IORSim Script User Guide

Contents

1 Introduction

The main motivation for IORSim was to add geochemistry based IOR methods to field simulation workflows without altering the reservoir simulator. To achieve this, the communication between IORSim and the reservoir simulator operate via interface-files. The current version of IORSim is developed to cooperate with Eclipse100. However, with some modifications, other reservoir simulators that allow communication via files can cooperate with IORSim.

IORSim is executed in forward or backward mode. In forward mode, the Eclipse run completes before IORSim reads the Eclipse output and performs the calculations. In backward mode, Eclipse is paused after every timestep, and IORSim then continue by reading the timestep output and prepare an interface-file for the next Eclipse timestep with updated keywords (such as SATNUM). IORSim is then paused and Eclipse resume to read the updated interface-file.

A python script is developed to handle the tasks of pausing and resuming Eclipse and IORSim at the right time, checking that output files a properly flushed, and preparing interface-files. The script can be executed in graphics mode (GUI), on the command line (terminal), or imported and used by other python scripts. The capabilities of the script include

- Run IORSim in backward, forward, and single mode
- Convert and merge IORSim and Eclipse restart files to make the results viewable in ResInsight
- Check IORSim input keywords
- Progress-bar with estimated remaining simulation time

The GUI version of the script includes these additional features

- Plot Eclipse and IORSim well-data, with an option to compare two cases
- Edit Eclipse and IORSim input files
- View log files
- Integrated searchable IORSim user guide

A running version of Eclipse and IORSim is required by the script, and the script is compatible with Linux and Windows. The latest version of the script can be downloaded from github.com/janlv/IORSim_GUI/releases/latest.

1.1 The GUI window

The input variables of the script are controlled by the Run-, Case- and Days-fields of the toolbar at the top of the GUI window. The Run-field is a dropdown-menu with four different running modes: Forward, Backward, Eclipse, and IORSim. The Case-field is a dropdown-menu of all imported cases, and the total simulation time is entered in the Days-field. A simulation is started by clicking the green triangle. The red square allows the user to stop the simulation before the total simulation time is reached. The Compare-field is a dropdown-menu, similar to the Case-field, that allows the user to compare plots from two different cases.

The main view of the GUI is the plot window and the Eclipse and IORSim plotting menus. This allows the user to follow the evolution of selected well-data during the simulation. A maximum of two plots can be view simultaneously. If more than two Y-axis boxes or more than two Wells boxes are selected, the last selected box is automatically unselected.

The Edit-menu opens an editor for Eclipse or IORSim input files with syntax highlighting and a search function. The View-menu opens a log file viewer showing the terminal output of Eclipse or IORSim runs, or the progress of the python script. The editor and log viewer are displayed in the plot window. The Plot option at the top of the View-menu brings back the default plot view.

The Edit-menu also gives access to a Settings window where the most important entry is the location of the executable IORSim program. This needs to be specified before running the first simulation. The remaining options are described in section 1.1.5.

The Help-menu opens a searchable IORSim user guide in a separate window

A status-bar showing the progress of the simulation and the estimated remaining time is located at the bottom of the GUI window.

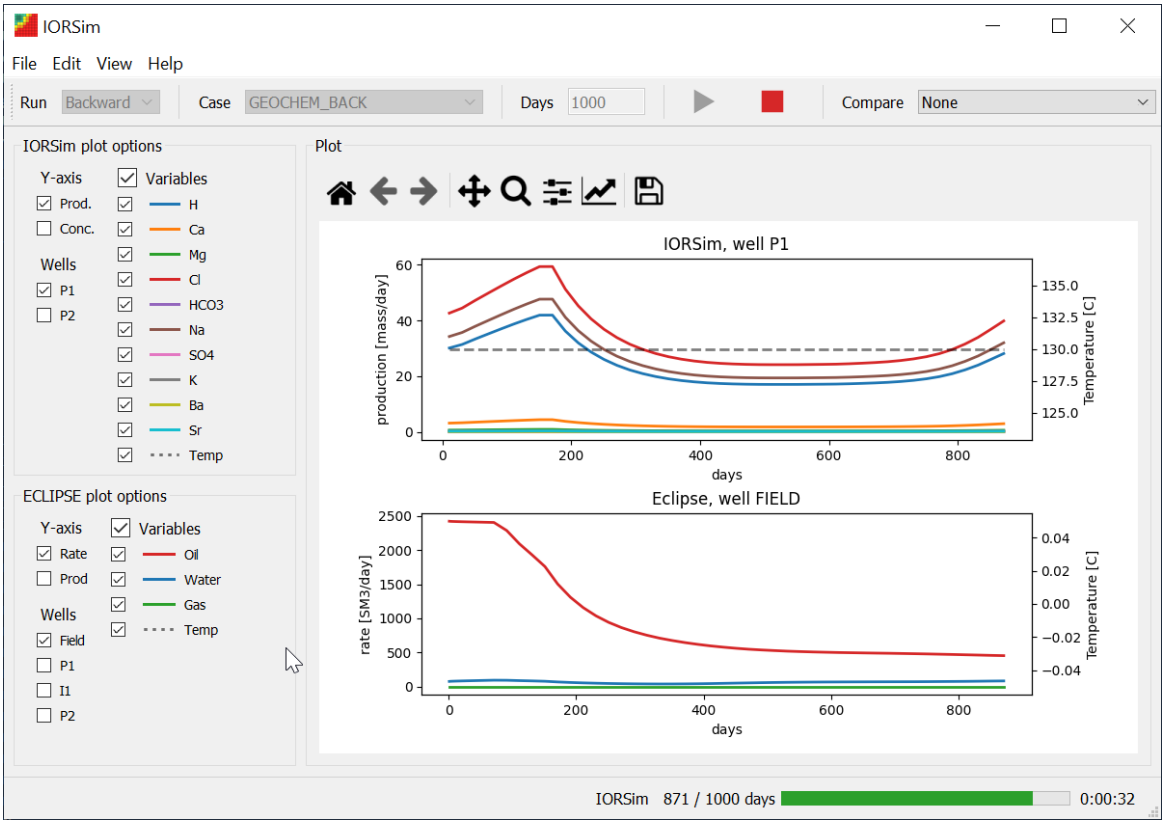

Figure 1

1.2 Running modes

Four different running modes are implemented and available from the Run dropdown-list: 1) Forward, 2) Backward, 3) Eclipse, and 4) IORSim

In Forward mode, Eclipse is first executed from start to finish to create restart (UNRST) and well-data (RFT) files. Then, IORSim reads the output from Eclipse and add tracer data, geochemical concentration values, or other relevant values in an Eclipse formatted restart file (FUNRST).

In Backward mode, Eclipse executes only one timestep before IORSim reads the Eclipse output and updates the flow variables. This allows IORSim to update the SATNUM values used by Eclipse in the next timestep. During a backward run, the Eclipse output files (UNRST and RFT) need to be checked to ensure they are properly flushed before Eclipse is paused and IORSim can resume.

The Eclipse and IORSim running modes simply execute Eclipse and IORSim as stand-alone applications. The IORSim mode can only be executed if Eclipse output files are present in the case-folder.

1.3 Case files

A minimal IORSim case consists of an Eclipse input file (.DATA-file) and an IORSim input file (.trcinp-file). If the geochemical option is used a chemistry-file must be included using the *CHEMFILE keyword in the IORSim input file. The Eclipse input file may also include different additional files. The case-files for a specific case are organized in separate case-folders under

'IORSim_cases' in the running directory. The path and name of the main case directory can be changed from the Settings window (Edit -> Settings).

For a DATA-file named Case.DATA the following files are copied to the folder CASE if they exist

- The Eclipse input file, Case.DATA
- The IORSim input file, Case.trcinp
- All files ending with .inc, .dat, .grdecl, .egrid, .vfp, or .sch
- All files included by the *CHEMFILE keyword in the trcinp-file

Note that the filename match is case-insensitive, also if executed under Linux.

1.4 Import and run an existing case

Import an existing case using File -> Import case… to bring up a file-navigation window where you can locate the .DATA-file of the case you want to import. After import, the name of the new case will appear in capital letters in the Case dropdown-menu in the toolbar. The running mode is selected from the Run dropdown-menu. A backward run is automatically selected if the READDATA keyword is present in the DATA-file. A Forward run is the default for all other cases. For forward-cases, Eclipse and IORSim can also be run in single mode by choosing Eclipse or IORSim from the Run menu.

For a backward case, the duration of the simulation is given in the Days field of the toolbar. For a forward-case, the total simulation time, given by the TSTEP keyword in the DATA-file, is displayed in the Days field without the option to edit it directly. However, the Eclipse input file can be opened for editing via the Edit menu, and the simulation time modified by searching for TSTEP and changing the value.

The first time the script is used you need to give the location of the IORSim executable under Edit -> Settings. Clicking the green go-button will also bring up the Settings-window shown in Figure 2 if the IORSim program is missing. Locate the IORSim executable using the Open-button.

If Eclipse is properly installed, Eclipse can be executed by the default eclrun macro. If this should fail, the full path of the eclrun macro must be given in Settings. The remaining Setting-options are discussed later.

Start the simulation by clicking the green triangle in the toolbar. The progress-bar at the bottom shows the estimated remaining simulation time and displays the current timestep of the simulation.

By default, the view area shows evolving plots of the well-data from both Eclipse and IORSim output-files. Which values to plot are selected in the IORSim and Eclipse plotting menus to the left of the view area. The View menu also gives access to log-files showing the printed output from Eclipse, IORSim or the python script.

1.5 Settings

The settings window is available at the bottom of the Edit-menu, or via the shortcut Ctrl+s.

1.5.1 IORSim program

The location of the executable IORSim program is the only setting that require input from the user before running the first simulation.

1.5.2 Eclipse program

The default value 'eclrun' require that the Path environment variable includes the full path of the ecl/macros-folder. Otherwise, the full path of the eclrun executable must be specified using the Open button.

1.5.3 Main case-folder location

All input and output files related to a given case are organized in a case folder with the same name as the input-files. These case folders are again organized in a main case-folder named *IORSim_cases* located in the folder the script is executed from. The name and location of the main case-folder can be changed by locating a new folder via the *Change* button.

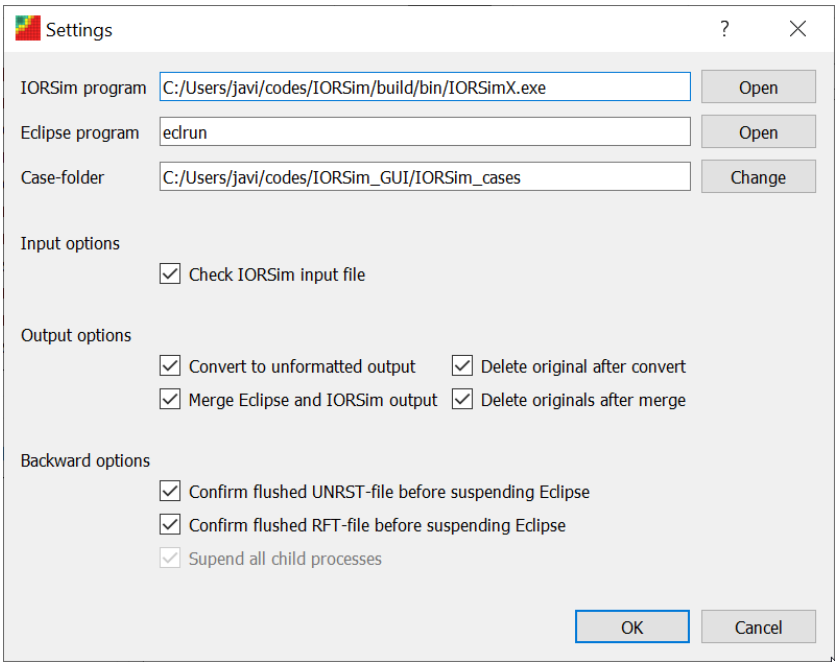

Figure 2: The settings window of the IORSim GUI script

1.5.4 Input options

If this option is ticked off, a simple check of the keywords in the IORSim input file is performed before the simulation starts. This will give a more useful feedback than if the IORSim application stops due to missing keywords or keywords in the wrong order.

1.5.5 Output options

IORSim outputs reservoir data in an ASCII restart file (with the extension FUNRST) that follow the Eclipse format. The data can be viewed in the FloViz program that is part of the Eclipse installation. However, some prefer to use ResInsight for reservoir visualization, but ResInsight can only import binary (unformatted) Eclipse restart files. In addition, Eclipse and IORSim data

are saved in separate files which makes it a bit cumbersome to get all the reservoir data available in the same view.

By checking the convert and merge boxes, the script first converts the ASCII output to binary output and then merge the original Eclipse and converted IORSim restart file into one unified restart file. By default, the original files are deleted after a successful convert and merge to save disk space. Uncheck the two delete-boxes to keep the original restart files.

1.5.6 Backward options

During a backward run it is important to check that the Eclipse restart file (the UNRST file) and the well-data file (the RFT file) are completely flushed before Eclipse is paused. Disabling these checks might speed up the simulation, but it will make backward runs less stable and prone to stop prematurely.

The option 'Suspend all child processes' means that all sub-processes of a running processes are suspended when the main process is suspended. This option is always on and unable to uncheck since it improves the stability of backward runs. However, the option can be disabled by giving the -alive_children argument to the command line version of the script (see Section 1.6)

```
usage: IORSim_GUI.py [-h] [-eclexe ECLEXE] [-iorexe IOREXE] [-no_unrst_check] [-no_rft_check] [-rft_size] [-iorsim]
                     [-eclipse] [-v V] [-keep_files] [-to_screen] [-only_convert] [-only_merge] [-delete]
                     [-alive_children] [-check_input]
                     root days

positional arguments:
  root              Eclipse case folder or full path of the DATA-file
  days              Simulation time interval

optional arguments:
  -h, --help        show this help message and exit
  -eclexe ECLEXE    Name of excecutable, default is 'eclrun'
  -iorexe IOREXE    Name of IORSim executable, default is 'IORSimX'
  -no_unrst_check   Backward mode: do not check flushed UNRST-file
  -no_rft_check     Backward mode: do not check flushed RFT-file
  -rft_size         Backward mode: Only check size of RFT-file, default is full check
  -iorsim           Run only iorsim
  -eclipse          Run only eclipse
  -v V              Verbosity level, higher number increase verbosity, default is 3
  -keep_files       Interface-files are not deleted after completion
  -to_screen        Print program log to screen
  -only_convert     Only convert+merge and exit
  -only_merge       Only merge and exit
  -delete           Delete obsolete output files after convert and merge has finished
  -alive_children   Only stop parent-processes (approx. 5% faster, but might be more unstable)
  -check_input      Check IORSim input file keywords
```

Figure 3 Description of the command line arguments of the IORSim python script. This information is displayed if -h is given as an argument to the script

1.6 Using the script on the command line (non-GUI version)

The script can also be executed on the command line by providing a path to a DATA-file and the simulation duration in days. The different arguments are listed and explained in Figure 3. It is recommended to run the GUI version of the script first and locate the IORSim executable in the Settings window to avoid specifying the executable on the command line every time the script is used.

1.7 Using the script as a python module

The script can also be imported as a python module and used in other python scripts or Jupyter notebooks. Figure 4 shows a python script that use the IORSim script to loop over a

```python
from ior2ecl import runsim
from pathlib import Path

def mypath(dirs):
    return Path.home().joinpath(*(dirs.split()))

def loop_run(cases, times):
    iorexe = mypath('codes IORSim build bin IORSimX.exe')
    cdir = mypath('codes IORSim_GUI GUI cases')
    n = 0
    for case,time in zip(cases, times):
        print(f'CASE {n} : {case}\n--------------------------')
        runsim(root=cdir/case/case, time=time, iorexe=iorexe)
        n += 1

if __name__ == '__main__':
    cases = ['L18', 'KURS-07A', 'SNORRE-IORSIM-OLD', 'GEOCHEM_BACK', 'GEOCHEM_FWD']
    times = [1100 , 2000      , 3700               , 1000          , 1000]
    loop_run(cases, times)
```

Figure 4: Possible python implementation that use the IORSim script to loop over a list of case-folders