that the `AoR` plot type is meant to be used only for one aquifer at a time, with that aquifer being represented by only one type of aquifer component (e.g., representing contaminant spread in aquifer 2 with a `FutureGen2Aquifer` component). For example, file *ControlFile_ex31a.yaml* has `SimpleReservoir` components that provide the input for `OpenWellbore` components, and the `OpenWellbore` components provide input to `FutureGen2Aquifer` components. The `FutureGen2Aquifer` components are set up to represent aquifer 2. If the user added an entry to the *.yaml* file for a `FutureGen2AZMI` aquifer component representing aquifer 1, the `AoR` plot could not make plots representing the impacts on both aquifers 1 and 2. In this case, one would need to create a separate *.yaml* file that creates `AoR` plots just for aquifer 1.

`AoR` plots can be created for the following types of outputs: **pressure**, **CO2saturation**, **pH_volume**, **TDS_volume**, **Dissolved_CO2_volume**, and **Dissolved_salt_volume**. The `AoR` plot type examines these metrics at each location in the domain (i.e., each hypothetical `OpenWellbore` location) and displays the maximum value over time (across all times or at specific times, depending on the `TimeList` entry provided; this entry is discussed below). For LHS simulations, the `AoR` plot displays the maximum values over time at each location from all LHS realizations. This approach is meant to depict how severe the reservoir and aquifer impacts could become. Using the `AoR` plot type leads to the creation of *.csv* files containing the values shown in the `AoR` plots. Note that model run times can increase dramatically with the number of `OpenWellbore` locations. Additionally, some aquifer components generally require longer model run times (e.g., `GenericAquifer`) in comparison with other aquifer components (e.g., `FutureGen2Aquifer`). Also note that `FutureGen2Aquifer` is meant to be set up for aquifers with bottom depths <= 700 m, while `FutureGen2AZMI` is meant to be set up for aquifers with bottom depths >= 700 m.

When using the `AoR` plot type, we recommend setting `GenerateOutputFiles` and `GenerateCombOutputFile` to `False` in the `ModelParams` section of the *.yaml* file. The large number of `OpenWellbore` locations commonly used for `AoR` plots causes a large number of output files. A reservoir and aquifer component is created for each `OpenWellbore` location, and every component will have its output saved. The `.csv` files created for the `AoR` plots contain all of the necessary information and these files are much smaller in size.

`AoR` plots can have six optional entries: `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SaveCSVFiles`, `FigureDPI`, `FigureSize`, and `TimeList`. All of these entries are described above.

If the TimeList entry is not provided for an AoR plot, the figure will show the maximum values at each location across all model times. If TimeList is provided as a list of times in years (e.g., `TimeList: [1, 5, 10]` or `TimeList: [10]`), then the figures created will represent the maximum values at each location at the specified time(s). Otherwise, an AoR figure can be made for every model time by providing `TimeList: All`. Evaluating how the potential impacts of a project change over time can inform, for example, how the required extents of surveying efforts change over time (i.e., discovering and effectively plugging legacy wells at larger distances from the injection site).

Below is an example of two AoR plot entries in a *.yaml* file. The first entry uses the default settings, while the second specifies all available options. Since the simulation uses a LookupTableReservoir this example includes InjectionCoordx and InjectionCoordy. These inputs are not required for other reservoir component types.

```
Plots:
    AoR_pH_Default_Settings:
        AoR: [pH_volume]
    AoR_TDS.tiff:
        AoR: [TDS_volume]
        PlotInjectionSites: True
        InjectionCoordx: 2.37e5
        InjectionCoordy: 4.41e6
        FigureDPI: 300
        SaveCSVFiles: False
        TimeList: All
```

For examples of AoR plots, see *ControlFile_ex31a.yaml* to *ControlFile_ex32c.yaml*.

2.7.5 TTFD

The time to first detection (TTFD) plot type uses contaminant plume output from aquifer components to simulate when a monitoring well would be able to detect the plume in the aquifer(s) considered. If the TTFD plot type is run without monitoring locations provided, it still produces maps showing the spread of contaminant plumes across the domain. These figures (and the .csv files that can be saved) could then be used to decide where to place monitoring sensors.

The TTFD plot type can produce three types of figures: maps of earliest plume timings across the domain (i.e., the earliest time at which the plume type occurs in each part of the aquifer(s) considered), maps showing the TTFD provided by the entered monitoring locations, and maps of the probability of plume occurrence in the aquifer(s) considered. The figures with the TTFD from monitoring locations are only created if monitoring locations are entered. The maps of plume probabilities are only created if the analysis type is Latin Hypercube Sampling (lhs) or Parameter Study (parstudy). Note that plume probabilities are calculated as the number of realizations in which a plume occurred at each location divided by the total number of realizations.

The TTFD plot type requires the use of at least one of the following aquifer component types (with the component(s) set up to represent the aquifer(s) considered): CarbonateAquifer, FutureGen2Aquifer, FutureGen2AZMI, GenericAquifer, DeepAlluviumAquifer, or DeepAlluviumAquiferML. Note that the FutureGen2Aquifer component is used for aquifers with bottom depths <= 700 m, while the FutureGen2AZMI component is used for aquifers with bottom depths >= 700 m. The aquifer component(s) must also produce the plume dimension metrics associated with the plume type considered (e.g., **TDS_dx**, **TDS_dy**, and **TDS_dz** for TDS plumes). Note that CarbonateAquifer components do not produce plume dimension outputs for different plume types, so the required outputs when using CarbonateAquifer are **dx** and **dy** (which represent the lengths of the impacted aquifer volume in the x- and y-directions, respectively).

The plume timing and plume probability figures made with the TTFD plot type show four subplots. Each subplot contains a quarter of the depth range from the top of the reservoir to the surface. Each subplot contains the results for sections of aquifers within the corresponding depth range. If monitoring sensor locations are provided, each subplot will also show any sensors with depth (z) values in the subplot's depth range as black triangles. Because there are multiple z grid points within each subplot, there can be different layers of results displayed. The code is set up to make the top layer shown be the layer with the lowest plume timing or highest plume probability (for the corresponding figure types). The matplotlib function used to display results by color (contourf) can fail to display results when there are very few points with results in a layer. To address such situations, if there are fewer than 25 points with results we display each value as a color-labelled circle.

While the plume timing plots show the earliest plume timings at each grid location across the domain, the monitoring `TTFD` plots only display plume timings that are sufficiently close to the sensor location(s) provided. The purpose of such graphs is to show when the sensors used could warn site operators that an aquifer has been impacted. If the chosen sensor `x`, `y`, and `z` values do not provide any warning of plumes in an aquifer, and there are plumes in that aquifer, then the monitoring locations should be changed. The distance over which sensors can detect a plume are controlled by the `VerticalWindow` and `HorizontalWindow` entries, which are discussed below. Note that the `TTFD` plot type can produce output for the DREAM tool (Design for Risk Evaluation And Management) if `WriteDreamOutput` is set to `True` (see below). DREAM is designed to optimize the placement of monitoring sensors.

Unlike most other plot types, the `TTFD` plot type has two required entries: `PlumeType` and `ComponentNameList`. `TTFD` plots will not be produced without appropriate input for these entries. `ComponentNameList` is discussed above.

- `PlumeType` - the type of plume metric being considered. Acceptable values are *Pressure*, *pH*, *TDS*, *Dissolved_CO2*, *Dissolved_salt*, and *CarbonateAquifer*. The dx, dy, and dz metrics (e.g., **Dissolved_CO2_dz**) for the PlumeType used must be produced by the aquifer components listed in `ComponentNameList`. The dz metrics are not required when using `CarbonateAquifer` components, however, as these components do not produce a dz plume metric. Additionally, when using `PlumeType: CarbonateAquifer` the plume timing and plume probability figures do not have different subplots for different depth ranges.

The `TTFD` plot type can have the following optional entries: `MonitoringLocations`, `SaveCSVFiles`, `WriteDreamOutput`, `SpecifyXandYLims`, `NumZPointsWithinAquifers`, `NumZPointsWithinShales`, `xGridSpacing`, `yGridSpacing`, `SpecifyXandYGridLims`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `FigureDPI`, and `FigureSize`. Three of these entries (`MonitoringLocations`, `SpecifyXandYLims`, and `SpecifyXandYGridLims`) are dictionaries containing additional entries (i.e., entries indented beneath mentioned keywords in a *.yaml* file). All of these entries except for `MonitoringLocations`, `WriteDreamOutput`, `NumZPointsWithinAquifers`, and `NumZPointsWithinShales` are described above.

The `NumZPointsWithinAquifers`, `NumZPointsWithinShales`, `xGridSpacing`, `yGridSpacing`, and `SpecifyXandYGridLims` entries all relate to the x-, y-, and z-coordinates of the grids used to evaluate plume extents and timings. The dx, dy, and dz plume dimension metrics (e.g., *pH_dy* or *TDS_dz*) are used to evaluate whether each (x, y, z) of a grid is within a plume area for each model timestep. Note that `NumZPointsWithinAquifers` and `NumZPointsWithinShales` do not have an effect when setup `PlumeType: CarbonateAquifer` is used because that `CarbonateAquifer` component does not produce a dz plume metric.

- `MonitoringLocations` - a dictionary containing five optional entries related to the sensors used to detect aquifer impacts. The five optional entries are `coordx`, `coordy`, `coordz`, `HorizontalWindow`, and `VerticalWindow`. Note that the lists provided for `coordx`, `coordy`, and `coordz` must all have the same length (although `coordz` is not used with option `PlumeType: CarbonateAquifer`).
- `coordx` - an entry under `MonitoringLocations` that specifies the x-coordinate(s) (m) of monitoring sensor(s), if any sensors are used. This entry must be provided as a list, even if only one location is used (e.g., [100] or [100, 200]).
- `coordy` - an entry under `MonitoringLocations` that specifies the y-coordinate(s) (m) of monitoring sensor(s), if any sensors are used. This entry must be provided as a list, even if only one location is used (e.g., [100] or [100, 200]).
- `coordz` - an entry under `MonitoringLocations` that specifies the depth(s) (z-coordinate(s), (m)) of monitoring sensor(s), if any sensors are used. Note that for this entry, depths beneath the surface are taken as negative values. This entry must be provided as a list, even if only one location is used (e.g., [-500] or [-500, -400]). The `coordz` entry is not required when using an option `plumeType: CarbonateAquifer`, as the `CarbonateAquifer` component does not produce a dz plume metric.

- `HorizontalWindow` - a (maximum) horizontal distance (m) from which monitoring sensor(s) will detect plumes (default is 1). For example, if the HorizontalWindow is 5 m, then the sensor will detect any plume at grid locations within 5 m of the sensor's `coordx` and `coordy` values (if the plume is also within `VerticalWindow` of the sensor's `coordz` value). This entry is meant to represent the sensitivity of a sensor, but that consideration must also involve the threshold used for the plume type considered (if the aquifer component has a user-defined threshold for plume detection). For example, **Dissolved_salt** plumes from the `GenericAquifer` are influenced by the **dissolved_salt_threshold** parameter. In contrast, the `FutureGen2Aquifer` component defines TDS plumes where the relative change in TDS is > 10% (i.e., no user-defined threshold). The inclusion of plumes at nearby grid points is also dependent on the spacing of grid points; the x- and y-spacings are controlled by `xGridSpacing` and `yGridSpacing`, while the z-spacing is controlled by `NumZPointsWithinAquifers` and `NumZPointsWithinShales`. Note that the grid is made to include the x-, y-, and z-coordinates for monitoring locations, so there will always be a grid point for each monitoring sensor.
- `VerticalWindow` - a (maximum) vertical distance (m) from which monitoring sensor(s) will detect plumes (default is 1). For example, if the `VerticalWindow` is 5 m, then the sensor will detect any plume within 5 m of the sensor's `coordz` values (if the plume is also within `HorizontalWindow` of the sensor's `coordx` and `coordy` value). This entry is meant to represent the sensitivity of a sensor, but that consideration must also involve the threshold used for the plume type considered (if the aquifer component has a user-defined threshold for plume detection). For example, **Dissolved_CO2** plumes from the `GenericAquifer` are influenced by the **dissolved_co2_threshold** parameter. In contrast, the `FutureGen2Aquifer` component defines pH plumes where the absolute change in pH is > 0.2 (i.e., no user-defined threshold). The inclusion of plumes at nearby grid points is dependent on the spacing of grid points; the x- and y-spacings are controlled by `xGridSpacing` and `yGridSpacing`, while the z-spacing is controlled by `NumZPointsWithinAquifers` and `NumZPointsWithinShales`. Note that the grid is made to include the x-, y-, and z-coordinates for monitoring locations, so there will always be a grid point for each monitoring sensor.
- `WriteDreamOutput` - the option to create *.iam* files containing plume timing results (default is `False`). These *.iam* files are the input for the DREAM program. DREAM is the Design for Risk Evaluation And Management tool, which was also developed by NRAP. The only acceptable values are `True` or `False`.
- `NumZPointsWithinAquifers` - the number of z-grid points extending from the bottom to the top of each aquifer (default is 10). The points are equally spaced.
- `NumZPointsWithinShales` - the number of z-grid points extending from the bottom to the top of each shale (default is 3). The points are equally spaced. Note that the top of an aquifer is also the bottom of a shale, and the same location is not entered twice. In other words, with the default values for `NumZPointsWithinAquifers` (10) and `NumZPointsWithinShales` (3) a z-grid will have ten points from the bottom to the top of an aquifer, then a point in the middle of the overlying shale (point 2 of 3 across the shale), and then ten points from the bottom to the top of the overlying aquifer (etc.). In this example, including points 1 and 3 for the shale would be redundant because those points are included for the aquifers below and above the shale.

Below, we show two examples of `TTFD` plots setup in the `Plots``section of a *.yaml* file. The first plot (*pH_Minimum_Input*) has only the entries required to set up the ``TTFD` plot type: `PlumeType` and `ComponentNameList`. The second plot (*TDS_All_Options_Specified.tiff*) includes all optional entries for the TTFD plot type. Although there are only two plot entries included, each entry can result in the creation of multiple figures (e.g., earliest plume timings, TTFD from monitoring locations, and plume probabilities for each model realization). Note that all entries for the `TTFD` plot type are indented under `TTFD` which is indented under the figure name.

```
Plots:
    pH_Minimum_Input:
        TTFD:
            PlumeType: pH
            ComponentNameList: [FutureGen2AZMI1, FutureGen2Aquifer1]
    TDS_All_Options_Specified.tiff:
        TTFD:
```

(continues on next page)

(continued from previous page)

```
            PlumeType: TDS
            ComponentNameList: [FutureGen2AZMI1, FutureGen2Aquifer1]
            FigureDPI: 300
            MonitoringLocations:
                coordx: [100, 200]
                coordy: [100, 200]
                coordz: [-407.5, -407.5]
                HorizontalWindow: 1
                VerticalWindow: 5
            PlotInjectionSites: True
            InjectionCoordx: 50
            InjectionCoordy: 50
            SpecifyXandYLims:
                xLims: [-25, 700]
                yLims: [-25, 700]
            NumZPointsWithinAquifers: 10
            NumZPointsWithinShales: 3
            xGridSpacing: 5
            yGridSpacing: 5
            SpecifyXandYGridLims:
                gridXLims: [25, 650]
                gridYLims: [25, 650]
            WriteDreamOutput: False
            SaveCSVFiles: True
```

For examples of TTFD plots, see *ControlFile_ex39.yaml* to *ControlFile_ex43.yaml*.

2.7.6 GriddedMetric

The `GriddedMetric` plot type produces map view images of a gridded metric. While the radial metrics shown by the `GriddedRadialMetric` plot type are defined in relation to radius and depth values, the metrics shown by the `GriddedMetric` plot type are defined relative to x-coordinates and y-coordinates. For example, the `GriddedMetric` plot type can display the gridded output produced by `SealHorizon` and `FaultFlow` components.

The `GriddedMetric` plot type has two required entries: `ComponentNameList` and and `MetricName`. Both are described above.

The `GriddedMetric` plot type has the following optional entries: `Realization`, `TimeList`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SpecifyXandYLims`, `SaveCSVFiles`, `EqualAxes`, `FigureDPI`, and `FigureSize`. All of these entries are discussed above.

Below, we show two examples of setting up `GriddedMetric` plots in a *.yaml* control file. The first plot (*Plot_Default_Settings*) includes only the required entries, while the second (*Plot_With_Options*) includes all optional entries.

```
Plots:
    Plot_Default_Settings:
        GriddedMetric:
            ComponentNameList: [Fault1]
            MetricName: mass_brine_aquifer
    Plot_With_Options:
        GriddedMetric:
            ComponentNameList: [Fault1]
```

(continues on next page)

(continued from previous page)

```
            MetricName: CO2_aquifer
            Realization: 0
            FigureDPI: 300
            TimeList: [1, 5, 10, 25, 50]
            SaveCSVFiles: False
            PlotInjectionSites: False
            InjectionCoordx: 4.68e+04
            InjectionCoordy: 5.11e+04
            SpecifyXandYLims:
                xLims: [38750, 40500]
                yLims: [48266, 48400]
            EqualAxes: False
```

For examples of `GriddedMetric` plots, see *ControlFile_ex18.yaml*, *ControlFile_ex19.yaml*, and *ControlFile_ex23.yaml*.

2.7.7 GriddedRadialMetric

The `GriddedRadialMetric` plot type produces map view images of a gridded radial metric. The `GenericAquifer` produces four kinds of gridded radial metrics: **r_coordinate**, **z_coordinate**, **Dissolved_CO2_mass_fraction**, and **Dissolved_salt_mass_fraction**. Regions of an aquifer with dissolved CO_2 and dissolved salt mass fractions exceeding the corresponding mass fraction threshold are included in the plume volumes for the corresponding plume type. Those plume volumes can be visualized with the `TTFD` plot type. The `GriddedRadialMetric` plot type, however, can show more general changes in dissolved CO_2 and salt mass fractions (e.g., seeing changes in mass fractions below the plume definition thresholds).

The `GriddedRadialMetric` plot type has three required entries: `ComponentNameList`, `ZList`, and `MetricName`. `ComponentNameList` and `MetricName` are discussed above.

- `ZList` - the depths (m) at which to evaluate the radial metric output. The depth in the radial grid (e.g., **z_coordinate** from `GenericAquifer`) that is closest to each value entered will be used. String inputs representing the bottom of a unit can also be provided. For example, the bottom and top depths of aquifer 2 can be set up by entering `ZList: [aquifer2Depth, shale3Depth]`. Shale 3 is on top of aquifer 2, so the bottom depth of shale 3 is the top depth of aquifer 2. Note that numeric values given for `ZList` (not string inputs like shale2Depth) are taken as being negative when they represent a depth beneath the surface (e.g., `ZList: [-500, -400]` for depths of 500 m and 400 m).

The `GriddedRadialMetric` plot type has 11 optional entries: `MinValue`, `DegreeInterval`, `Realization`, `TimeList`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SpecifyXandYLims`, `SaveCSVFiles`, `EqualAxes`, `FigureDPI`, and `FigureSize`. All of these entries except for `MinValue` and `DegreeInterval` are discussed above.

- `MinValue` - the minimum value used for the colorbar on the figures. Any values beneath this minimum will not be displayed graphically, but the entire range of values is still displayed in the title of each figure. This parameter has a significant impact on `GriddedRadialMetric` figures. For example, the **Dissolved_CO2_mass_fraction** and **Dissolved_salt_mass_fraction** outputs saved by a `GenericAquifer` for a time of 0 years will all have values of zero. The outputs saved at other times, however, can have very low but nonzero values. The **Dissolved_CO2_mass_fraction** values can be as low as 5.0e-3 at the highest radii, while the **Dissolved_salt_mass_fraction** values can be as low as 1.0e-30. If the `MinValue` provided is zero, then the figures created will be zoomed out to encompass such low values at the highest radii evaluated (about 77.5 km). These large extents will make the figures visually unclear. For these figures to be useful, one should specify a `MinValue` that is high enough to enable the figure to focus on the area of interest (i.e., near the component's location) but low enough to not exclude too much of the output data. We recommend using a `MinValue` of 0.002 when evaluating **Dissolved_salt_mass_fraction** (10 times lower than the default **dissolved_salt_threshold** of

0.02) and 0.01 when evaluating **Dissolved_CO2_mass_fraction** (equal to the default **dissolved_salt_threshold** of 0.01). If `MinValue` is not entered, these values will be used as the defaults for the corresponding output type (dissolved salt or dissolved CO_2). If all of the times evaluated only have values less or equal to `MinValue`, then one figure will be made. This figure has a title that includes 'All Model Times.' Note that the .csv files saved when `SaveCSVFiles` is set to `True` will only include values above `MinValue`.

- `DegreeInterval` - the interval (degrees) used to create a map-view image from the radial output. The accepted values are 1, 5, 10, 15, 30, and 45. If `DegreeInterval` is not entered, the default values is 15 degrees.

Note that although the `Realization` entry for the `GriddedRadialMetric` plot type follows the indexing conventions of Python (i.e., `Realization: 0` for the first realization), the figure files and .csv files saved by the `GriddedRadialMetric` plot type will present the simulation number as ranging from one to the total number of realizations (e.g., Simulation 1 instead of Simulation 0).

Below, we show two examples of `GriddedRadialMetric` plot entries in a control file. The first entry (*Min_Input_Dissolved_Salt*) uses the minimum input required for the `GriddedRadialMetric` plot type. The second entry (*All_Input_Dissolved_Salt*) uses all entries available for the `GriddedRadialMetric` plot type.

```
Plots:
    Min_Input_Dissolved_Salt:
        GriddedRadialMetric:
            ComponentNameList: [GenericAquifer1]
            MetricName: Dissolved_salt_mass_fraction
            ZList: [aquifer2Depth]
    All_Input_Dissolved_Salt:
        GriddedRadialMetric:
            ComponentNameList: [GenericAquifer1]
            MetricName: Dissolved_salt_mass_fraction
            ZList: [aquifer2Depth, shale3Depth]
            TimeList: [1, 5, 10, 15, 20]
            MinValue: 0.002
            FigureDPI: 300
            PlotInjectionSites: True
            InjectionCoordx: 100
            InjectionCoordy: 100
            DegreeInterval: 1
            Realization: 0
            EqualAxes: True
            SaveCSVFiles: True
            SpecifyXandYLims:
                xLims: [-200, 400]
                yLims: [-200, 400]
```

For examples of `GriddedRadialMetric` plots, see *ControlFile_ex53a.yaml* to *ControlFile_ex53d.yaml*

2.7.8 AtmPlumeSingle

The `AtmPlumeSingle` plot type produces map view images depicting how CO_2 leakage at the surface creates atmospheric CO_2 plumes. These images are created for each time step during one realization of a simulation. Note that simulations using the Latin Hypercube Sampling (`lhs`) or Parameter Study (`parstudy`) analysis types have many realizations, while a simulation using a `forward` analysis type only has one realization. For `AtmPlumeSingle` plot type with `lhs` or `parstudy` simulations, the visualization corresponding to the realization of interest can be setup with the `Realization` entry in the *.yaml* file (discussed above). Note that using the `AmtPlumeSingle` plot type requires the use of an AtmosphericROM component.

Here is an example of the ModelParams section from *ControlFile_ex40.yaml*, where the number of LHS realizations is set as `siz: 30`.

```
ModelParams:
    EndTime: 15.
    TimeStep: 1
    Analysis:
        type: lhs
        siz: 30
    Components: [LookupTableReservoir1, MultisegmentedWellbore1,
                 FutureGen2AZMI1, FutureGen2AZMI2]
    OutputDirectory: output/output_ex40_{datetime}
    Logging: Info
```

The produced figures show the source of the CO_2 leak as a red circle and the plume as a blue circle. The source location(s) are set by the x and y coordinate(s) of the component that the `AtmosphericROM` is connected to. For example, in *ControlFile_ex9a.yaml*, the `AtmosphericROM` component is connected to an `OpenWellbore` component and the `OpenWellbore` component has its locations entered with `coordx` and `coordy`. The `coordx` and `coordy` values serve as the coordinates of sources for the `AtmosphericROM` component. In the `AtmPlumeSingle` figures, the `coordx` and `coordy` values are shown as the CO_2 sources. In the final figures the plumes are labeled as *Critical Areas* because the area is defined as being within the **critical_distance** output (from an `AtmosphericROM`) from the corresponding source. The critical areas are, therefore, the areas in which the CO_2 concentrations exceed the value defined by the parameter **CO_critical**. The **critical_distance** is the radius of each plume circle shown in `AtmPlumeSingle` plots, and this **critical_distance** is also displayed on the figure with text.

Note that when multiple atmospheric plumes overlap enough, they will be displayed as one plume. The source shown will be between the sources of each individual plume.

`AtmosphericROM` components can be provided with receptor locations, which are meant to represent home or business locations where people will be present. If receptors are provided and the *.yaml* input for the `AtmPlumeSingle` includes the entry `PlotReceptors: True`, then receptor locations will be shown.

The `AtmPlumeSingle` plot type can have the following optional entries `Realization`, `PlotReceptors`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SpecifyXandYLims`, `FigureDPI`, and `FigureSize`. All of these entries except for `PlotReceptors` are described above.

- `PlotReceptors` - option to plot receptor locations (default is `False`). The only acceptable values are `True` or `False`. If the receptors are far away from the source location(s) and/or the injection site, plotting the receptors may cause the x and y limits to be spread too far. The plumes may then be difficult to see.

Below is an example of the `AtmPlumeSingle` plot input in a *.yaml* control file. Note that `InjectionCoordx` and `InjectionCoordy` only have to be provided when using a `LookupTableReservoir` and setting `PlotInjectionSites: True`.

```
Plots:
    ATM_single:
        AtmPlumeSingle:
            Realization: 10
            FigureDPI: 300
            PlotInjectionSites: True
            InjectionCoordx: 3.68e4
            InjectionCoordy: 4.83e4
            PlotReceptors: True
            SpecifyXandYLims:
                xLims: [3.58e4, 3.78e4]
                yLims: [4.73e4, 4.93e4]
```

For examples of `AmtPlumeSingle` plots, see *ControlFile_ex9a.yaml* to *ControlFile_ex9c.yaml*.

2.7.9 AtmPlumeEnsemble

The `AtmPlumeEnsemble` plot type can only be used in simulations with Latin Hypercube Sampling (`lhs`) or Parameter Study (`parstudy`) analysis types. This plot type involves concepts similar to those as those of the `AtmPlumeSingle` plot type. While the `AtmPlumeSingle` plot type dislays the critical areas for one realization, the `AtmPlumeEnsemble` plot type displays the probability of critical areas occuring in the domain. These probabilities are calculated with the results from all realizations of the `lhs` or `parstudy` simulation. The probabilities specifically represent the likelihood of CO_2 plume concentrations exceeding the threshold set with the **CO_critical** parameter for `AtmosphericROM` components. The probabilities are shown as gridded data. The `AtmPlumeEnsemble` plot type is available only when example setup includes an `AtmosphericROM` component.

The `AtmPlumeEnsemble` plot type has the optional entries `PlotReceptors`, `PlotInjectionSites`, `InjectionCoordx`, `InjectionCoordy`, `SpecifyXandYGridLims`, `xGridSpacing`, `yGridSpacing`, `SpecifyXandYLims`, `FigureDPI`, and `FigureSize`. All of these entries are described above.

Below is an example of a `AtmPlumeEnsemble` plot entry in a *.yaml* file:

```
Plots:
    ATM_Ensemble.tiff:
        AtmPlumeEnsemble:
            FigureDPI: 300
            PlotInjectionSites: True
            InjectionCoordx: 200
            InjectionCoordy: 200
            PlotReceptors: False
            xGridSpacing: 1
            yGridSpacing: 1
            SpecifyXandYGridLims:
                gridXLims: [-100, 300]
                gridYLims: [-100, 300]
            SpecifyXandYLims:
                xLims: [-125, 325]
                yLims: [-125, 325]
```

For examples of `AmtPlumeEnsemble` plots, see *ControlFile_ex9a.yaml* and *ControlFile_ex9c.yaml*.

2.8 Units

Data passed between models need to have consistent units. Here is a list of units used by the NRAP-Open-IAM.

- Pressure is assumed to be in units of Pascals (Pa).
- Time is assumed to be in days.
- Distance, width, length, height, depth are assumed to be in units of meters (m).
- Flow rates are assumed to be in units of kilograms per second (kg/s).
- Mass is assume to be in units of kilograms (kg).
- Viscosities are assumed to be in units of Pascal seconds ($Pa{\cdot}s$).
- Permeability is assumed to be in units of meters squared (m^2).

CHAPTER

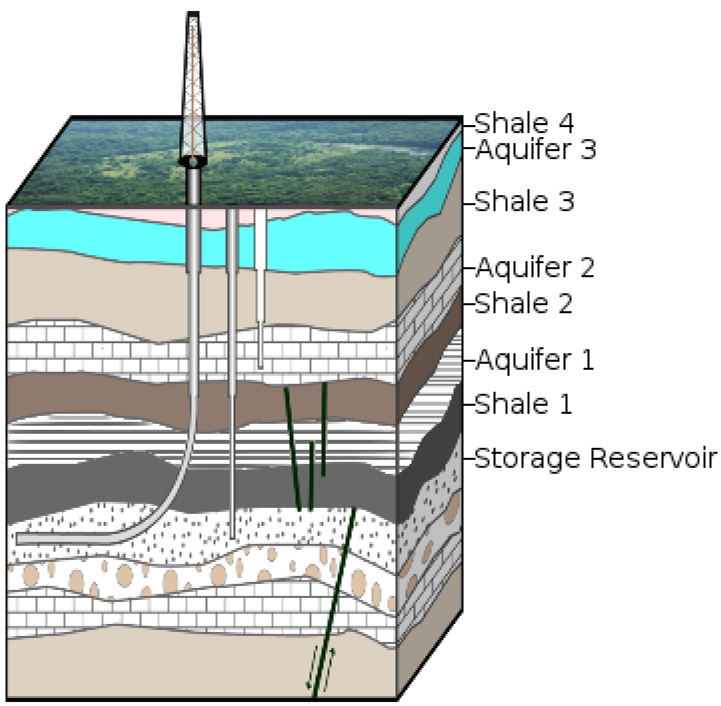
THREE

COMPONENTS DESCRIPTION

This section of the document will describe each of the component models available for use in the NRAP-Open-IAM. For each component model all parameters that can be specified are described along with units and acceptable ranges. Description of the outputs that can be returned by each component is also provided.

3.1 Stratigraphy Component

The Stratigraphy component is a component containing parameters describing the structure of the storage reservoir system. The stratigraphy component allows the number of shale (or aquitard) layers to be set, thus, setting the total number of layers in the system. Each shale or aquifer layer can take on the default thickness for that layer type or be assigned a user defined value with `shale#Thickness` or `aquifer#Thickness` keywords where # is replaced by an index of the layer (e.g., `shale1Thickness`). Layers are numbered from the bottom upward: shale 1 is the layer above the storage reservoir, and, with N shale layers total, shale N is the surface layer.



The description of the component's parameters is provided below.

- **numberOfShaleLayers** [-] (3 to 30) - number of shale layers in the system (default: 3). The shale units must be separated by an aquifer.

- **shaleThickness** $[m]$ (1 to 1600) - thickness of shale layers (default 250). Thickness of shale layer 1, for example, can be defined by **shale1Thickness**; otherwise, shale layers for which the thickness is not defined will be assigned a default thickness.
- **aquiferThickness** $[m]$ (1 to 1600) - thickness of aquifers (default: 100). Thickness of aquifer 1, for example, can be defined by **aquifer1Thickness**; otherwise, aquifers for which the thickness is not defined will be assigned a default thickness.
- **reservoirThickness** $[m]$ (1 to 1600) - thickness of reservoir (default: 50)
- **datumPressure** $[Pa]$ (80,000 to 300,000) - pressure at the top of the system (default: 101,325)
- **depth** $[m]$ (5 to 30,000) - depth to the top of reservoir (default: 950).

For control file examples the following composite parameters are produced:

- **shale#Depth** $[m]$ (boundaries depend on the user input) - depth to the base of the shale layer with index # (default value is not defined)
- **aquifer#Depth** $[m]$ (boundaries depend on the user input) - depth to the base of the aquifer layer with index # (default value is not defined)
- **reservoirDepth** $[m]$ (boundaries depend on the user input) - depth to the base of the reservoir (default value is not defined).

For script examples these parameters have to be added explicitly as composite parameters of the stratigraphy component.

3.2 Simple Reservoir Component

The Simple Reservoir component model is a semi-analytical model for the reservoir. It is focused on flow across relatively large distances and does not take into account discrete features of the flow paths such as fractures, cracks, etc. The model is based on work of Nordbotten et al., [6]. Further reading can be found in [23], [5], [24], [22].

In the NRAP-Open-IAM control file, the type name for the Simple Reservoir component is `SimpleReservoir`. The description of the component's parameters is provided below:

- **logResPerm** $[\log_{10} m^2]$ (-14 to -9) - logarithm of reservoir permeability (default: -12)
- **reservoirPorosity** [-] (0.01 to 1) - porosity of reservoir (default: 0.3)
- **brineDensity** $[kg/m^3]$ (900 to 1500) - density of brine phase (default: 1000)
- **CO2Density** $[kg/m^3]$ (100 to 1500) - density of CO_2 phase (default: 479)
- **brineViscosity** $[Pa{\cdot}s]$ (1.0e-4 to 5.0e-3) - viscosity of brine phase (default: 2.535e-3)
- **CO2Viscosity** $[Pa{\cdot}s]$ (1.0e-6 to 1.0e-4) - viscosity of CO_2 phase (default: 3.95e-5)
- **brineResSaturation** [-] (0 to 0.7) - residual saturation of brine phase (default: 0.1)
- **compressibility** $[Pa^{-1}]$ (5.0e-11 to 1.0e-9) - compressibility of brine and CO_2 phases (assumed to be the same for both phases) (default: 5.1e-11)
- **injRate** $[m^3/s]$ (1.0e-3 to 10) - CO_2 injection rate (default: 0.1)
- **numberOfShaleLayers** [-] (3 to 30) - number of shale layers in the system (default: 3); *linked to Stratigraphy*. The shale units must be separated by an aquifer.
- **shaleThickness** $[m]$ (1 to 1600) - thickness of shale layers (default 250); *linked to Stratigraphy*. Thickness of shale layer 1, for example, can be defined by **shale1Thickness**; otherwise, shale layers for which the thickness is not defined will be assigned a default thickness.

- **aquiferThickness** $[m]$ (1 to 1600) - thickness of aquifers (default: 100); *linked to Stratigraphy*. Thickness of aquifer 1, for example, can be defined by **aquifer1Thickness**; otherwise, aquifers for which the thickness is not defined will be assigned a default thickness.
- **reservoirThickness** $[m]$ (1 to 1600) - thickness of reservoir (default: 50); *linked to Stratigraphy*
- **datumPressure** $[Pa]$ (80,000 to 300,000) - pressure at the top of the system (default: 101,325); *linked to Stratigraphy*.

Possible observations from the Simple Reservoir component are:

- **pressure** $[Pa]$ - pressure at top of the reservoir at the user defined location(s)
- **CO2saturation** [-] - CO_2 saturation at the top of the reservoir at the user defined location(s)
- **mass_CO2_reservoir** $[kg]$ - mass of the CO_2 in the reservoir.

3.3 Analytical Reservoir Component

The Analytical Reservoir component model is a semi-analytical model for the reservoir. It is focused on flow across relatively large distances and does not take into account discrete features of the flow paths such as fractures, cracks, etc. The model is based on work of Nordbotten et al., [6]. Further reading can be found in [22], [3].

In the NRAP-Open-IAM control file, the type name for the analytical reservoir component is `AnalyticalReservoir`. The description of the component's parameters is provided below:

- **logResPerm** $[\log_{10} m^2]$ (-15.3 to -12) - logarithm of reservoir permeability (default: -13.69897)
- **reservoirPorosity** [-] (0.1 to 0.3) - porosity of reservoir (default: 0.15)
- **reservoirRadius** $[m]$ (500 to 100,000) - distance between injection well and outer reservoir boundary (default: 500)
- **brineDensity** $[kg/m^3]$ (965 to 1195) - density of brine phase (default: 1045)
- **CO2Density** $[kg/m^3]$ (450 to 976) - density of CO_2 phase (default: 479)
- **brineViscosity** $[Pa{\cdot}s]$ (2.3e-4 to 15.9e-4) - viscosity of brine phase (default: 2.535e-4)
- **CO2Viscosity** $[Pa{\cdot}s]$ (0.455e-6 to 1.043e-4) - viscosity of CO_2 phase (default: 3.95e-5)
- **brineResSaturation** [-] (0 to 0.25) - residual saturation of brine phase (default: 0)
- **brineCompressibility** $[Pa^{-1}]$ (3.63e-12 to 2.31e-11) - brine compressibility (default: 4.5e-12 = 3.1e-8 1/psi)
- **injRate** $[m^3/s]$ (0.0024 to 3.776) - CO_2 injection rate (default: 0.01)
- **numberOfShaleLayers** [-] (3 to 30) - number of shale layers in the system (default: 3); *linked to Stratigraphy*. The shale units must be separated by an aquifer.
- **shaleThickness** $[m]$ (1 to 1600) - thickness of shale layers (default: 250); *linked to Stratigraphy*. Thickness of shale layer 1, for example, can be defined by **shale1Thickness**; otherwise, shale layers for which the thickness is not defined will be assigned a default thickness.
- **aquiferThickness** $[m]$ (1 to 1600) - thickness of aquifers (default: 100); *linked to Stratigraphy*. Thickness of aquifer 1, for example, can be defined by **aquifer1Thickness**; otherwise, aquifers for which the thickness is not defined will be assigned a default thickness.
- **reservoirThickness** $[m]$ (15 to 500) - thickness of reservoir (default: 50); *linked to Stratigraphy*
- **datumPressure** $[Pa]$ (80,000 to 300,000) - pressure at the top of the system (default: 101,325); *linked to Stratigraphy*.

Possible observations from the Analytical Reservoir component are:

- **pressure** $[Pa]$ - pressure at top of the reservoir at the user defined location(s)
- **pressureAve** $[Pa]$ - pressure vertically averaged at the user defined location(s)
- **CO2saturation** [-] - CO_2 saturation vertically averaged at the user defined location(s)
- **mass_CO2_reservoir** $[kg]$ - (injected total) mass of the CO_2 in the reservoir.

3.4 Lookup Table Reservoir Component

The Lookup Table Reservoir component model is a reduced order model based on interpolation of data from a set of lookup tables. The lookup tables are based on the full-physics scale simulations. Each lookup table is determined by a particular set of M input model parameters which define a signature of the given set of lookup tables.

In the NRAP-Open-IAM control file, the type name for the Lookup Table Reservoir component is `LookupTableReservoir`. The component's parameters depend on the M input model parameters used to create lookup tables data. The minimum and maximum values of lookup table parameters determine boundaries of component parameters. Moreover, the component parameters values as a set can only be one of the combination of values that went into one of the lookup table linked to the component.

In the NRAP-Open-IAM control file a `FileDirectory` keyword must be specified. It indicates the directory where files with the simulation data for the lookup tables are located. A `TimeFile` keyword is a name of the .csv file that stores the time points (in years) at which the results in the tables are provided. If `TimeFile` is not specified then, by default, the name of the file with time data is assumed to be *time_points.csv*. The time file must be located in the directory specified by `FileDirectory`.

A `ParametersFilename` keyword can also be specified. It defines the names and values of lookup table parameters that were used to create the given set of lookup tables. Additionally, it lists the names of the *.csv* files containing simulation data for each of the lookup tables in the set (e.g., results from different parameterizations of a reservoir simulation). By default, the file with parameters data is assumed to be named *parameters_and_filenames.csv*. The parameters file should be in a comma separated values format. The first M entries in the first row of the file are the names of the lookup table parameters which were varied for different realizations; the (M+1)st entry is a word *filename*. Each subsequent row of the parameters file contains the M values of the lookup table parameters followed by the name of file (lookup table) with the corresponding realization data. The provided filename must match with one of the files in the `FileDirectory`.

The user should make sure that the information provided in `ParametersFilename` file on parameters and simulation data files is accurate and complete. In general, a given parameter of the Lookup Table Reservoir component can have any possible name. At the same time the (possibly random) names specified by user in the `ParametersFilename` file should be the same names that the user would use in the control file for the description of the Lookup Table Reservoir component parameters. Due to the way the lookup tables are produced, each parameter of the reservoir component can only take on certain deterministic values. The possible values of a given parameter should be listed after the `values:` keyword followed by the list in square brackets ('['). The weights for each parameter can be specified with the `weights:` keyword followed by the list of weights for each value in square brackets. The weights should sum to 1, otherwise, they will be normalized. If no weights are provided all values are assumed to be equally likely to occur.

There exists an option to sample the data from the lookup tables without direct reference to any of the parameters used for creating the tables. User can use an auxiliary parameter `index` added to the Lookup Table Reservoir component to sample data from a particular lookup table file based on its index in the file *parameters_and_filenames.csv*. This option allows to use lookup tables in the scenarios where the total number of lookup table data files is less than the number of all possible combinations of the lookup table parameters.

Simulation data files (listed in `ParametersFilename` file) in a comma separated values format contain the reservoir simulation data, e.g., pressure and CO_2 saturation, varying over time. The data is used to build the Lookup Table Reservoir component output. Each realization file begins with a header line that is ignored by the code. Each subsequent row of the file represents a particular spatial location. The first and the second columns are the x- and y-coordinates of the location, respectively. The subsequent columns contain reservoir simulation data at the location defined in the first two columns. The names of the columns should represent the data in them and have the form *base.obs.nm_#* where *base.obs.nm* is the name of observation as used in the system model and # is an index of the time point at which the given observation is provided. The indexing of the reservoir simulation data should always start with 1 not with 0. For example, the pressure data at the first time point (even if this time point is 0) should always be indexed as *pressure_1*. Further, if the column contains pressure data at the second time point, its name should be *pressure_2*, and so on. If the column contains saturation data at the 12th time point, its name should be *CO2saturation_12*. The order of the columns in the lookup table except the first two x and y columns is arbitrary. If some reservoir simulation data does not vary in time then the column name should indicate it: in any case its name should not contain underscore symbol _ followed by number (time index). For example, column with name #temperature# would indicate that the provided temperature data is constant in time.

The Lookup Table Reservoir component produces the output using interpolation in space and time within the spatio-temporal domain defined by the lookup tables simulation model setup. Observations from the Lookup Table Reservoir component are:

- **pressure** $[Pa]$ - pressure at top of the reservoir at the wellbore location(s)
- **CO2saturation** [-] - CO_2 saturation at the top of the reservoir at the wellbore location(s).

Observations *pressure* and *CO2saturation* are mandatory for the Lookup Table Reservoir component which means that the linked lookup tables should contain the necessary data to produce them. In addition, the component can return any other type of observations provided in the lookup tables.

3.5 Generic Reservoir Component

The Generic Reservoir component model is a reduced-order-model to predict pressure and CO_2 saturation of the top part of the reservoir during CO_2 injection (25 years) and post-injection period (50 years).

The model is a machine learning regression model fitted to the results of STOMP-CO2E multiphase flow transport simulations using Random Forest and scikit-learn library. Total ~2,614,400 data from ~4,000 numerical simulations were used to develop the model. The model predicts pressure and saturation only for the top part of the storage reservoir. Homogeneous reservoir model with radius 150 km was used. Input parameters were sampled using Latin Hypercube Sampling across wide ranges.

In the NRAP-Open-IAM control file, the type name for the generic reservoir component is `GenericReservoir`. The description of the component's parameters is provided below:

- **reservoirDepth** $[m]$ (1000 to 3500) - depth to the base of reservoir (default: 2000); *linked to Stratigraphy*
- **logResPerm** $[\log_{10} m^2]$ (-15 to -12) - logarithm of reservoir permeability (default: -14.0)
- **reservoirThickness** $[m]$ (15 to 200) - reservoir thickness (default: 50); *linked to Stratigraphy*
- **resTempGradient** $[°C/km]$ (18 to 32) - reservoir temperature gradient (default: 30)
- **injRate** $[kg/s]$ (29 to 179) - CO_2 injection rate (default: 100)
- **initialSalinity** [-] (0.001 to 0.05) - reservoir initial salinity (default: 0.05)
- **reservoirPorosity** [-] (0.08 to 0.40) - reservoir porosity (default: 0.1).

Possible observations from the Generic Reservoir component are:

- **pressure** $[Pa]$ - pressure at top of the reservoir at the user defined location(s)
- **CO2saturation** [-] - CO_2 saturation at top of the reservoir at the user defined location(s).

3.6 Multisegmented Wellbore Component

The Multisegmented Wellbore component estimates the leakage rates of brine and CO_2 along wells in the presence of overlying aquifers or thief zones. The model is based on work of Nordbotten et al., [24]. Further reading can be found in [22].

The model is focused on flow across relatively large distances and does not take into account discrete features of the flow paths such as fractures, cracks, etc. It assumes that leakage is occurring in the annulus between the outside of the casing and borehole. This area is assigned an "effective" permeability of the flow path. The permeability is applied over a length along the well that corresponds to the thickness of a shale formation. Each well is characterized by an effective permeability assigned to each segment of the well that crosses an individual formation. For example, if a well crosses N permeable formations, then it is characterized by N different permeability values. The model utilizes the one-dimensional multiphase version of Darcy's law to represent flow along a leaky well.

In the NRAP-Open-IAM control file, the type name for the Multisegmented Wellbore component is `MultisegmentedWellbore`. The description of the component's parameters are provided below. Names of the component parameters coincide with those used by `model` method of the `MultisegmentedWellbore` class.

- **logWellPerm** [$\log_{10} m^2$] (-101 to -9) - logarithm of well permeability along shale layer (default: -13). Logarithm of well permeability along shale 3, for example, can be defined by **logWell3Perm**. Permeability of the well along the shale layers not defined by user will be assigned a default value.
- **logAquPerm** [$\log_{10} m^2$] (-17 to -9) - logarithm of aquifer permeability (default: -12). Logarithm of aquifer 1 permeability, for example, can be defined by **logAqu1Perm**. Aquifer permeability not defined by user will be assigned a default value.
- **brineDensity** [kg/m^3] (900 to 1500) - density of brine phase (default: 1000)
- **CO2Density** [kg/m^3] (100 to 1000) - density of CO_2 phase (default: 479)
- **brineViscosity** [$Pa{\cdot}s$] (1.0e-4 to 5.0e-3) - viscosity of brine phase (default: 2.535e-3)
- **CO2Viscosity** [$Pa{\cdot}s$] (1.0e-6 to 1.0e-4) - viscosity of CO_2 phase (default: 3.95e-5)
- **aquBrineResSaturation** [-] (0 to 0.99) - residual saturation of brine phase in each aquifer (default: 0.0). For example, the residual brine saturation of aquifer2 can be defined by **aqu2BrineResSaturation**; otherwise, aquifer layers for which the residual brine saturation is not defined will be assigned a default value.
- **compressibility** [Pa^{-1}] (1.0e-13 to 1.0e-8) - compressibility of brine and CO_2 phases (assumed to be the same for both phases) (default: 5.1e-11)
- **wellRadius** [m] (0.01 to 0.5) - radius of leaking well (default: 0.05)
- **numberOfShaleLayers** [-] (3 to 30) - number of shale layers in the system (default: 3); *linked to Stratigraphy*. The shale units must be separated by an aquifer.
- **shaleThickness** [m] (1 to 3000) - thickness of shale layers (default: 250); *linked to Stratigraphy*. Thickness of shale layer 1, for example, can be defined by **shale1Thickness**; otherwise, shale layers for which the thickness is not defined will be assigned a default thickness.
- **aquiferThickness** [m] (1 to 1600) - thickness of aquifers (default: 100); *linked to Stratigraphy*. Thickness of aquifer 1, for example, can be defined by **aquifer1Thickness**; otherwise, aquifers for which the thickness is not defined will be assigned a default thickness.
- **reservoirThickness** [m] (1 to 1600) - thickness of reservoir (default: 30); *linked to Stratigraphy*.
- **datumPressure** [Pa] (80,000 to 300,000) - pressure at the top of the system (default: 101,325); *linked to Stratigraphy*

The possible outputs from the Multisegmented Wellbore component are leakage rates of CO_2 and brine to each of the aquifers in the system and atmosphere. The names of the observations are of the form:

- **CO2_aquifer1**, **CO2_aquifer2**,. . . , **CO2_atm** $[kg/s]$ - CO_2 leakage rates
- **brine_aquifer1**, **brine_aquifer2**,. . . , **brine_atm** $[kg/s]$ - brine leakage rates
- **mass_CO2_aquifer1**, **mass_CO2_aquifer2**,. . . , **mass_CO2_aquiferN** $[kg]$ - mass of the CO_2 leaked into the aquifer.

3.7 Cemented Wellbore Component

The Cemented Wellbore component model is based on a multiphase well leakage model implemented in the NRAP-IAM-CS, [10]. The model is built off detailed full-physics Finite Element Heat and Mass (FEHM) simulations, [39]. The FEHM simulations are three-dimensional (3-D), multiphase solutions of heat and mass transfer of water and supercritical, liquid, and gas CO_2. After the simulations are completed, the surrogate model is built based on the key input parameters and corresponding output parameters. The approximate (surrogate) model is represented by polynomials in terms of input parameters that then can be sampled to estimate leakage rate for wells. Early development work can be found in [15].

When using the control file interface with more than 3 shale layers, the `ThiefZone` keyword can be used to specify the thief zone aquifer and the `LeakTo` keyword can be specified to name the upper aquifer. These values will default to *aquifer1* and *aquifer2*, respectively, if are not provided by user. In the FEHM simulations used to create the surrogate model some of the stratigraphy layers were setup with a fixed thickness. In particular, shale above aquifer had thickness 11.2 m; aquifer and thief zone to which leakage was simulated were set to have thicknesses 19.2 m and 22.4 m, respectively; and reservoir had thickness of 51.2 m.

Component model input definitions:

- **logWellPerm** $[\log_{10} m^2]$ (-13.95 to -10.1) - logarithm of wellbore permeability (default: -13)
- **logThiefPerm** $[\log_{10} m^2]$ (-13.9991 to -12.00035) - logarithm of thief zone permeability (default: -12.2)
- **wellRadius** $[m]$ (0.025 to 0.25) - radius of the wellbore (default: 0.05)
- **initPressure** $[Pa]$ (1.0e+5 to 5.0e+7) - initial pressure at the base of the wellbore (default: 2.0e+7 Pa, or 20 MPa); *from linked component*
- **wellDepth** $[m]$ (960 to 3196.8) - depth in meters from ground surface to top of reservoir (default: 1500); *linked to Stratigraphy*
- **depthRatio** [-] (0.30044 to 0.69985) - fraction of well depth to the center of the thief zone from the top of the reservoir (default: 0.5); *linked to Stratigraphy*.

Temporal inputs of the Cemented Wellbore component are not provided directly to the component model method but rather are calculated from the current and several past values of pressure and CO_2 saturation. The calculated temporal inputs are then checked against the boundary assumptions of the underlying reduced order model. The Cemented Wellbore component model temporal inputs are:

- **deltaP** $[Pa]$ (105891.5 to 9326181.69) - difference between the current and initial pressure at the wellbore
- **pressurePrime** $[Pa/s]$ (-6675.03 to 2986.7) - first pressure derivative
- **pressureDPrime** $[Pa/s^2]$ (-111.265 to 10.806) - second pressure derivative
- **saturation** [-] (0.001 to 1.0) - CO_2 saturation at the wellbore
- **saturationPrime** $[1/s]$ (-4.290e-7 to 1.117e-3) - first CO_2 saturation derivative
- **saturationDPrime** $[1/s^2]$ (-6.923e-6 to 1.176e-6) - second CO_2 saturation derivative.

The possible outputs from the Cemented Wellbore component are leakage rates of CO_2 and brine to aquifer, thief zone and atmosphere. The names of the observations are of the form:

- **CO2_aquifer1**, **CO2_aquifer2**, **CO2_atm** $[kg/s]$ - CO_2 leakage rates

- **brine_aquifer1**, **brine_aquifer2**, **brine_atm** $[kg/s]$ - brine leakage rates
- **mass_CO2_aquifer1**, **mass_CO2_aquifer2** $[kg]$ - mass of CO_2 leaked into aquifers.

3.8 Cemented Wellbore (WR) Component

The Cemented Wellbore WR (wider ranges) component model is an updated version of cemented wellbore component available in NRAP-Open-IAM. The model is built off detailed full-physics Finite Element Heat and Mass (FEHM) simulations, [39]. The FEHM simulations are three-dimensional (3-D), multiphase solutions of heat and mass transfer of water and supercritical, liquid, and gas CO_2. After the simulations are completed, the surrogate model is built based on the key input parameters and corresponding output parameters. The approximate (surrogate) model is represented by polynomials in terms of input parameters that then can be sampled to estimate leakage rate for wells. Early development work can be found in [15].

When using the control file interface with more than 3 shale layers, the `ThiefZone` keyword can be used to specify the thief zone aquifer and the `LeakTo` keyword can be specified to name the upper aquifer. These values will default to *aquifer1* and *aquifer2* respectively. In the FEHM simulations used to create the surrogate model some of the stratigraphy layers were setup with a fixed thickness. In particular, shale above aquifer had thickness 9.6 m; shale layer between the aquifer and the thief zone varied between 228.4 m and 2902.9 m.

Component model input definitions:

- **logWellPerm** $[\log_{10} m^2]$ (-13.95 to -10.1) - logarithm of wellbore permeability (default: -13)
- **logThiefPerm** $[\log_{10} m^2]$ (-13.986 to -12.023) - logarithm of thief zone permeability (default: -12.5)
- **thiefZoneThickness** $[m]$ (0 to 99) - thickness of thief zone (default: 50); in 2 aquifers system linked to Stratigraphy parameter aquifer1Thickness
- **aquiferThickness** $[m]$ (10 to 230) - thickness of aquifer (default: 50); in 2 aquifers system linked to Stratigraphy parameter aquifer2Thickness
- **reservoirThickness** $[m]$ (16 to 193) - thickness of reservoir (default: 50); *linked to Stratigraphy*
- **wellRadius** $[m]$ (0.025 to 0.25) - radius of the wellbore (default: 0.05)
- **initPressure** $[Pa]$ (1.0e+5 to 5.0e+7) - initial pressure at the base of the wellbore (default: 2.0e+7 Pa, or 20 MPa); *from linked component*
- **wellDepth** $[m]$ (1056.0 to 3194.0) - depth in meters from ground surface to top of reservoir (default: 1500); *linked to Stratigraphy*
- **depthRatio** [-] (0.054 to 0.69798) - fraction of well depth to the center of the thief zone from the top of the reservoir (default: 0.5); *linked to Stratigraphy*.

Temporal inputs of the Cemented Wellbore component are not provided directly to the component model method but rather are calculated from the current and several past values of pressure and CO_2 saturation. The calculated temporal inputs are then checked against the boundary assumptions of the underlying reduced order model. The Cemented Wellbore component model temporal inputs are:

- **deltaP** $[Pa]$ (-1246850.9 to 8326262.6) - difference between the current and initial pressure at the wellbore
- **pressurePrime** $[Pa/s]$ (-5665.23 to 8898.32) - first pressure derivative
- **pressureDPrime** $[Pa/s^2]$ (-232.64 to 232.58) - second pressure derivative
- **saturation** [-] (4.63e-10 to 1.0) - CO_2 saturation at the wellbore
- **saturationPrime** $[1/s]$ (-2.638e-7 to 1.294e-3) - first CO_2 saturation derivative
- **saturationDPrime** $[1/s^2]$ (-8.806e-6 to 7.269e-6) - second CO_2 saturation derivative.

The possible outputs from the Cemented Wellbore component are leakage rates of CO_2 and brine to aquifer, thief zone and atmosphere. The names of the observations are of the form:

- **CO2_aquifer1**, **CO2_aquifer2**, **CO2_atm** $[kg/s]$ - CO_2 leakage rates
- **brine_aquifer1**, **brine_aquifer2**, **brine_atm** $[kg/s]$ - brine leakage rates
- **mass_CO2_aquifer1**, **mass_CO2_aquifer2** $[kg]$ - mass of CO_2 leaked into aquifers.

3.9 Open Wellbore Component

The Open Wellbore model is a lookup table reduced order model based on the drift-flux approach, see [26]. This model treats the leakage of CO_2 up an open wellbore or up an open (i.e., uncemented) casing/tubing. The lookup table is populated using T2Well/ECO2N Ver. 1.0 [25], which treats the non-isothermal flow of CO_2 and brine up an open wellbore, allows for the phase transition of CO_2 from supercritical to gaseous, with Joule-Thompson cooling, and considers exsolution of CO_2 from the brine phase.

By default, when used within the control file interface the Open Wellbore is connected to the upper aquifer (e.g., aquifer 2 if there are 2 aquifers in the system). For user-defined scenarios the `LeakTo` keyword can be used to specify either the name of the aquifer (e.g., *aquifer1*) CO_2 leaks to or *atmosphere* for leakage to the atmosphere. The default value is *aquifer#* where # is an index of the uppermost aquifer.

The Open Wellbore component can be used to calculate leakage rates following positive change in reservoir pressure or only from changes in reservoir pressure above a critical pressure. To use the latter approach, the argument `crit_pressure_approach` should be set to *True* for the setup of the component. Here is an example of this setup in a script application:

> OpenWellbore(name='ow', parent=sm, crit_pressure_approach=True)

To set `crit_pressure_approach` to *True* in the control file interface, the `Controls` section in the .yaml entry for the Open Wellbore should be included with an additional entry `critPressureApproach: True` indented beneath `Controls`. For an example setup, see control file example 31a.

If `crit_pressure_approach` is set to *True*, the default approach is for critical pressure to be calculated as:

> Pcrit = (rho_w * g * d_aq) + (rho_br * g * (d_res - d_aq)),

where rho_w and rho_br are the densities of water and brine (by default, 1000 kg/m^3) defined by the brineDensity parameter, respectively, g is gravitational acceleration(9.8 m/s^2), d_aq is the depth to the bottom of the aquifer impacted by leakage (m) (defined by the wellTop parameter value; if wellTop is 0 m, then the atmosphere receives leakage), and d_res is the depth to the top of the reservoir (m). Higher brine densities generally produce lower leakage rates.

Instead of calculating critical pressure in this manner, one can enforce a particular critical pressure (the **critPressure** parameter) by setting the argument `enforce_crit_pressure` to *True* for the setup of the component. Here is an example in a script setup:

> **OpenWellbore(name='ow', parent=sm, crit_pressure_approach=True,** enforce_crit_pressure=True)

To set `enforce_crit_pressure` to *True* in the control file interface, file interface, the `Controls` section in the .yaml entry for the Open Wellbore should be included with additional entry `enforceCritPressure: True` indented beneath `Controls`. If enforce_crit_pressure is not set to *True*, then the **critPressure** parameter will not be used.

When a critical pressure is used, flow through an Open Wellbore can still occur at pressures beneath the critical pressure if a CO_2 plume is present in the reservoir at the base of the well (i.e., buoyancy effects from the CO_2).

Component model input definitions:

- **logReservoirTransmissivity** $[\log_{10} m^3]$ (-11.27 to -8.40) - reservoir transmissivity (default: -9.83)
- **logAquiferTransmissivity** $[\log_{10} m^3]$ (-11.27 to -8.40) - reservoir transmissivity (default: -9.83)

- **brineSalinity** [-] (0 to 0.2) - brine salinity (mass fraction) (default: 0.1)
- **brineDensity** [kg/m^3] (900 to 1200) - brine density (default: 1012)
- **wellRadius** [m] (0.025 to 0.25) - radius of the wellbore (default: 0.05)
- **wellTop** [m] (0 to 500) - depth of well top (default: 500); *linked to Stratigraphy*. Note that this parameter represents how far leakage can extend from the reservoir along the open wellbore. For example, the cement used to plug the well may be of poor quality or damaged, but the damage may not allow leakage to reach the surface at 0 m. When using control files, wellTop can be set to the bottom depth of an aquifer by entering 'aquifer#Depth,' where # is the aquifer number. If the aquifer is too deep, however, the limits for this parameter would still be enforced. In control files, if the 'LeakTo' entry is provided as the name of the aquifer receiving leakage from the open wellbore (e.g., 'LeakTo: aquifer2') then the wellTop parameter will automatically be set to the bottom depth of the corresponding aquifer. If 'LeakTo' is set to 'atmosphere', wellTop will automatically be set to 0.
- **critPressure** [Pa] (1.0e+5 to 9.0e+7) - pressure above which the model initiates leakage rates calculations. Default value of this parameter is not defined: either the user provides it through component setup or the value is calculated based on the value of **brineDensity** parameter.
- **reservoirDepth** [m] (1000 to 4000) - depth of reservoir (well base) (default: 2000); *linked to Stratigraphy*. Note that if 'shale1Depth' is entered for this parameter in a control file, the parameter will be set to the bottom depth of shale 1 (which is also the top of the reservoir).

The possible outputs from the Open Wellbore component are leakage rates of CO_2 and brine to aquifer and atmosphere. The names of the observations are of the form:

- **CO2_aquifer** and **CO2_atm** [kg/s] - CO_2 leakage rates
- **brine_aquifer** and **brine_atm** [kg/s] - brine leakage rates.

3.10 Generalized Flow Rate Component

The Generalized Flow Rate component model is a model representing wide range of carbon dioxide (CO_2) and brine leakage flow rates and created based on the results of multiple wellbore simulations. The generalized models facilitate the implementation of flow rates in an uncertainty quantification (UQ) framework since the relevant leakage rate and time parameters can be generated randomly. The basic shape for these models were constructed from the results of numerical wellbore simulations based on pressure and saturation profiles derived from the Kimberlina reservoir model [33] coupled with wellbore permeability to yield CO_2 and complimentary brine leakage functions. More details covering derivation and application of the model can be found in [21].

In the IAM control file, the type name for the Generalized Flow Rate component is `GeneralizedFlowRate`. The description of the possible component's parameters are provided below.

- **numberOfShaleLayers** [-] (3 to 30) - number of shale layers in the system (default: 3); *linked to Stratigraphy*. The shale units must be separated by an aquifer.
- **logPeakCO2Rate** [$\log_{10}$ kg/s] (-inf to 5) - logarithm of the largest CO_2 flow rate (default: -5)
- **timePeakCO2Rate** [years] (0 to 1000) - time to reach the largest CO_2 flow rate from initial time (default: 5)
- **durationPeakCO2Rate** [years] (0 to 1000) - length of time period during which CO_2 flow rate was the largest (default: 10)
- **durationPeakZeroCO2Rate** [years] (0 to 1000) - length of time period during which CO_2 flow rate decreased from the largest rate to zero (default: 100)
- **logInitBrineRate** [$\log_{10}$ kg/s] (-inf to 5) - logarithm of the initial brine flow rate (default: -10)
- **logFinalBrineRate** [$\log_{10}$ kg/s] (-inf to 5) - logarithm of the final brine flow rate (default: -11.5). Ratio of initial brine rate over final brine rate is recommended to be between 0.2 and 0.3

- **durationInitBrineRate** [years] (0 to 1000) - length of initial brine flow rate time period (default: 2)
- **durationInitFinalBrineRate** [years] (0 to 1000) - length of time period during which brine flow rate decreased from initial to final rate (default: 10)
- **mitigationTime** [years] (0 to inf) - time at which the leakage was remediated (default: 10000)

The possible outputs from the Generalized Flow Rate component are leakage rates of CO_2 and brine to the aquifer specified by user. The names of the observations are of the form:

- **CO2_aquifer#** $[kg/s]$ - CO_2 leakage rates where # is an aquifer index
- **brine_aquifer#** $[kg/s]$ - brine leakage rates
- **mass_CO2_aquifer#** $[kg]$ - mass of CO_2 leaked into the specified aquifer.

3.11 Hydrocarbon Leakage Component

The HydrocarbonLeakage component model is a reduced order model predicting liquid and gas leakage to the shallow aquifer between 100 to 410 years post-injection of CO_2 to depleted hydrocarbon field. The model output begins 100 years and extends to 410 years after injection stops. The component is based on a machine learning regression model fitted to the results of compositional multiphase flow transport simulations using a neural network. Total 192,000 data from ~1,000 numerical simulations were used to develop the model. The model predicts CO_2 and methane leakage in liquid and gas phases to a shallow aquifer. The model also predicts the total liquid (oil) and gas leakage to the shallow aquifer, where these total masses include all hydrocarbons (light, intermediate, and heavy) as well as CO_2. The depth to the bottom of the shallow aquifer is assumed to be 60 ft (18.288 m) below the surface. The top of the shallow aquifer extends to the surface. Input parameters were sampled using Latin Hypercube Sampling across wide ranges.

Values of input parameters FCO2, FC1, FC4, and FC7Plus must sum to one. To allow some leniency (e.g., for issues related to rounding errors), the sum of these values must be within 0.0001 (0.01%) of one (0.9999 to 1.0001). This option was created for cases when the sum of the values is different from 1 by a small value (e.g., 1.0e-6). If the sum of the provided values is not sufficiently close to one, then a warning message is printed.

Since the temporal bounds for the HydrocarbonLeakage component are years 100 to 410 after injection stops, the component produces zero results for any times outside of this range. Any results outside the applicable time range should not be considered valid, however.

The description of the component's parameters is presented below:

- **reservoirDepth** $[m]$ (914.4 to 2743.2) - depth to the top of the reservoir (default: 2000); *linked to Stratigraphy*
- **NTG** [-] (0.4 to 1.0) - net-to-gross ratio representing the fraction of reservoir contributing to the flow (default: 0.6)
- **logResPerm** [log10 m^2] (-14.0057 to -13.0057) - logarithm of reservoir permeability (default: -13.5)
- **reservoirPressureMult** [-] (1.0 to 1.2) - factor used to represent a state of the reservoir pressurization post-injection (relative to the reservoir pressure calculated lithostatically) (default: 1.1)
- **logWellPerm** [log10 m^2] (-17.0057 to -12.0057) - logarithm of wellbore permeability (default: -13.0)
- **avgWaterSaturation** [-] (0.471 to 0.792) - average water saturation in the reservoir (default: 0.5)
- **FCO2** [-] (0.432 to 0.693) - mole fraction of CO_2 in the reservoir post-injection (default: 0.55)
- **FC1** [-] (0.010 to 0.113) - mole fraction of methane in the reservoir post-injection (default: 0.05)
- **FC4** [-] (0.010 to 0.111) - mole fraction of intermediate hydrocarbons in the reservoir post-injection (default: 0.05)
- **FC7Plus** [-] (0.123 to 0.500) - mole fraction of heavy hydrocarbons in the reservoir post-injection (default: 0.35)

Component model outputs:

- **mass_oil_aquifer** $[kg]$ - cumulative mass of oil leaked to the aquifer. This output includes all hydrocarbons (light, intermediate, and heavy hydrocarbons) and CO_2 in the liquid phase.
- **mass_gas_aquifer** $[kg]$ - cumulative mass of gas leaked to the aquifer. This output includes all hydrocarbons (light, intermediate, and heavy hydrocarbons) and CO_2 in the gas phase.
- **mass_methane_gas_aquifer** $[kg]$ - cumulative mass of methane gas leaked to the aquifer
- **mass_methane_oil_aquifer** $[kg]$ - cumulative mass of the methane in oil phase leaked to the aquifer
- **mass_CO2_aquifer** $[kg]$ - cumulative mass of liquid CO_2 leakage to aquifer
- **mass_CO2_gas_aquifer** $[kg]$ - cumulative mass of CO_2 gas leaked to the aquifer

3.12 Seal Horizon Component

The Seal Horizon component model simulates the flow of CO_2 through a low permeability but fractured rock horizon (a "seal" formation) overlying the storage reservoir into which CO_2 is injected.

The rock horizon is represented by a number of "cells" arranged (conceptually) in an arbitrary shape grid. A two-phase, relative permeability approach is used with Darcy's law for one-dimensional (1D) flow computations of CO_2 through the horizon in the vertical direction. The code also allows the simulation of time-dependent processes that can influence such flow.

The model is based on an earlier code, NSealR, created with GoldSim, and described in [19]. A stand-alone version of this code in Python is also available on the NETL EDX system, described as Seal ROM.

In the NRAP-Open-IAM control file, the type name for the component is SealHorizon. The following is a list of the component parameters, including the parameter names, units, accepted value range and the default value.

Reference parameters for each cell:

- **area** $[m^2]$ (1 to 2.6e+5) - area of the cell (default: 10000.0)
- **thickness** $[m]$ (5 to 1000) - thickness of the cell (vertically) (default: 100)
- **baseDepth** $[m]$ (800 to 9500) - depth to the base of seal (default: 1100)
- **permeability** $[m^2]$ (1.0e-22 to 1.0e-15) - cell equivalent initial permeability (default: 1.0e-18)
- **entryPressure** $[Pa]$ (100 to 2.0e+6) - entry threshold pressure that controls flow into rock (default: 5000)

Distribution parameters for thickness of the seal layer

- **aveThickness** $[m]$ (10 to 1000) - mean of the truncated normal distribution for thickness (default: 100)
- **stdDevThickness** $[m]$ (0 to 500) - standard deviation of the thickness distribution (default: 0)
- **minThickness** $[m]$ (5 to 1000) - minimum thickness; this value truncates the distribution and limits lower values (default: 75)
- **maxThickness** $[m]$ (10 to 1000) - maximum thickness; this value truncates the distribution and limits higher values (default: 125)

Note: The setup of the four distribution parameters above is not yet implemented in the control file interface or GUI of NRAP-Open-IAM and available only in the script interface.

Distribution parameters for permeability of the seal layer:

- **avePermeability** $[m^2]$ (1.0e-22 to 1.0e-16) - mean total vertical permeability of a lognormal distribution; equivalent value for fractured rock (default: 2.5e-16)

- **stdDevPermeability** [m^2] (0 to 1.0e-17) - standard deviation of the total vertical permeability distribution (default: 0.0)
- **minPermeability** [m^2] (1.0e-24 to 1.0e-17) - minimum total vertical permeability; this value truncates (censors) the vertical random distribution and limits lower values (default: 1.0e-18)
- **maxPermeability** [m^2] (1.0e-21 to 1.0e-12) - maximum total vertical permeability; this value truncates (censors) the random distribution and limits higher values (default: 1.0e-15)

Note: The setup of the four distribution parameters above is not yet implemented in the control file interface or GUI of NRAP-Open-IAM and available only in the script interface.

- **heterFactor** [-] (1.0e-2 to 100) - increase factor of the permeability of cells selected for heterogeneity, if the heterogeneity approach is used (default: 0.5).

Reference parameters for all cells:

- **aveBaseDepth** [m] (800 to 9500) - average depth to base of cell/reservoir top; interpolation depth (default: 1100)
- **aveBasePressure** [Pa] (1.0e+6 to 6.0e+7) - average pressure at seal base during injection (default: 3.3e+7)
- **aveTemperature** [$°C$] (31 to 180) - average temperature of seal (default: 50)
- **salinity** [ppm] (0 to 80000) - average salinity of seal (default: 1.5e+4)
- **staticDepth** [m] (800 to 9500) - reference depth for computing static pressure at top of seal (default: 1000)
- **staticPressure** [Pa] (1.0e+6 to 6.0e+7) - pressure at static reference depth for computing pressure at the cell top (default: 1.0e+7).

Fluid (conditions) parameters:

- **brineDensity** [kg/m^3] (880 to 1080) - density of brine phase (default: 1004)
- **CO2Density** [kg/m^3] (93 to 1050) - density of CO_2 phase (default: 597.8)
- **brineViscosity** [$Pa{\cdot}s$] (1.5e-4 to 1.6e-3) - viscosity of brine phase (default: 5.634e-4)
- **CO2Viscosity** [$Pa{\cdot}s$] (1.8e-5 to 1.4e-4) - viscosity of CO_2 phase (default: 4.452e-5)
- **CO2Solubility** [mol/kg] (0 to 2) - solubility of CO_2 phase in brine (default: 0.035).

Two-phase model parameters for LET model:

- **wetting1** [-] (0.5 to 5) - wetting phase parameter L (default: 1)
- **wetting2** [-] (0.1 to 30) - wetting phase parameter E (default: 10)
- **wetting3** [-] (0 to 3) - wetting phase parameter T (default: 1.25)
- **nonwet1** [-] (0.5 to 5) - nonwetting phase parameter L (default: 1.05)
- **nonwet2** [-] (0.1 to 30) - nonwetting phase parameter E (default: 10)
- **nonwet3** [-] (0 to 3) - nonwetting phase parameter T (default: 1.25)
- **capillary1** [-] (0.01 to 5) - LET-model parameter L for capillary pressure (default: 0.2)
- **capillary2** [-] (0.01 to 30) - LET-model parameter E for capillary pressure (default: 2.8)
- **capillary3** [-] (0.01 to 3) - LET-model parameter T for capillary pressure (default: 0.43)
- **maxCapillary** [Pa] (100 to 2.0e+8) - maximum capillary pressure for model (default: 1.0e+7)

Note: Parameters **wetting1**, **wetting2**, **wetting3**, **nonwet1**, **nonwet2**, **nonwet3**, **capillary1**, **capillary2**, **capillary3**, and **maxCapillary** are used only if parameter **relativeModel** is set to *LET*.

Parameters for BC model:

- **lambda** [-] (0 to 5) - lambda parameter in Brooks-Corey model (default: 2.5)

Note: Parameter **lambda** is used only if parameter/keyword argument **relativeModel** is set to *BC*.

Additional parameters for two-phase flow:

- **brineResSaturation** [-] (0.01 to 0.35) - residual brine saturation (default: 0.15)
- **CO2ResSaturation** [-] (0 to 0.35) - residual CO_2 saturation (default: 0)
- **relativeModel** [-] (LET or BC) - relative permeability model (default: LET)
- **permRatio** [-] (0 to 1.5) - ratio of nonwetting to wetting permeability (default: 0.6).

Time-model and rock type parameters:

- **influenceModel** [-] (integer: 0, 1, 2) - time-dependent permeability model (default: 0); deterministic parameter, i.e. cannot be set to be random. Model type used to compute the influence factor of the fluid flow on permeability for time-dependent response:
  - 0: No influence factor used.
  - 1: Use a time-dependent model based on exposure time to CO_2. Parameters **rateEffect** and **totalEffect** control the initial time delay and the maximum extent of effect.
  - 2: Use a multivariant model that considers **reactivity**, **clayType**, **clayContent** and **carbonateContent** values together with **rateEffect** and **totalEffect** parameters to establish the magnitude of the influence factor.
- **influence** [-] (0 to 1) - initial permeability influence factor (default: 1)
- **rateEffect** [-] (0.01 to 0.65) - time variance parameter; this parameter controls the initial time delay in the permeability effect of the model (default: 0.1)
- **totalEffect** [-] (0.01 to 200) - time variance parameter; this parameter defines the total change in permeability of the model (as a factor) (default: 0.1)
- **reactivity** [-] (0 to 10) - reactivity of time model; factor controls the magnitude of permeability change (default: 8)
- **clayType** [-] (smectite, illite, or chlorite) - predominate clay mineral content in the seal horizon, defined as one of following categories:
  - smectite (high swelling material)
  - illite (moderate swelling material)
  - chlorite (low swelling material) (default: smectite)
- **carbonateContent** [%] (0 to 100) - carbonate content in seal layer rock (default: 8)
- **clayContent** [%] (0 to 100) - clay mineral content in seal layer rock (default: 60).

Note: Parameters **rateEffect** and **totalEffect** are used only when parameter **influenceModel** is set to 1 or 2. These parameters control the initial time delay and the maximum extent of effect.

Note: Parameters **reactivity**, **clayType**, **carbonateContent**, and **clayContent** are used only when parameter **influenceModel** is set to 2.

The possible outputs from the Seal Horizon component are leakage rates of CO_2 and brine to aquifer through seal layer. The names of the observations are of the form:

- **CO2_aquifer**, **brine_aquifer** $[kg/s]$ - CO_2 and brine leakage rates to aquifer through seal layer (individual cells) into overlying aquifer
- **mass_CO2_aquifer**, **mass_brine_aquifer** $[kg]$ - mass of the CO_2 and brine leaked through seal layer (individual cells) into overlying aquifer

- **CO2_aquifer_total**, **brine_aquifer_total** $[kg/s]$ - cumulative (for all cells) CO_2 and brine leakage rates to aquifer through seal layer into overlying aquifer
- **mass_CO2_aquifer_total**, **mass_brine_aquifer_total** $[kg]$ - cumulative (for all cells) mass of the CO_2 and brine leaked through seal layer into overlying aquifer.

3.13 Fault Flow Component

The Fault Flow component model simulates the flow of carbon dioxide along a low permeability fault from an injection horizon (into which carbon dioxide is injected) up to a freshwater aquifer. The theoretical base is predicated on one-dimension (1D), steady-state, two-phase flow of CO_2 through a saturated discontinuity (parallel plates) under CO_2 supercritical conditions. The flow in the current implementation uses the near-surface CO_2 supercritical point to be the upper point of flow. The surrounding rock matrix is considered relatively impermeable.

In the NRAP-Open-IAM control file, the type name for the Fault Flow component is `FaultFlow`. The description of the component's parameters is provided below:

Fault core setup parameters:

- **strike** [°] (0 to 360) - direction of fault: trend of fault strike taken clockwise from north (default: 30)
- **dip** [°] (10 to 90) - inclination of fault plane from strike, using right-hand rule from strike (default: 70)
- **length** $[m]$ (0 to 10,000) - length of fault trace at surface from start point (default: 100)
- **xStart** $[m]$ (-5.0e+07 to 5.0e+07) - x-coordinate of the fault start point taken as the left point on fault trace (default: 500)
- **yStart** $[m]$ (-5.0e+07 to 5.0e+07) - y-coordinate of the fault start point taken as the left point on fault trace (default: 500)
- **nSegments** [-] (1 to 100) - number of separate fault divisions of the fault (default: 4)
- **faultProbability** [%] (0 to 100) - probability of fault existence (default: 100)

Fault aperture setup parameters:

- **aperture** $[m]$ (0 to 0.05) - effective aperture of fault (default: 2.5e-6)
- **SGR** [-] (0 to 100) - shale gouge ratio for fault (default: 0)
- **stateVariable** [-] (0 to 1) - correction factor for near-surface flow (default: 1)

For variability of fault properties and orientation setup one can use the following eight distribution parameters.

Note: The setup of the eight distribution parameters below is not yet implemented in the control file interface or GUI of NRAP-Open-IAM and available only in the script interface.

Strike distribution parameters:

- **aveStrike** [°] (0 to 360) - average direction of fault: average trend of fault strike taken clockwise from north (default: 90); also default value for no variation in strike
- **spreadStrike** [°] (0 to 180) - spread in strike orientation (range of 2-sigma around average) (default: 0)

Dip distribution parameters:

- **aveDip** [°] (10 to 90) - average inclination of fault plane from strike, using right-hand rule from strike (default: 90)
- **stdDevDip** [°] (0 to 90) - standard deviation of angle of dip (default: 0)

Aperture distribution parameters:

- **aveAperture** $[m]$ (0 to 1.01e-1) - average effective aperture of fault (default: 1.0e-2)
- **stdDevAperture** $[m]$ (0 to 2.0e-2) - standard deviation of effective aperture (default: 0.0)
- **minAperture** $[m]$ (0 to 1.0e-3) - minimum aperture (default: 1.0e-7)
- **maxAperture** $[m]$ (0 to 5.0e-2) - maximum aperture (default: 2.0e-2)

Field parameters:

- **aquiferDepth** $[m]$ (200 to 2,000) - depth to base of deepest aquifer along/above fault (default: 240)
- **aquiferTemperature** $[°C]$ (15 to 180) - temperature of brine of deepest aquifer at base (default: 22)
- **aquiferPressure** $[Pa]$ (1.0e+6 to 6.0e+8) - pressure at base of aquifer (default: 1.42E+07)
- **injectDepth** $[m]$ (860 to 20,000) - reference depth positive below grade to top of injection horizon (default: 1880)
- **injectTemperature** $[°C]$ (31 to 180) - average temperature of brine at injection depth in reservoir (default: 95)
- **fieldPressure** $[Pa]$ (1.0e+5 to 6.0e+7) - initial pressure at injection depth before injection starts (default: 1.9140e+07)
- **injectPressure** $[Pa]$ (7.0e+6 to 6.0e+8) - average pressure at base during injection period for interpolation of viscosity and density (default: 2.9290E+07)
- **finalPressure** $[Pa]$ (1.0e+5 to 6.0e+7) - final average pressure at injection depth for interpolation of viscosity and density (default: 1.9140e+07)
- **injectX** $[m]$ (-5.0e+07 to 5.0e+07) - x-coordinate of the location of injection well (default: 0)
- **injectY** $[m]$ (-5.0e+07 to 5.0e+07) - y-coordinate of the location of injection well (default: 0)
- **injectEndTime [years] (0 to 10000) - time when injection stops** (default: 50)

Reservoir conditions parameters:

- **salinity** $[ppm]$ (0 to 80000) - salinity of the brine (default: 0). The value is used to compute density and viscosity of the brine
- **CO2Density** $[kg/m^3]$ (93 to 1050) - average density of CO_2 phase for fault (default: 673.84). The value is used if interpolation is not conducted by code
- **CO2Viscosity** $[Pa{\cdot}s]$ (1.8e-05 to 1.4e-04) - viscosity of CO_2 phase for fault (default: 5.5173e-05). The value is used if interpolation is not conducted by code
- **brineDensity** $[kg/m^3]$ (880 to 1080) - density of brine phase for fault (default: 974.895). The value is used if interpolation is not conducted by code
- **brineViscosity** $[Pa{\cdot}s]$ (1.5e-04 to 1.6e-03) - viscosity of brine phase for fault (default: 3.0491e-04). The value is used if interpolation is not conducted by code
- **CO2Solubility** $[mol/kg]$ (0 to 2) - solubility of CO_2 phase in brine for fault (default: 0.035). The value is used if interpolation is not conducted by code

Aquifer conditions parameters:

- **aquiferCO2Density** $[kg/m^3]$ (93 to 1050) - density of CO_2 phase in the aquifer (default: 886.44)
- **aquiferCO2Viscosity** $[Pa{\cdot}s]$ (1.1e-05 to 1.4e-04) - viscosity of CO_2 phase in the aquifer (default: 8.8010e-05)
- **aquiferBrineDensity** $[kg/m^3]$ (880 to 1080) - density of brine phase in the aquifer (default: 1004.10)
- **aquiferBrineViscosity** $[Pa{\cdot}s]$ (1.5e-04 to 1.6e-03) - viscosity of brine phase in the aquifer (default: 3.0221e-04)

Relative flow parameters:

- **brineResSaturation** [-] (0.01 to 0.35) - residual wetting brine saturation used in two-phase model (default: 0.15)
- **CO2ResSaturation** [-] (0 to 0.35) - residual nonwetting CO_2 saturation used in two-phase model (default: 0.0)
- **relativeModel** [-] (LET or BC) - relative permeability model (default: LET)
- **permRatio** [-] (0 to 1.5) - ratio of maximum nonwetting permeability to the maximum wetting permeability (default: 0.6)
- **entryPressure** $[Pa]$ (100 to 2.0e+6) - entry/threshold/bubbling pressure that controls flow into rock (default: 5000)

Two-phase model parameters for LET model:

- **wetting1** [-] (0.5 to 5) - wetting phase parameter L (default: 1)
- **wetting2** [-] (1 to 30) - wetting phase parameter E (default: 10)
- **wetting3** [-] (0 to 3) - wetting phase parameter T (default: 1.25)
- **nonwet1** [-] (0.5 to 5) - nonwetting phase parameter L (default: 1.05)
- **nonwet2** [-] (1 to 30) - nonwetting phase parameter E (default: 10)
- **nonwet3** [-] (0 to 3) - nonwetting phase parameter T (default: 1.25)
- **capillary1** [-] (0.01 to 5) - LET-model parameter L for capillary pressure (default: 0.2)
- **capillary2** [-] (0.01 to 30) - LET-model parameter E for capillary pressure (default: 2.8)
- **capillary3** [-] (0.01 to 3) - LET-model parameter T for capillary pressure (default: 0.43)
- **maxCapillary** $[Pa]$ (100 to 2.0e+8) - maximum capillary pressure for model (default: 1.0e+7)

Note: Parameters **wetting1**, **wetting2**, **wetting3**, **nonwet1**, **nonwet2**, **nonwet3**, **capillary1**, **capillary2**, **capillary3**, and **maxCapillary** are used only if parameter **relativeModel** is set to *LET*.

BC model parameters:

- **lambda** [-] (0 to 5) - lambda term in Brooks-Corey model (default: 2.5)

Note: Parameter **lambda** is used only if parameter **relativeModel** is set to *BC*.

Stress parameters:

- **maxHorizontal** $[Pa]$ (0 to 5.0e+7) - secondary maximum horizontal principal stress at top of injection horizon (default: 3.0e+7)
- **minHorizontal** $[Pa]$ (0 to 5.0e+7) - secondary minimum horizontal principal stress at top of injection interval (default: 2.0e+7)
- **maxTrend** [°] (0 to 180) - strike of secondary maximum horizontal stress clockwise from north (default: 55)

The possible outputs from the Fault Flow component are leakage rates of CO_2 and brine to aquifer through fault. The names of the observations are of the form:

- **CO2_aquifer**, **brine_aquifer** $[kg/s]$ - CO_2 and brine leakage rates to aquifer through fault (individual segments) into overlying aquifer
- **mass_CO2_aquifer**, **mass_brine_aquifer** $[kg]$ - mass of the CO_2 and brine through fault (individual segments) to overlying aquifer
- **CO2_aquifer_total**, **brine_aquifer_total** $[kg/s]$ - cumulative CO_2 and brine leakage rates to aquifer through fault into overlying aquifer
- **mass_CO2_aquifer_total**, **mass_brine_aquifer_total** $[kg]$ - cumulative mass of the CO_2 and brine through fault (individual cells) to overlying aquifer.

Observations with names CO2_aquifer, brine_aquifer, mass_CO2_aquifer and mass_brine_aquifer are provided as arrays of values of length equal to the number of fault segments. To output observations corresponding to a particular fault segment (e.g., segment 1) one can add observations with names CO2_aquifer_segm# where # is an index of a segment of interest (e.g., CO2_aquifer_segm1) to the output of the Fault Flow component.

3.14 Fault Leakage Component

The Fault Leakage Model component uses deep neural networks to estimate the flow of brine carbon dioxide along a fault. The dynamics of multiphase flow in the storage reservoir and shallow aquifer into which the fault leaks are both taken into consideration in the estimation of leakage rates. The CO_2 is treated as supercritical throughout the leakage process, including within the shallow aquifer. No phase change occurs. The fault is modeled as a continuous, homogeneous, isotropic porous medium with Darcy flow. This component assumes the following:

- The storage reservoir is 4000 m long (distance away from the fault), 2400 m wide, and 200 m thick.
- The shallow aquifer has similar dimensions to the storage reservoir, but is located on the opposite side of the fault.
- The fault has a thickness of 3 m and a width of 2400 m, contacting the entire width of both the storage reservoir and the shallow aquifer.
- The fault dip angle varies.
- There is a caprock on both sides of the fault with a thickness of 100 m perpendicular to the fault. The caprock has a permeability of 1.0e-18 (impermeable) and a porosity of 0.1.
- The storage reservoir has no-flow boundaries.
- The shallow aquifer has a constant pressure boundary at its side boundary. All other boundaries are no-flow.
- The domain is 1750 m in depth from the top of the shallow aquifer to the bottom of the storage reservoir.
- The pressure and temperature at the top of the shallow aquifer are 8.15 MPa and 50 $^{\circ}C$, respectively.
- The CO_2 injection temperature is 32 $^{\circ}C$.
- The thermal conductivity of rock is 3 W/(m*K).
- The specific heat capacity of rock is 920 J/(kg*K).
- Well is located about 178-190 m above the bottom of the storage reservoir.

The description of the component's parameters is provided below:

- **damage_zone_perm** [$\log_{10} m^2$] (-15 to -12) - the permeability of the fault, including both the fault core and damage zone (default: -13.5)
- **damage_zone_por** [-] (0.001 to 0.1) - the porosity of the fault, including both the fault core and damage zone (default: 0.01)
- **shallow_aquifer_perm** [$\log_{10} m^2$] (-14 to -12) - the permeability of the shallow aquifer (default: -13.0)
- **deep_aquifer_perm** [$\log_{10} m^2$] (-14 to -12) - the permeability of the storage aquifer (e.g., reservoir) (default: -13.0)
- **shallow_aquifer_por** [-] (0.05 to 0.5) - the porosity of the shallow aquifer (default: 0.25)
- **deep_aquifer_por** [-] (0.05 to 0.35) - the porosity of the deep aquifer (default: 0.2)
- **well_index** [-] (integer: 0, 1, 2) - a proxy for the horizontal distance of the well from the fault; value of 0 means that the well is about 200 m from the fault, value of 1 means that the well is about 400 m from the fault; value of 2 means that the well is about 600 m from the fault (default: 0)

- **well_rate** [kg/s] (0.5 to 25) - injection rate of the well for the aquifer (default: 15.8)
- **dip_angle** [°] (integer: 40, 60, 80, 100, 120, 140) - dip angle of the fault, measured from the horizontal plane (default: 60)
- **injection_time** [$years$] (10 to 50) - duration of injection (default: 30)
- **geothermal_gradient** [$°C/km$] (8 to 44) - the geothermal gradient in the formation (default: 30)

The possible outputs from the Fault Leakage component are the leakage rates and cumulative leakage amounts of brine and CO_2 to the shallow aquifer through the fault. The names of the observations are:

- **brine_aquifer**, **CO2_aquifer** [kg/s] - brine and CO_2 leakage rates to the shallow aquifer, respectively.
- **mass_brine_aquifer**, **mass_CO2_aquifer** [kg] - cumulative brine and CO_2 leakage into the shallow aquifer, respectively.

Notes:

- Due to the use of the trapezoidal rule in integrating instantaneous leakage rates, the cumulative leakage values might not conserve mass for CO_2 depending on the time step size used.
- Leakage estimates are available only up to 100 years.
- After extensive sensitivity analysis the key parameters for the model have been found to be: damage zone permeability, deep aquifer permeability, injection rate of the well, injection duration, and dip angle. The time at which the leakage rate or amount is to be estimated (i.e., simulation time) is also a key parameter. Uncertainties in these variables are, therefore, most important.

3.15 Carbonate Aquifer Component

The Carbonate Aquifer component model is a reduced-order model that can be used to predict the impact that carbon dioxide (CO_2) and brine leaks from a CO_2 storage reservoir might have on overlying aquifers. The model predicts the size of "impact plumes" according to nine water quality metrics, see [2], [7], [17].

Although the Carbonate Aquifer model was developed using site-specific data from the Edwards aquifer, the model accepts aquifer characteristics as variable inputs and, therefore, may have more broad applicability. Careful consideration should be given to the hydrogeochemical character of the aquifer before using this model at a new site. Guidelines and examples are presented in [16].

The size of "impact plumes" are calculated using two alternative definitions of "impact" which should be selected by user: 1) changes that cause an exceedance of a drinking water standard or maximum contaminant level (MCL); and 2) changes that are above and beyond "natural background variability" in the aquifer, [18].

Component model input definitions:

- **ithresh** [-] (1 or 2) - threshold, either 1: MCL or 2: No-impact (default: 2)
- **rmin** [m] (0 to 100) - maximum distance between leaks for them to be considered one leak (default: 15)
- **perm_var** [$\log_{10} m^4$] (0.017 to 1.89) - logarithm of permeability variance (default: 0.9535)
- **corr_len** [m] (1 to 3.95) - correlation length (default: 2.475)
- **aniso** [-] (1.1 to 49.1) - anisotropy factor: ratio of horizontal to vertical permeability (default: 25.1)
- **mean_perm** [$\log_{10} m^2$] (-13.8 to -10.3) - logarithm of mean permeability (default: -12.05)
- **hyd_grad** [-] (2.88e-4 to 1.89e-2) - horizontal hydraulic gradient (default: 9.59e-03)
- **calcite_ssa** [m^2/g] (0 to 1.0e-2) - calcite surface area (default: 5.5e-03)
- **organic_carbon** [-] (0 to 1.0e-2) - organic carbon volume fraction (default: 5.5e-03)

- **benzene_kd** [$\log_{10}$ K_oc] (1.49 to 1.73) - benzene distribution coefficient (default: 1.61)
- **benzene_decay** [$\log_{10}$ day] (0.15 to 2.84) - benzene decay constant (default: 0.595)
- **nap_kd** [$\log_{10}$ K_oc] (2.78 to 3.18) - naphthalene distribution coefficient (default: 2.98)
- **nap_decay** [$\log_{10}$ day] (-0.85 to 2.04) - naphthalene decay constant (default: 0.595)
- **phenol_kd** [$\log_{10}$ K_oc] (1.21 to 1.48) - phenol distribution coefficient (default: 1.35)
- **phenol_decay** [$\log_{10}$ day] (-1.22 to 2.06) - phenol decay constant (default: 0.42)
- **cl** [$\log_{10}$ molality] (0.1 to 6.025) - brine salinity (default: 0.776)
- **logf** [-] (0 or 1) - type of transform of output plume volume; 0: linear, 1: log (default: 0)
- **aqu_thick** [m] (100 to 500) - aquifer thickness (default: 300); *linked to Stratigraphy*

Component model dynamic inputs:

- **brine_rate** [kg/s] (0 to 0.075) - brine rate
- **brine_mass** [kg] (0 to 2.0e+8) - cumulative brine mass
- **co2_rate** [kg/s] (0 to 0.5) - CO_2 rate
- **co2_mass** [kg] (0 to 2.0e+9) - cumulative CO_2 mass.

Possible observations from the Carbonate Aquifer component are:

- **pH_volume** [m^3] - volume of aquifer below pH threshold
- **Flux** [kg/s] - CO_2 leakage rate to atmosphere
- **dx** [m] - length of impacted aquifer volume in x-direction
- **dy** [m] - width of impacted aquifer volume in y-direction
- **TDS_volume** [m^3] - volume of aquifer above TDS threshold in mg/L
- **As_volume** [m^3] - volume of aquifer above arsenic threshold in $\mu g/L$
- **Pb_volume** [m^3] - volume of aquifer above lead threshold in $\mu g/L$
- **Cd_volume** [m^3] - volume of aquifer above cadmium threshold in $\mu g/L$
- **Ba_volume** [m^3] - volume of aquifer above barium threshold in $\mu g/L$
- **Benzene_volume** [m^3] - volume of aquifer above benzene threshold
- **Naphthalene_volume** [m^3] - volume of aquifer above naphthalene threshold
- **Phenol_volume** [m^3] - volume of aquifer above phenol threshold.

3.16 Deep Alluvium Aquifer Component

The Deep Alluvium Aquifer component model is a reduced order model which can be used to predict the changes in diluted groundwater chemistry if CO_2 and brine were to leak into a deep alluvium aquifer similar to the one located below the Kimberlina site, in the Southern San Joaquin Valley, California. The protocol allows uncertainty and variability in aquifer heterogeneity, fluid transport, and potential CO_2 and brine leakage rates from abandoned or damaged oil and gas wells to be collectively evaluated to assess potential changes in groundwater pH, total dissolved solids (TDS), and changes in the aquifer pressure resulting from leakage.

Although the Deep Alluvium Aquifer model was developed using site-specific data from the LLNL's Kimberlina Model (version 1.2), the model accepts aquifer characteristics as variable inputs and, therefore, may have broader applicability.

Careful consideration should be given to the hydrogeochemical character of the aquifer before using this model at a new site.

Model was created using the py-earth Python package [29]. Simulation data used to build this model was created by Mansoor et al. [20]. In the NRAP-Open-IAM control file, the type name for the Deep Alluvium Aquifer component is DeepAlluviumAquifer.

Component model input definitions:

- **logK_sand1** [$\log_{10}$ m^2] (-12.92 to -10.92) - permeability of layer 1 at depth between 10 and 546 m (default: -11.92)
- **logK_sand2** [$\log_{10}$ m^2] (-12.72 to -10.72) - permeability of layer 2 at depth between 546 and 1225 m (default: -11.72)
- **logK_sand3** [$\log_{10}$ m^2] (-12.7 to -10.7) - permeability of layer 3 at depth between 1225 and 1411 m (default: -11.70)
- **logK_caprock** [$\log_{10}$ m^2] (-16.699 to -14.699) - permeability of caprock at depth between 0 and 10 m (default: -15.70)
- **correlationLengthX** [m] (200 to 2000) - correlation length in x-direction (default: 1098.99)
- **correlationLengthZ** [m] (10 to 150) - correlation length in z-direction (default: 79.81)
- **sandFraction** [-] (0.7 to 0.9) - sand volume fraction (default: 0.8)
- **groundwater_gradient** [-] (0.001000 to 0.001667) - regional groundwater gradient (dh/dx=change in hydraulic head/distance) (default: 0.001333)
- **leak_depth** [m] (424.36 to 1341.48) - depth of leakage interval (default: 885.51).

Component model dynamic inputs:

- **brine_rate** [kg/s] (0 to 0.017) - brine rate (default: 0.0003)
- **brine_mass** [kg] (238.14419 to 8689604.29) - cumulative brine mass (default: 84722.74=10**4.928)
- **co2_rate** [kg/s] (0 to 0.385) - CO_2 rate (default: 0.045)
- **co2_mass** [kg] (1.002 to 1.621e+9) - cumulative CO_2 mass (default: 1.636e+7=10**7.214).

Observations from the Deep Alluvium Aquifer component are:

- **TDS_volume** [m^3] - volume of plume above baseline TDS change in mg/L (change in TDS > 100 mg/L)
- **TDS_dx** [m] - length of plume above baseline TDS change in mg/L (change in TDS > 100 mg/L)
- **TDS_dy** [m] - width of plume above baseline TDS change in mg/L (change in TDS > 100 mg/L)
- **TDS_dz** [m] - height of plume above baseline TDS change in mg/L (change in TDS > 100 mg/L)
- **Pressure_volume** [m^3] - volume of plume above baseline pressure change in Pa (change in pressure > 500 Pa)
- **Pressure_dx** [m] - length of plume above baseline pressure change in Pa (change in pressure > 500 Pa)
- **Pressure_dy** [m] - width of plume above baseline pressure change in Pa (change in pressure > 500 Pa)
- **Pressure_dz** [m] - height of plume above baseline pressure change in Pa (change in pressure > 500 Pa)
- **pH_volume** [m^3] - volume of plume below pH threshold (pH < 6.75)
- **pH_dx** [m] - length of plume below pH threshold (pH < 6.75)
- **pH_dy** [m] - width of plume below pH threshold (pH < 6.75)
- **pH_dz** [m] - height of plume below pH threshold (pH < 6.75).

3.17 FutureGen2 Aquifer Component

The FutureGen 2.0 Aquifer component model is a reduced order model that can be used to predict the impact that carbon dioxide (CO_2) and brine leakage from the CO_2 storage reservoir at the FutureGen 2.0 site might have on overlying aquifers or monitoring units. The model predicts the size of "impact plumes" according to four metrics: pH, total dissolved solids (TDS), pressure and dissolved CO_2.

The FutureGen 2.0 Aquifer model is a regression model fitted to the results of STOMP-CO2E-R multiphase flow and reactive transport simulations of CO_2 and brine leakage using the `py-earth` Python package ([29]). The `py-earth` package is a Python implementation of the Multivariate Adaptive Regression Splines algorithm ([8]), in the style of `scikit-learn` ([27]), a library of machine-learning methods.

The aquifer simulations used to train the FutureGen 2.0 Aquifer component model were based on modeling done for monitoring program design at the FutureGen 2.0 site ([32]), as well as porosity and permeability values from the ELAN logs and core samples taken from the characterization well. Isothermal simulations were performed for training the aquifer component model.

In the NRAP-Open-IAM control file, the type name for the FutureGen 2.0 Aquifer component is `FutureGen2Aquifer`. The description of the possible component's parameters are provided below:

- **aqu_thick** $[m]$ (30 to 90) - thickness of unit (default: 33.2); *linked to Stratigraphy*
- **depth** $[m]$ (100 to 700) - depth to bottom of unit (default: 590.1); *linked to Stratigraphy*
- **por** [-] (0.02 to 0.2) - porosity of unit (default: 0.118)
- **log_permh** $[\log_{10} m^2]$ (-14 to -11) - horizontal permeability (default: -13.39)
- **log_aniso** $[\log_{10}]$ (0 to 3) - anisotropy ratio (default: 0.3)
- **rel_vol_frac_calcite** [-] (0 to 1) - relative volume fraction of calcite in solid phase (default: 0.01).

Component model dynamic inputs:

- **brine_rate** $[kg/s]$ (0 to 31.622) - brine rate
- **brine_mass** $[kg]$ (0 to 6.985e+10) - cumulative brine mass
- **co2_rate** $[kg/s]$ (0 to 31.622) - CO_2 rate
- **co2_mass** $[kg]$ (0 to 6.985e+10) - cumulative CO_2 mass.

Observations from the FutureGen 2.0 Aquifer component are:

- **Pressure_volume** $[m^3]$ - volume of plume where relative change in pressure > 0.065%
- **Pressure_dx** $[m]$ - length of plume where relative change in pressure > 0.065%
- **Pressure_dy** $[m]$ - width of plume where relative change in pressure > 0.065%
- **Pressure_dz** $[m]$ - height of plume where relative change in pressure > 0.065%
- **pH_volume** $[m^3]$ - volume of plume where absolute change in pH > 0.2
- **pH_dx** $[m]$ - length of plume where absolute change in pH > 0.2
- **pH_dy** $[m]$ - width of plume where absolute change in pH > 0.2
- **pH_dz** $[m]$ - height of plume where absolute change in pH > 0.2
- **TDS_volume** $[m^3]$ - volume of plume where relative change in TDS > 10%
- **TDS_dx** $[m]$ - length of plume where relative change in TDS > 10%
- **TDS_dy** $[m]$ - width of plume where relative change in TDS > 10%
- **TDS_dz** $[m]$ - height of plume where relative change in TDS > 10%

- **Dissolved_CO2_volume** $[m^3]$ - volume of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dx** $[m]$ - length of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dy** $[m]$ - width of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dz** $[m]$ - height of plume where relative change in dissolved CO_2 concentration > 20%

3.18 FutureGen2 AZMI Component

The FutureGen 2.0 Above Zone Monitoring Interval (AZMI) component model is a reduced order model that can be used to predict the impact that carbon dioxide (CO_2) and brine leakage from the CO_2 storage reservoir at the FutureGen 2.0 site might have on overlying aquifers or monitoring units. The model predicts the size of "impact plumes" according to five metrics: pH, total dissolved solids (TDS), pressure, dissolved CO_2 and temperature.

The FutureGen 2.0 AZMI model is a regression model fitted to the results of STOMP-CO2E-R multiphase flow and reactive transport simulations of CO_2 and brine leakage using the `py-earth` Python package ([29]). The `py-earth` package is a Python implementation of the Multivariate Adaptive Regression Splines algorithm ([8]), in the style of `scikit-learn` ([27]), a library of machine-learning methods.

The aquifer simulations used to train the FutureGen 2.0 AZMI component model were based on modeling done for monitoring program design at the FutureGen 2.0 site ([32]), as well as porosity and permeability values from the ELAN logs and core samples taken from the characterization well. Nonisothermal simulations were performed for training the AZMI component model.

In the NRAP-Open-IAM control file, the type name for the FutureGen 2.0 AZMI component is `FutureGen2AZMI`. The description of the possible component's parameters are provided below:

- **aqu_thick** $[m]$ (30 to 90) - thickness of unit (default: 33.2); *linked to Stratigraphy*
- **depth** $[m]$ (700 to 1600) - depth to bottom of unit (default: 1043.9); *linked to Stratigraphy*
- **por** [-] (0.02 to 0.2) - porosity of unit (default: 0.118)
- **log_permh** $[\log_{10} m^2]$ (-14 to -11) - horizontal permeability (default: -13.39)
- **log_aniso** $[\log_{10}]$ (0 to 3) - anisotropy ratio (default: 0.3)
- **rel_vol_frac_calcite** [-] (0 to 1) - relative volume fraction of calcite in solid phase (default: 0.01).

Component model dynamic inputs:

- **brine_rate** $[kg/s]$ (0 to 31.622) - brine rate
- **brine_mass** $[kg]$ (0 to 6.985e+10) - cumulative brine mass
- **co2_rate** $[kg/s]$ (0 to 31.622) - CO_2 rate
- **co2_mass** $[kg]$ (0 to 6.985e+10) - cumulative CO_2 mass.

Observations from the FutureGen 2.0 AZMI component are:

- **Pressure_volume** $[m^3]$ - volume of plume where relative change in pressure > 0.065%
- **Pressure_dx** $[m]$ - length of plume where relative change in pressure > 0.065%
- **Pressure_dy** $[m]$ - width of plume where relative change in pressure > 0.065%
- **Pressure_dz** $[m]$ - height of plume where relative change in pressure > 0.065%
- **pH_volume** $[m^3]$ - volume of plume where absolute change in pH > 0.2
- **pH_dx** $[m]$ - length of plume where absolute change in pH > 0.2
- **pH_dy** $[m]$ - width of plume where absolute change in pH > 0.2

- **pH_dz** $[m]$ - height of plume where absolute change in pH > 0.2
- **TDS_volume** $[m^3]$ - volume of plume where relative change in TDS > 10%
- **TDS_dx** $[m]$ - length of plume where relative change in TDS > 10%
- **TDS_dy** $[m]$ - width of plume where relative change in TDS > 10%
- **TDS_dz** $[m]$ - height of plume where relative change in TDS > 10%
- **Dissolved_CO2_volume** $[m^3]$ - volume of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dx** $[m]$ - length of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dy** $[m]$ - width of plume where relative change in dissolved CO_2 concentration > 20%
- **Dissolved_CO2_dz** $[m]$ - height of plume where relative change in dissolved CO_2 concentration > 20%
- **Temperature_volume** $[m^3]$ - volume of plume where relative change in temperature > 0.03%
- **Temperature_dx** $[m]$ - length of plume where relative change in temperature > 0.03%
- **Temperature_dy** $[m]$ - width of plume where relative change in temperature > 0.03%
- **Temperature_dz** $[m]$ - height of plume where relative change in temperature > 0.03%

3.19 Generic Aquifer Component

The Generic Aquifer component model is a surrogate model that can be used to predict the leakage of carbon dioxide (CO_2) and brine from a CO_2 storage reservoir. The model predicts the mass fraction of CO_2 and salt on a 100x10 radial grid surrounding the leaky well and outputs these as gridded observations. The model also predicts the volume and dimensions of aquifer where pore water concentrations exceed specified threshold values of dissolved CO_2 and salt.

The Generic Aquifer model is a machine learning regression model fitted to the results of STOMP-CO2E-R multiphase flow and reactive transport simulations of CO_2 and brine leakage using Tensorflow 2.4. 50,000 nonisothermal multiphase flow simulations were used to train the Generic Aquifer component model. Input parameters were varied using Latin Hypercube Sampling across wide ranges.

In the NRAP-Open-IAM control file, the type name for the Generic Aquifer component is `GenericAquifer`. The description of the component's parameters are provided below:

- **aqu_thick** $[m]$ (25 to 250) - thickness of unit (default: 33.2); *linked to Stratigraphy*
- **top_depth** $[m]$ (100 to 4100) - depth to the top of the aquifer (default: 590.1); *linked to Stratigraphy*
- **por** [-] (0.02 to 0.25) - porosity of unit (default: 0.118)
- **log_permh** $[\log_{10} m^2]$ (-14 to -10) - horizontal permeability (default: -13.39)
- **log_aniso** [-] (0 to 3) - anisotropy ratio Kh/Kv (default: 0.3)
- **aquifer_salinity** [-] (0.0 to 0.015) - salt mass fraction in aquifer water (default: 0.005)
- **reservoir_salinity** [-] (0.015 to 0.05) - salt mass fraction in leak water (default: 0.03)
- **dissolved_salt_threshold** [-] (0.0 to 1.0) - threshold for salt mass fraction (default: 0.02)
- **dissolved_co2_threshold** [-] (0.0 to 1.0) - threshold for CO2 mass fraction (default: 0.01)

Component model dynamic inputs:

- **brine_mass** $[kg]$ (0 to 6.985e+10) - cumulative brine mass
- **co2_mass** $[kg]$ (0 to 6.985e+10) - cumulative CO_2 mass.

Observations from the Generic Aquifer component are:

- **Dissolved_salt_volume** $[m^3]$ - volume of plume where relative change in salt mass fraction > dissolved_salt_threshold
- **Dissolved_salt_dr** $[m]$ - radius of plume where relative change in salt mass fraction > dissolved_salt_threshold
- **Dissolved_salt_dz** $[m]$ - height of plume where relative change in salt mass fraction > dissolved_salt_threshold
- **Dissolved_CO2_volume** $[m^3]$ - volume of plume where dissolved CO_2 mass fraction > dissolved_co2_threshold
- **Dissolved_CO2_dr** $[m]$ - radius of plume where dissolved CO_2 mass fraction > dissolved_co2_threshold
- **Dissolved_CO2_dz** $[m]$ - height of plume where dissolved CO_2 mass fraction > dissolved_co2_threshold

Gridded observations from the Generic Aquifer component are:

- **Dissolved_CO2_mass_fraction** [-] - mass fraction of CO_2 in aquifer pore water on a 100x10 radial grid surrounding the leaky well
- **Dissolved_salt_mass_fraction** [-] - mass fraction of salt in aquifer pore water on a 100x10 radial grid surrounding the leaky well
- **r_coordinate** [m] - radial coordinates of the points in the 100x10 radial grid surrounding the leaky well. The 100 radii are within range from 1.62 m to about 77.5 km.
- **z_coordinate** [m] - depth coordinates of the points in the 100x10 radial grid surrounding the leaky well. The 10 depths used are within the aquifer modeled by the GenericAquifer. The minimum depth is 5% of the aquifer's thickness above the base of the aquifer, while the maximum depth is 95% of the aquifer's thickness above the base of the aquifer. The increment used between depth values is 10% of the aquifer's thickness.

3.20 Atmospheric Model Component

The Atmospheric model is meant to be used for performing scoping studies for CO_2 dispersion after leakage out of the ground. The employed method is an extension of the nomograph approach of Britter and McQuaid (1988) [4] developed for estimating dense gas plume length from a single or multiple leakage sources. The method is very fast and, therefore, amenable to general system-level geologic carbon sequestration (GCS) risk assessment. The method is conservative: it assumes the wind could be from any direction and handles multiple sources by a simple superposition approach [38]. A user's manual for the standalone model is available at [37].

The model is intended to be used for large CO_2 leakage rates (e.g., leakage from an open wellbore). It may not be suitable for very small leakage rate, as, in general, small release rates (e.g., less than 1.0e-5 kg/s) do not form a dense gas release due to ambient mixing. The inputs to the model are leakage rate(s) from leaky well(s), location(s) of leaky well(s), ambient conditions (wind speed), and receptor locations (home or business locations where people are present). The outputs from the model are flags at receptors indicating whether the CO_2 concentration at the location exceeds a pre-defined critical value, and the critical downwind distance from the sources.

Within the control file interface, receptor locations can be specified with the `receptors` keyword argument assigned a full path (including a name) to a csv file containing x- and y-coordinates of the receptors. Alternatively, the `x_receptor` and `y_receptor` keywords can be assigned a list of x- and y-coordinates of the receptors, respectively. In the NRAP-Open-IAM control file, the type name for the Atmospheric model component is `AtmosphericROM`.

Component model input definitions:

- **T_amb** $[°C]$ (5 to 40) - ambient temperature (default: 15)
- **P_amb** [atmosphere] (0.7 to 1.08) - ambient pressure (default: 1)
- **V_wind** $[m/s]$ (1.e-10 to 20) - wind velocity (default: 5)
- **C0_critical** [-] (0.002 to 0.1) - critical concentration (default: 0.01)
- **T_source** $[°C]$ (5 to 50) - released CO_2 temperature (default: 15)

- **x_receptor** $[m]$ - x-coordinate of receptor
- **y_receptor** $[m]$ - y-coordinate of receptor

Possible observations from the Atmospheric Model component are:

- **outflag_r###** [-] - count of critical distances receptor is within from original leak points; here, ### is a receptor number starting at 000
- **num_sources** [-] - number of sources. The possible maximum is a number of leakage points; could be less as leakage sources can potentially coalesce.
- **x_new_s###** $[m]$ - x-coordinate of leakage source; here ### is a source number starting at 000
- **y_new_s###** $[m]$ - y-coordinate of leakage source
- **critical_distance_s###** $[m]$ - critical downwind distance from each source.

3.21 Plume Stability Component

The Plume Stability component model produces quantitative metrics of the area, change in area over time, mobility and spreading [11]. Plume mobility is the effective centroid velocity including the speed and direction of movement. Plume spreading is the effective longitudinal dispersion of the plume along its direction of maximum elongation. This direction is returned by the model as well. The mobility and spreading metrics are comprehensive in that they can effectively handle and account for complex continuous and discontinuous plumes and intra-plume migration. The metrics are calculated using 2D-scalar attribute field values as inputs. The model can read in field values formatted in the NRAP-Open-IAM dataset format. In order to process 3D scalar field data for the model, it is recommended to collapse the 3D data to 2D using a maximization approach as described in [11].

In the NRAP-Open-IAM control file, the type name for the Plume Stability component is `PlumeStability`. The data files providing input for the Plume Stability component need to satisfy the same requirements imposed on the data files used as input for the Lookup Table Reservoir component. In particular, for control file setup of the Plume Stability component the following three keywords have the same meaning:

- `FileDirectory` is a directory where files with the simulation data for the component are located, and which contains files described below;
- `TimeFile` keyword specifies a name of the .csv file that stores the time points (in years) at which the results in the data files are provided;
- `ParametersFilename` keyword contains a name of the .csv file containing the names and values of the parameters used to create the given set of data files; in addition, it lists the names of the .csv files in the folder `FileDirectory` containing simulation data for each of the data file in the set.

The additional keywords of the component's control file interface are:

- `Variables` is a list of observations names provided in the data files and for which some (or all) metrics will be calculated;
- `Thresholds` is a dictionary of pairs (observation name, value) providing threshold value above which the change in the observation value should be taken into account for the calculation of the plume stability metrics.

The only component model input parameter is `index` which indicates the index of the data file from the list in the last column of `ParametersFilename` to be used to produce plume stability metrics. The minimum and maximum value of the parameter is defined by the indices of data files provided in the list.

Possible observations from the Plume Stability component are

- **{obs}_areas** - area of the plume above the predefined threshold
- **{obs}_areas_dt** - change in the area of the plume above the predefined threshold

- **{obs}_mobility** - velocity of centroid of the plume above the predefined threshold
- **{obs}_mobility_angles** - angles/direction at which the centroid of the plume above the predefined threshold is changing
- **{obs}_spreading** - longitudinal dispersion of the plume above the predefined threshold along its direction of maximum elongation
- **{obs}_spreading_angles** - angles/direction at which the dispersion of the plume occurs.

Above, {obs} determines the name of observation for which the plume stability metrics are to be calculated and for which the data is provided in the data files used as input for the component, e.g. **pressure**, **CO2saturation**, etc.

3.22 Chemical Well Sealing Component

The Chemical Well Sealing component is based on the model described in [13]. It predicts whether a fracture at the cement caprock interface, upon exposure to CO_2, would self-seal or not due to calcite precipitation. The model couples two-phase flow of supercritical CO_2 and brine through fractures, advective and diffusive transport along the fracture, diffusive transport within the cement, and chemical reactions between cement and carbonated brine. If the fracture is predicted to self-seal, the time required for sealing is also computed. The original model is described in [34], [36], and [14] and was calibrated using experimental data presented in [34], [36], and [35].

Component model input definitions:

- **fractureAperture** $[m]$ (1.0e-5 to 2.0e-3) - aperture of the fractured leakage path (default: 2.0e-5). Any input aperture that is lower than 10 micron (lower bound) is set to 10 micron.
- **fractureLength** $[m]$ (10.0 to 400.0) - length of the fractured leakage path (default: 20)
- **maxOverpressure** $[Pa]$ (1.0e+6 to 1.5e+7) - maximum overpressure the base of the fracture is expected to experience (default: 5.0e+6).

The output from the Chemical Well Sealing component informs about the sealing ability of the fractured leakage pathway. In case the fracture seals, the component would also report sealing time.

- **seal_flag** [-] - flag informing whether a fracture would seal (1) or not (0) due to calcite precipitation
- **seal_time** $[s]$ - predicted time for sealing a fracture by calcite recipitation. If fracture doesn't seal this variable is set to 0.0.

CHAPTER

FOUR

COMPONENT COMPARISON

This section of the documentation provides the comparison tables for the wellbore component models available in NRAP-Open-IAM.

Table 4.1 provides comparison of the models parameters.

Table 4.1: Comparison of wellbore components in NRAP-Open-IAM

Input variable	Cemented wellbore ROM	Multisegmented wellbore ROM	Open wellbore ROM
logWellPerm (well Permeability)	-13.95 to -10.1	-17 to -9	Open tubing or casing
logAquPerm or logThiefPerm (aquifer or thief zone permeability)	-13.995 to -12	-14 to -9	log of aquifer and reservoir transmissivity: -11.27 to -8.39
wellRadius (radius of the leaky well)	0.025 to 0.25	0.01 to 0.5	0.025 to 0.25
Brine and CO_2 properties	Included in the ROM simulations	Calculated as a function of depth. Pressure gradient: 9792 Pa/m. Temperature gradient: 25 C/km.	Mass fraction of salt: 0 to 0.2. Temperature gradient: 25 C/km. Temperature at surface: 15 degrees Celcius.
Number of shale layers	3	3 to 30	1
reservoirDepth and wellDepth (depth of the reservoir, well)	960 to 3200	No limitation	1000 to 4000
shaleThickness, aquiferThickness, reservoirThickness (thickness of shale, aquifer or reservoir)	reservoir: 51.2 m, thief zone: 22.4 m, aquifer: 29.2 m, upper caprock layer: 11.2 m	1 to 1600 m	Well Top: 0 to 500 m
brineResSaturation (residual brine saturation)	0.001	Used as tuning parameter to allow CO_2 to accumulate in intermediate aquifers	Not needed
deltaP (increase in pressure due to injection)	0.1 to 9.3 MPa	No limitations	0 to 20 MPa
Energy transport consideration	Nonisothermal	Isothermal	Nonisothermal
Governing wellbore equation employed	Two phase Darcy equation	Two phase Darcy equation	Drift flux
Type of ROM	numerical	analytical	numerical

continues on next page

Table 4.1 – continued from previous page

Input variable	Cemented wellbore ROM	Multisegmented wellbore ROM	Open wellbore ROM
Reference	D. R. Harp, R. Pawar, J. W. Carey, C. W. Gable, Reduced order models of transient CO_2 and brine leakage along abandoned wellbores from geologic carbon sequestration reservoirs, Int. J. Greenhouse Gas Control, 45 (2016), pp. 150-162	S. Baek, D. H. Bacon, N. J. Huerta, NRAP-Open-IAM Multisegmented Wellbore Reduced-Order Model, 2021, PNNL-32364	L. H. Pan, S. W. Webb, C. M. Oldenburg, Analytical solution for two-phase flow in a wellbore using the drift-flux model. Adv. Water Resour. 34 (2011), pp. 1656-1665

Table 4.2 describes limitations of each model.

Table 4.2: Limitations of the wellbore components in NRAP-Open-IAM

Cemented wellbore ROM	Multisegmented wellbore ROM	Open wellbore ROM
Limited flexibility in specifying layer thicknesses	Deviations from full physics simulations due to analytical nature of the model	The ROM was derived from short-time simulations. Simulations were terminated after 1,000-time steps or 100 hours, whichever came first. It should not be used for long term predictions if leakage rate is high and would result in significant mass loss from reservoir.
Limited flexibility in specifying the number of thief zones (only 1 thief zone supported)	Large shale thicknesses, "Large wellbore segment length for shale (or cap rock) layers may lead to errors due to the nature of the analytical approach employed. Less than 100 m for each segment is recommended with the current model. Although a particular combination of model parameters can relax the degree of the error or enlarge the applicable length of the segment, the quality of the performance is not guaranteed with large segment lengths."	

continues on next page

Table 4.2 – continued from previous page

Cemented wellbore ROM	Multisegmented wellbore ROM	Open wellbore ROM
Limitations on total reservoir pressure change. Reservoir pressure changes greater than 9.3 MPa are outside the bounds of the model. Pressure changes in saline aquifers are typically below this range. However, pressure changes can be larger than this range in other formations that are more limited in extent (e.g., depleted gas fields)	No leakage of CO_2 in intermediate aquifers, "The analytical model does not consider the lateral leakage into intermediate aquifers, resulting in zero accumulation of CO_2 in intermediate aquifers. All the CO_2 is transported to the topmost aquifer, yielding an overestimation of the amount of CO_2 leaking into the topmost aquifer. Residual brine saturation can be used as a proxy for lateral leakage into intermediate aquifers. Users can calibrate this parameter to yield a desirable CO_2 leakage rate/mass into the thief layers (intermediate aquifers). The parameter would have to be recalibrated if any of the other input parameters, like reservoir pressure, well permeability, etc., are changed."	
Limitations of the first and second derivative of pressure and saturation changes		

CHAPTER FIVE

USE CASES

In the folder *examples/Control_Files* there are a number of example control files distributed with the NRAP-Open-IAM tool. The description of the files is provided in the Table 5.1.

Table 5.1: Control file examples distributed with NRAP-Open-IAM

File name	Included components	Comments
ControlFile_ex1a.yaml	Simple reservoir, cemented wellbore	Forward simulation. The saturation and pressure output produced by simple reservoir model is used as input for cemented wellbore model. The example produces three TimeSeries type of plots. The first two plots is of the CO_2 leakage rates into the intermediate aquifer (aquifer 1) and the shallow aquifer (aquifer 2). The third plot is of the pressure in the reservoir at the wellbore locations.
ControlFile_ex1b.yaml	Simple reservoir, cemented wellbore	Forward simulation. The saturation and pressure output produced by simple reservoir model is used as input for cemented wellbore model. The example produces several TimeSeries type of plots. In addition, this example demonstrates the use of the `UseMarkers`, `UseLines`, and `VaryLineStyles` entries for TimeSeries plots.
ControlFile_ex2.yaml	Simple reservoir, multisegmented wellbore	Latin hypercube sampling, 30 realizations. A fixed seed can be specified for the sampling setup: symbol # before the seed has commented it out. This example shows a way to specify wellbore locations inside the wellbore component specification with the `Locations` keyword.
ControlFile_ex3.yaml	Simple reservoir, multisegmented wellbore, carbonate aquifer	Latin hypercube sampling, 30 realizations. Example illustrates use of known and random wellbore locations.
ControlFile_ex4a.yaml	Simple reservoir, open wellbore, carbonate aquifer	Latin hypercube sampling, 30 realizations. Example demonstrates several plotting options: `TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats`. For analysis of the large number of realizations a user might find it more helpful to plot the time series statistics rather than the data for each realization using `TimeSeriesStats` plotting option. `TimeSeriesAndStats` plots the simulated values and the related statistics. The `subplot` keyword can be used to present some of the results in subplots.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex4b.yaml	Simple reservoir, open wellbore	Latin hypercube sampling, 30 realizations. Example demonstrates several plotting options: `TimeSeries`, `TimeSeriesStats`, and `TimeSeriesAndStats`. For analysis of the large number of realizations a user might find it more helpful to plot the time series statistics rather than the data for each realization using `TimeSeriesStats` plotting option. `TimeSeriesAndStats` plots the simulated values and the related statistics. The `subplot` keyword can be used to present some of the results in subplots. Example also illustrates an option to setup open wellbore with an option to calculate the critical pressure.
ControlFile_ex5.yaml	Simple reservoir, cemented wellbore	Example is used to demonstrate parameter study analysis.
ControlFile_ex6.yaml	Lookup table based reservoir	Latin hypercube sampling, 10 realizations. To run this example, the additional *Kimberlina* data set have to be downloaded and unzipped into the *source/components/reservoir/lookuptables/Kimb_54_sims* folder. The data files can be downloaded from the same EDX directory where the NRAP-Open-IAM was downloaded or from https://gitlab.com/NRAP/Kimberlina_data. The data set comes from simulation work done by Wainwright et. al. [33].
ControlFile_ex7a.yaml	Multisegmented wellbore	Latin hypercube sampling, 30 realizations. Example illustrates the use of dynamic input to drive a wellbore model without attaching a reservoir model. This functionality can be used to quickly evaluate behavior of a single component model with a fixed input or interactions between several component models without constructing a full systems model. In the example the dynamic input is provided as a list of pressure and CO_2 saturation data at the simulation time points.
ControlFile_ex7b.yaml	Multisegmented wellbore	Forward simulation. Example illustrates the use of dynamic input to drive a wellbore model without attaching a reservoir model. In the example the dynamic input is provided as a path to the file containing pressure and CO_2 saturation data at the simulation time points.
ControlFile_ex8a.yaml	Simple reservoir, multisegmented wellbore	Latin hypercube sampling, 300 realizations. Example demonstrates the use of the `Analysis` section to compute correlation and sensitivity coefficients and create visualizations of the analysis results. Each subsection of the `Analysis` section shows some of the available options for the user to control.
ControlFile_ex8b.yaml	Simple reservoir, multisegmented wellbore	Latin hypercube sampling, 300 realizations. Example demonstrates the use of the `Analysis` section to compute correlation and sensitivity coefficients and create visualizations of the analysis results. Each subsection of the `Analysis` section shows some of the available options for the user to control.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex8c.yaml	Simple reservoir, open wellbore, FutureGen2 aquifer	Latin hypercube sampling, 100 realizations. Example demonstrates the use of the `Analysis` section to compute correlation and sensitivity coefficients and create visualizations of the analysis results. Each subsection of the `Analysis` section shows some of the available options for the user to control.
ControlFile_ex9a.yaml	Simple reservoir, open wellbore, atmospheric ROM	Latin hypercube sampling, 30 realizations. The AtmosphericROM model simulates CO_2 dispersion in the atmosphere as a dense gas leak. Example illustrates two plotting options available only for the AtmosphericROM model. `AtmPlumeSingle` keyword can be used to plot the critical distance from leakage points for a single realization. For multiple Monte-Carlo simulations probability of being within a critical distance can be estimated and plotted using `AtmPlumeEnsemble` keyword. Both types of plots produce a map-view of the release area for each time step.
ControlFile_ex9b.yaml	Simple reservoir, open wellbore, atmospheric ROM	Forward simulation. The AtmosphericROM model simulates CO_2 dispersion in the atmosphere as a dense gas leak. Example illustrates `AtmPlumeSingle` plotting option.
ControlFile_ex9c.yaml	Lookup table based reservoir, open wellbore, atmospheric ROM	Latin hypercube sampling, 30 realizations. The AtmosphericROM model simulates CO_2 dispersion in the atmosphere as a dense gas leak. Example illustrates two plotting options available only for the AtmosphericROM model. `AtmPlumeSingle` keyword can be used to plot the critical distance from leakage points for a single realization. For multiple Monte-Carlo simulations probability of being within a critical distance can be estimated and plotted using `AtmPlumeEnsemble` keyword. Both types of plots produce a map-view of the release area for each time step.
ControlFile_ex10.yaml	Lookup table based reservoir, cemented wellbore, deep alluvium aquifer	Latin hypercube sampling, 30 realizations. Example illustrates a setup connecting lookup table based reservoir, wellbore and aquifer impact components.
ControlFile_ex11.yaml	Simple reservoir, multisegmented wellbore, carbonate aquifer	Forward simulation. The saturation/pressure output produced by simple reservoir model is used to drive leakage from five multisegmented wellbore models separated into two groups according to their properties. Two carbonate aquifer components are linked to both groups of wellbores and estimate the impact from the leakage of CO_2 and brine into the aquifers 1 and 2.
ControlFile_ex12.yaml	Generalized flow rate	Forward simulation. Example simulates brine and CO_2 leakage rates to the deepest aquifer (aquifer 1) illustrating the use of the generalized flow rate component. The component does not require linking to other components.
ControlFile_ex13.yaml	Generalized flow rate, carbonate aquifer	Forward simulation. Example computes the leakage rates of brine and CO_2 to aquifers 1 and 2 utilizing the generalized flow rate component, and calculates the associated impact with the help of carbonate aquifer component.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex14.yaml	Lookup table based reservoir, multisegmented wellbore, FutureGen2 aquifer	Latin hypercube sampling, 50 realizations. Example illustrates a setup connecting lookup table based reservoir, multisegmented wellbore and aquifer impact components.
ControlFile_ex15.yaml	Lookup table based reservoir, multisegmented wellbore, FutureGen2 AZMI	Latin hypercube sampling, 100 realizations. Example illustrates a setup connecting lookup table based reservoir, multisegmented wellbore and aquifer impact components.
ControlFile_ex16.yaml	Plume stability	Example is used to demonstrate parameter study analysis for plume stability component.
ControlFile_ex17.yaml	Fault flow	Example demonstrates forward simulation for fault flow component. Dynamic inputs are provided as arrays.
ControlFile_ex18.yaml	Lookup table based reservoir, fault flow	Latin hypercube sampling, 10 realizations. To run this example, the additional *Kimberlina* data set have to be downloaded and unzipped into the *source/components/reservoir/lookuptables/Kimb_54_sims* folder. The data files can be downloaded from the same EDX directory where the NRAP-Open-IAM was downloaded or from https://gitlab.com/NRAP/Kimberlina_data. The data set comes from simulation work done by Wainwright et. al. [33].
ControlFile_ex19.yaml	Lookup table based reservoir, seal horizon	Latin hypercube sampling, 5 realizations. To run this example, the additional *Kimberlina* data set have to be downloaded and unzipped into the *source/components/reservoir/lookuptables/Kimb_54_sims* folder. The data files can be downloaded from the same EDX directory where the NRAP-Open-IAM was downloaded or from https://gitlab.com/NRAP/Kimberlina_data. The data set comes from simulation work done by Wainwright et. al. [33].
ControlFile_ex20.yaml	Analytical reservoir	Latin hypercube sampling, 10 realizations. Example illustrates use of analytical reservoir component.
ControlFile_ex21.yaml	Multisegmented wellbore, deep alluvium aquifer (ML)	Forward simulation. Example illustrates use of deep alluvium aquifer component (ML). Dynamic input provided through input files is used to drive wellbore component.
ControlFile_ex22.yaml	Chemical well sealing component	Forward simulation. Example illustrates setup of the chemical well sealing component.
ControlFile_ex23.yaml	Lookup table based reservoir, seal horizon	Forward simulation. To run this example, the additional *Kimberlina* data set have to be downloaded and unzipped into the *source/components/reservoir/lookuptables/Kimb_54_sims* folder. The data files can be downloaded from the same EDX directory where the NRAP-Open-IAM was downloaded or from https://gitlab.com/NRAP/Kimberlina_data. The data set comes from simulation work done by Wainwright et. al. [33].

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex24.yaml	Lookup table based reservoir, multisegmented wellbore, generic aquifer	Forward simulation. Example illustrates a setup connecting lookup table based reservoir, multisegmented wellbore and aquifer impact components. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex25.yaml	Multisegmented wellbore, generic aquifer	Latin hypercube sampling, 10 realizations. Example illustrates the use of dynamic input to drive a wellbore component providing input for aquifer impact component.
ControlFile_ex26.yaml	Lookup table based reservoir, multisegmented wellbore, FutureGen2 aquifer	Forward simulation. Example illustrates a setup connecting lookup table based reservoir linked to 3d data to multisegmented wellbore and aquifer impact components. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex27.yaml	Lookup table based reservoir, multisegmented wellbore	Forward simulation. Example illustrates a setup connecting lookup table based reservoir and multisegmented wellbore components. Reservoir component is linked to lookup tables with additional temperature metric.
ControlFile_ex28.yaml	Lookup table based reservoir, multisegmented wellbore	Forward simulation. Example illustrates a setup connecting lookup table based reservoir and multisegmented wellbore components. Reservoir component is linked to lookup tables with additional data metric.
ControlFile_ex29.yaml	Lookup table based reservoir, multisegmented wellbore	Forward simulation. Example illustrates use of known and random wellbore locations in 3d domain. The pressure and CO2 saturation are obtained from a reservoir component linked to 3d lookup tables.
ControlFile_ex30.yaml	Lookup table based reservoir, multisegmented wellbore	Forward simulation. Example illustrates different options of setting locations both for lookup table reservoir and multisegmented wellbore components: through direct entry, through file, as equally spaced array points, and as a grid. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex31a.yaml	Simple reservoir, open wellbore, FutureGen2 aquifer	Forward simulation. Example demonstrates the setup of the AoR plot as well as an option for user to control which and in what form the model outputs are saved.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex31b.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Example demonstrates the setup of the AoR plot as well as an option for user to control which and in what form the model outputs are saved.
ControlFile_ex31c.yaml	Simple reservoir, open wellbore, FutureGen2 aquifer	Latin hypercube sampling, 70 realizations. Example demonstrates the setup of the AoR plot.
ControlFile_ex31d.yaml	Simple reservoir, open wellbore, FutureGen2 aquifer	Latin hypercube sampling, 70 realizations. Example demonstrates the setup of the AoR plot with time list option.
ControlFile_ex32a.yaml	Lookup table based reservoir, open wellbore, FutureGen2 aquifer	Forward simulation. Locations for wellbore components are provided through an input file. Example demonstrates the use of the AoR plot as well as an option for user to control which and in what form the model outputs are saved. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex32b.yaml	Lookup table based reservoir, open wellbore, FutureGen2 aquifer	Forward simulation. Locations for wellbore components are provided through an input file. Stratigraphy is setup as spatially varying. Example demonstrates the use of the AoR and stratigraphy plot types as well as an option for user to control which and in what form the model outputs are saved. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex32c.yaml	Lookup table based reservoir, open wellbore, FutureGen2 aquifer	Forward simulation. Locations for wellbore components are provided through an input file. Stratigraphy is setup as spatially varying. Example demonstrates the use of the AoR and stratigraphy plot types. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex33a.yaml	Simple reservoir, open wellbore	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Setup of stratigraphy and stratigraphic column plots is illustrated.
ControlFile_ex33b.yaml	Simple reservoir, open wellbore	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Setup of stratigraphy and stratigraphic column plots is illustrated.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex34.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Setup of default stratigraphy plot is illustrated.
ControlFile_ex35.yaml	Simple reservoir, open wellbore	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Setup of stratigraphy plot is illustrated.
ControlFile_ex36.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Locations are setup with grid option. Setup of stratigraphy plot is illustrated.
ControlFile_ex37.yaml	Lookup table based reservoir, multisegmented wellbore	Forward simulation. Setup of stratigraphy plot is illustrated. This example requires the additional *FutureGen 2.0* data set that can be downloaded from the following source: https://edx.netl.doe.gov/dataset/futuregen-2-0-1008-simulation-reservoir-lookup-table. The data set has to be placed into the *source/components/reservoir/lookuptables/FutureGen2/1008_sims* folder.
ControlFile_ex38.yaml	Simple reservoir, open wellbore	Forward simulation. Stratigraphy is setup as spatially varying. Parameters of wellbore component are defined as linked to the stratigraphy explicitly. Setup of stratigraphy plot is illustrated.
ControlFile_ex39a.yaml	Simple reservoir, multisegmented wellbore, FutureGen2 AZMI	Latin hypercube sampling, 30 realizations. Example demonstrates the setup of the TTFD plot.
ControlFile_ex39b.yaml	Simple reservoir, multisegmented wellbore, FutureGen2 AZMI	Forward simulation. Example demonstrates the setup of the TTFD and stratigraphy plots.
ControlFile_ex40.yaml	Lookup table based reservoir, multisegmented wellbore, FutureGen2 AZMI	Latin hypercube sampling, 30 realizations. Example demonstrates the setup of the TTFD plot.
ControlFile_ex41.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Example demonstrates the setup of the TTFD plot.
ControlFile_ex42.yaml	Simple reservoir, multisegmented wellbore, carbonate aquifer	Latin hypercube sampling, 30 realizations. Example demonstrates the setup of the TTFD plot.
ControlFile_ex43.yaml	Simple reservoir, multisegmented wellbore, deep alluvium aquifer	Latin hypercube sampling, 30 realizations. Example demonstrates the setup of the TTFD plot.
ControlFile_ex44a.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined with the array of values.
ControlFile_ex44b.yaml	Theis reservoir	Latin hypercube sampling, 30 realizations. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined with the array of values.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex45a.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined with the array of values.
ControlFile_ex45b.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined through input files.
ControlFile_ex45c.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined through input files.
ControlFile_ex45d.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. Injection rates and times are defined through the same input file.
ControlFile_ex46a.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. The reservoir component is setup with multiple injection wells. Injection rates and times are defined with multidimensional arrays.
ControlFile_ex46b.yaml	Theis reservoir	Forward simulation. Example demonstrates the setup of simulation with Theis reservoir component. The reservoir component is setup with multiple injection wells. Injection rates and times are defined through input files.
ControlFile_ex47.yaml	Generic reservoir	Forward simulation. Example illustrates setup of generic reservoir component.
ControlFile_ex48.yaml	Fault flow	Forward simulation. Example illustrates setup of fault flow component. Dynamic inputs are provided as arrays.
ControlFile_ex49.yaml	Fault flow	Forward simulation. Example illustrates setup of fault flow component. Dynamic inputs are provided as arrays.
ControlFile_ex50.yaml	Simple reservoir, fault flow	Latin hypercube sampling, 30 realizations. Example demonstrates the application of fault flow component. Input to the component is provided from the linked reservoir component.
ControlFile_ex51a.yaml	Analytical reservoir, cemented wellbore (wr)	Forward simulation. Example illustrates setup of system model with analytical reservoir component providing required input for cemented wellbore (wr) component. Example also illustrates use of known and random wellbore locations.
ControlFile_ex51b.yaml	Analytical reservoir, cemented wellbore (wr)	Latin hypercube sampling, 32 realizations. Example illustrates setup of system model with analytical reservoir component providing required input for cemented wellbore (wr) component. Example also illustrates use of only random wellbore locations.
ControlFile_ex51c.yaml	Analytical reservoir, cemented wellbore (wr)	Forward simulation. Example illustrates setup of system model with analytical reservoir component providing required input for cemented wellbore (wr) component. Example illustrates setup of cemented wellbore components simulating leakage to different aquifer layers.
ControlFile_ex52a.yaml	Hydrocarbon leakage	Forward simulation. Example illustrates setup of system model with hydrocarbon leakage component.

continues on next page

Table 5.1 – continued from previous page

File name	Included components	Comments
ControlFile_ex52b.yaml	Hydrocarbon leakage	Latin hypercube sampling, 30 realizations. Example illustrates setup of system model with hydrocarbon leakage component.
ControlFile_ex53a.yaml	Fault leakage	Forward simulation. Example illustrates setup of system model with fault leakage component.
ControlFile_ex53b.yaml	Fault leakage	Latin hypercube sampling, 25 realizations. Example illustrates setup of system model with fault leakage component.
ControlFile_ex54a.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Example illustrates setup of gridded radial metric plot.
ControlFile_ex54b.yaml	Simple reservoir, open wellbore, generic aquifer	Forward simulation. Example illustrates setup of gridded radial metric plot with default settings.
ControlFile_ex54c.yaml	Lookup table based reservoir, multisegmented wellbore, generic aquifer	Forward simulation. Example illustrates setup of gridded radial metric plot.
ControlFile_ex54d.yaml	Simple reservoir, open wellbore, generic aquifer	Latin hypercube sampling, 30 realizations. Example illustrates setup of gridded radial metric plot.

Beyond control files the described scenarios can be implemented with a help of a Python script. Example scripts can be found in the *examples/scripts* folder.

BIBLIOGRAPHY

[1] D. H Bacon, D. I. Demirkanli, and S. K. White. Probabilistic risk-based area of review (aor) determination for a deep-saline carbon storage site. *International Journal of Greenhouse Gas Control*, 102:103153, 2020.

[2] D. H. Bacon, Z. Dai, and L. Zheng. Geochemical impacts of carbon dioxide, brine, trace metal and organic leakage into an unconfined, oxidizing limestone aquifer. *Energy Procedia*, 63:4684–4707, 2014. doi:10.1016/j.egypro.2014.11.502.

[3] S. Baek, D. Bacon, and N Huerta. NRAP-Open-IAM Analytical Reservoir Model. Development and Testing. Technical Report, Pacific Northwest National Laboratory, Richland, Washington, 2021. PNNL-31418.

[4] R. E. Britter and J. D. McQuaid. *Workbook on the dispersion of dense gases*. Contract Research Report 17 Health and Safety Executive, Sheffield, UK, 1988.

[5] M. A. Celia, J. M. Nordbotten, S. Bachu, M. Dobossy, and B. Court. Risk of leakage versus depth of injection in geological storage. *Energy Procedia*, 1(1):2573–2580, 2009. Greenhouse Gas Control Technologies 9 Proceedings of the 9th International Conference on Greenhouse Gas Control Technologies (GHGT-9), 16-20 November 2008, Washington DC, USA. doi:10.1016/j.egypro.2009.02.022.

[6] M. A. Celia, J. M. Nordbotten, B. Court, M. Dobossy, and S. Bachu. Field-scale application of a semi-analytical model for estimation of CO2 and brine leakage along old wells. *International Journal of Greenhouse Gas Control*, 5(2):257–269, 2011.

[7] Z. Dai, E. Keating, D. Bacon, H. Viswanathan, P. Stauffer, A. Jordan, and R. Pawar. Probabilistic evaluation of shallow groundwater resources at a hypothetical carbon sequestration site. *Scientific Reports*, 4:4006, 2014. URL: http://www.ncbi.nlm.nih.gov/pubmed/24844225, doi:10.1038/srep04006.

[8] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[9] D. R. Harp. Model analysis toolkit (MATK). URL: http://matk.lanl.gov.

[10] D. R. Harp, R. Pawar, J. W. Carey, and C. W. Gable. Reduced order models of transient CO2 and brine leakage along abandoned wellbores from geologic carbon sequestration reservoirs. *International Journal of Greenhouse Gas Control*, 45:150–162, 2016. URL: http://www.sciencedirect.com/science/article/pii/S1750583615301493, doi:10.1016/j.ijggc.2015.12.001.

[11] Dylan Harp, Tsubasa Onishi, Shaoping Chu, Bailian Chen, and Rajesh Pawar. Development of quantitative metrics of plume migration at geologic CO2 storage sites. *Greenhouse Gases: Science and Technology*, 9(4):687–702, 2019. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/ghg.1903.

[12] J. Herman and W. Usher. SALib: an open-source python library for sensitivitiy analysis. *Journal of Open Source Software*, 2(9):97, 2017. URL: https://github.com/SALib/SALib, doi:10.21105/joss.00097.

[13] Jaisree Iyer, Xiao Chen, and Susan A. Carroll. Impact of chemical and mechanical processes on leakage from damaged wells in CO2 storage sites. *Environmental Science & Technology*, 54(2):1196–1203, 2020. doi:10.1021/acs.est.9b05039.

[14] Jaisree Iyer, Stuart D.C. Walsh, Yue Hao, and Susan A. Carroll. Incorporating reaction-rate dependence in reaction-front models of wellbore-cement/carbonated-brine systems. *International Journal of Greenhouse Gas Control*, 59:160–171, 2017. doi:https://doi.org/10.1016/j.ijggc.2017.01.019.

[15] A. B. Jordan, P. H. Stauffer, D. Harp, J. W. Carey, and R. J. Pawar. A response surface model to predict CO2 and brine leakage along cemented wellbores. *International Journal of Greenhouse Gas Control*, 33:27–39, 2015. doi:10.1016/j.ijggc.2014.12.002.

[16] E. Keating, D. Bacon, S. Carroll, K. Mansoor, Y. Sun, L. Zheng, D. Harp, and Z. Dai. Applicability of aquifer impact models to support decisions at CO2 sequestration sites. *International Journal of Greenhouse Gas Control*, 52:319–330, 2016. doi:10.1016/j.ijggc.2016.07.001.

[17] E. H. Keating, D. H. Harp, Z. Dai, and R. J. Pawar. Reduced order models for assessing CO2 impacts in shallow unconfined aquifers. *International Journal of Greenhouse Gas Control*, 46:187–196, 2016. doi:10.1016/j.ijggc.2016.01.008.

[18] G. V. Last, C. J. Murray, and Y. Bott. Derivation of groundwater threshold values for analysis of impacts predicted at potential carbon sequestration sites. *International Journal of Greenhouse Gas Control*, 49:138–148, 2016. doi:10.1016/j.ijggc.2016.03.004.

[19] E. Lindner. NSealR—A User's Guide, third-generation. Technical Report, National Energy Technology Laboratory, Morgantown, WV, 2015.

[20] K. Mansoor, T. A. Buscheck, X. Yang, S. A. Carroll, and X. Chen. LLNL Kimberlina 1.2 NUFT Simulations, June 2018. URL: https://edx.netl.doe.gov/dataset/llnl-kimberlina-1-2-nuft-simulations-june-2018.

[21] K. Mansoor, Y. Sun, and Carroll S. Development of a general form CO2 and brine flux input model. NRAP Technical Report Series, Lawrence Livermore National Laboratory, 2014.

[22] J. M. Nordbotten and M. A. Celia. *Geological storage of CO2: modeling approaches for large-scale simulation*. John Wiley & Sons, 2011. doi:10.1002/9781118137086.

[23] J. M. Nordbotten, M. A. Celia, S. Bachu, and H. K. Dahle. Semianalytical solution for CO2 leakage through an abandoned well. *Environmental Science & Technology*, 39(2):602–611, 2005. doi:10.1021/es035338i.

[24] J. M. Nordbotten, D. Kavetski, M. A. Celia, and S. Bachu. Model for CO2 leakage including multiple geological layers and multiple leaky wells. *Environmental Science & Technology*, 43(3):743–749, 2009. doi:10.1021/es801135v.

[25] L. H. Pan and C. M. Oldenburg. T2Well - an integrated wellbore-reservoir simulator. *Computers & Geosciences*, 65:46–55, 2014. doi:10.1016/j.cageo.2013.06.005.

[26] L. H. Pan, S. W. Webb, and C. M. Oldenburg. Analytical solution for two-phase flow in a wellbore using the drift-flux model. *Advances in Water Resources*, 34(12):1656–1665, 2011. doi:10.1016/j.advwatres.2011.08.009.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] E. Plischke. An effective algorithm for computing global sensitivity indices (EASI). *Reliability Engineering & System Safety*, 95(4):354–360, 2010. URL: http://www.sciencedirect.com/science/article/pii/S0951832009002579, doi:10.1016/j.ress.2009.11.005.

[29] J. Rudy. Py-earth. 2013. URL: https://contrib.scikit-learn.org/py-earth.

[30] I.M. Sobol' and S. Kucherenko. Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation*, 79(10):3009–3017, 2009. URL: http://www.sciencedirect.com/science/article/pii/S0378475409000354, doi:10.1016/j.matcom.2009.01.023.

[31] J.-Y. Tissot and C. Prieur. Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliability Engineering & System Safety*, 107:205–213, 2012. SAMO 2010. URL: http://www.sciencedirect.com/science/article/pii/S0951832012001159, doi:10.1016/j.ress.2012.06.010.

[32] Vince R Vermeul, James E Amonette, Chris E Strickland, Mark D Williams, and Alain Bonneville. An overview of the monitoring program design for the FutureGen 2.0 CO2 storage site. *International Journal of Greenhouse Gas Control*, 51:193–206, 2016.

[33] H. Wainwright, S. Finsterle, Q. Zhou, and J. Birkholzer. Modeling the performance of large-scale CO2 storage systems: a comparison of different sensitivity analysis methods. Technical Report, U.S. Department of Energy, National Energy Technology Laboratory, Morgantown, WV, 2012.

[34] Stuart D.C. Walsh, Wyatt L. Du Frane, Harris E. Mason, and Susan A. Carroll. Permeability of wellbore-cement fractures following degradation by carbonated brine. *Rock Mechanics and Rock Engineering*, 46(3):455–464, 2013. doi:https://doi.org/10.1007/s00603-012-0336-9.

[35] Stuart D.C. Walsh, Harris E. Mason, Wyatt L. Du Frane, and Susan A. Carroll. Experimental calibration of a numerical model describing the alteration of cement/caprock interfaces by carbonated brine. *International Journal of Greenhouse Gas Control*, 22:176–188, 2014. doi:https://doi.org/10.1016/j.ijggc.2014.01.004.

[36] Stuart D.C. Walsh, Harris E. Mason, Wyatt L. Du Frane, and Susan A. Carroll. Mechanical and hydraulic coupling in cement–caprock interfaces exposed to carbonated brine. *International Journal of Greenhouse Gas Control*, 25:109–120, 2014. doi:https://doi.org/10.1016/j.ijggc.2014.04.001.

[37] Y. Zhang and C. Oldenburg. *Multiple Source Leakage Reduced-Order Model (MSLR) Tool User's Manual, Version: 2016.11-1.0.1*. U.S. Department of Energy, National Energy Technology Laboratory, Morgantown, WV, 2016. NRAP Technical Report Series NRAP-TRS-III-016-2016. doi:10.18141/1592689.

[38] Y. Zhang, C. M. Oldenburg, and L. Pan. Fast estimation of dense gas dispersion from multiple continuous CO2 surface leakage sources for risk assessment. *International Journal of Greenhouse Gas Control*, 49:323–329, 2016. URL: http://www.sciencedirect.com/science/article/pii/S1750583616301074, doi:10.1016/j.ijggc.2016.03.002.

[39] G. Zyvoloski. FEHM: a control volume finite element code for simulating subsurface multi-phase multi-fluid heat and mass transfer. Technical Report, Los Alamos National Laboratory, Los Alamos, NM, 2007. URL: https://fehm.lanl.gov/pdfs/FEHM_LAUR-07-3359.pdf.

PYTHON MODULE INDEX

O

openiam.AnalyticalReservoir, 43
openiam.AtmosphericROM, 65
openiam.CarbonateAquifer, 59
openiam.CementedWellbore, 47
openiam.CementedWellboreWR, 48
openiam.ChemicalWellSealing, 67
openiam.DeepAlluviumAquifer, 60
openiam.FaultFlow, 55
openiam.FaultLeakage, 58
openiam.FutureGen2Aquifer, 62
openiam.FutureGen2AZMI, 63
openiam.GeneralizedFlowRate, 50
openiam.GenericAquifer, 64
openiam.GenericReservoir, 45
openiam.HydrocarbonLeakage, 51
openiam.LookupTableReservoir, 44
openiam.MultisegmentedWellbore, 46
openiam.OpenWellbore, 49
openiam.PlumeStability, 66
openiam.SealHorizon, 52
openiam.SimpleReservoir, 42
openiam.Stratigraphy, 41