An Open Source Coupling Algorithm With Applications Towards CO_2 Storage Modelling

Dimier Alain

Eifer: European Institute for Energy Research
Karlsruhe institute of technology
Karlsruhe D-76131, Germany
E-mail alain.dimier@eifer.org

Keywords

CO_2 storage, hydrology, geochemistry, flow, transport modelling, multiphysics, phreeqC, Elmer, finite element analysis, Python, object oriented programming

Abstract

We present here the way to bring together two open source modellers, PhreeqC and Elmer, with disjunctive application fields. They are gathered in a single environment to enable multi-component, reactive solute transport studies in three-dimensional saturated/ unsaturated ground-water flow systems. The choice of Python as programming environment is explained, the way the tools become so called Python modules is presented. Based on Open Source, it enables the study of natural and contaminated ground-water flow systems at a variety of scales ranging from laboratory experiments to local and regional field scales.
Two illustrative examples are given. The first one is considering dissolution/precipitation phenomena in the near field of a geothermal injection well. The second study is considering at a laboratory scale the interaction of a CO_2 brine with a Calcite plug, that one is directly issued from the CO_2 application field.

Introduction

At the end of the twentieth century, members of the scientific community warned for greenhouse gas effects on the global climate warming. Climate models enabled to establish with a good confidence that one of its major source was the raising of greenhouse gases emissions. So, after water vapour, CO_2 was pointed out as one of the main contributors of that global warming; carbon dioxide concentrations rising from 280 ppmv in 1850 to about 389 ppmv today. The main factor for these rising concentrations has been identified as being the fossil fuels combustion. Depending on the scenarios, economic growth, effects of political decisions on CO_2 emissions, the global climate warming ranges from 1.4 °C to more than 5 °C over the 21st century.
One of the main studied solutions to that problem is the so called Capture and Carbone Storage, C.C.S., namely the geological storage of "industry-induced" CO_2 in depleted oil reservoirs or brine bearing aquifers. Carbon dioxide has to be initially captured directly from industrial sources via post-combustion capture, precombustion capture, or combustion of fossil fuels in a pure oxygen environment. Once captured, the CO_2 will be pumped down wells into a host formation within a supercritical state. Porous layers should provide storage place while some layers, like clay ones, should provide the sealing of the CO_2 storage reservoir.
Apart from its cost, the key question of the storage process is the trapping duration: "how long CO_2 will be trapped underground"? At a first glance, expertise gained through the petroleum industry enables to predict that such storages should be safe. But potential leakages events risks exist and research studies have to be conducted to better analyze all trapping mechanisms like CO_2 dissolution in brine, capillary retention, adsorption and mineral trapping; but also in the mean time, all potential dissolution processes induced by the acidification of the underground medium through the supercritical CO_2 plume evolution.
In order to investigate these phenomena, associated to the acquisition of an experimental CO_2-brine-rock interactions test bench, Eifer has decided to investigate numerical modelling on that field, so called T-H-M-C modelling. The association of modelling and experiments should enable a better understanding of involved processes, their characterization and evaluation enabling to determine how the subsurface will perform as a storage container, but also estimate the potential risk of leakage at abandoned wells or natural fractures within the media.
One way to efficiently implement a

Proceedings 25th European Conference on Modelling and Simulation ©ECMS Tadeusz Burczynski, Joanna Kolodziej Aleksander Byrski, Marco Carvalho (Editors)
ISBN: 978-0-9564944-2-9 / ISBN: 978-0-9564944-3-6 (CD)

multidimensional numerical tool to model such a phenomenology is to couple already available open source tools. That can be made using the operator splitting methodology eventually combined with a sequential iterative approach, see [9] and [10]. Operator splitting enables to have a modular approach, already available tools becoming part of a new tool giving access to the broader targeted phenomenology.

Tool Choice

Considering the evolution over time of a CO_2 storage, the geochemistry of the reservoir is one of the main mechanisms to be evaluated. Looking at available geochemical software, phreeqC is probably the most widely used geochemical model. Developed by the USGS [4][5] User's guide to phreeqC (version 2), it enables to simulate chemical reactions in natural or polluted water; cf. [5]. The tool is based on equilibrium chemistry of aqueous solutions interacting with minerals, gases, solid solutions, exchangers and sorption-surfaces. It also enables to handle kinetically controlled reactions. The choice of phreeqC was primarily determined, among potential open source software's, by its ability to make a mass balance on involved gaseous components, that point being mandatory when considering unsaturated hydro-geochemical modelling. It is to note that, within the coupling, we only employ its batch reactions capacities.
To deal with flow, ion transport, temperature and mechanics, we have made the choice of Elmer. Developed by CSC-IT [7], a Finnish institute, it is a soft based on finite element technologies; written mainly in fortran90, it also uses C and C++. Elmer, in its 6.1 version, is distributed under the GNU license (GPL 2.). The use of Elmer gives access to up to date algebraic solvers. For direct methods, Lapack or Umfpack libraries are made accessible; and for iterative methods, preconditioned Krylov subspace or, multilevel methods are made available. It can also be run parallel using the MPI tool. It uses domain decomposition to distribute the load to multiple processes that are being run either on different cores or CPU's. Here, mesh partitioning can be made using Metis, see [2].

Coupling Methodology

To setup the coupling between phreeqC, as geochemical tool, and Elmer as multi-physic modeller, we need to use a high-level programming language as a glue to tie components together to enrich the physics to be handled. An interpreted language, like Python or Ruby, enabling code readability and maintainability appeared as the best potential solution. A Comparison of these languages can be found under [6]. Being familiar with Python, we decided to stay on that choice.
As already mentioned, Python is an interpreted interactive, object-oriented programming language. It provides high-level data structures such as list and dictionaries (associative arrays), dynamic typing and dynamic binding, modules, classes, exceptions, automatic memory management, etc… Moreover, the speed at which a working code can be generated, the lack of a time-consuming compile cycle, the resulting code being easy to read and to modify for any user without much effort, are some of the elements that argue in favour of Python. Python is also free and offers a broad and dynamic community of users. As a consequence, associated to Python, one find a large number of modules that can be imported, these ones providing useful embedded numerical methods for scientific computing. We use mainly one of these modules, ScientificPython [8].
Apart from Python as a tool manager, we also use wxPython, [14], to provide a functional graphical user interface.
It is worth mentioning that using Python, the resulting tool is portable on any linux system and probably on windows without requiring a lot of time investment.

Python as an integration tool

The scripting framework being defined, the first step was to setup a data model covering the physics to be handled. As an example, for chemistry, to match phreeqC requirements, a generic species class has to be created, see [10]; then a masterspecies and a secondaryspecies class will inherit from that ancestor to enable part of the chemical problem definition.
The use of such an object-oriented programming enables to define a case study through class instantiation and module parametrisation. In that way, the user gains more insight in its studies due to the setup of comprehensive data / object bindings.
The script containing all data being read through the interpreter, we generate in that way the two specific data files of Elmer and phreeqC; the tool being launched within Python with two specific shared objets, WElmer.so and WPhreeqc.so.
At a first glance, the specific syntax of their respective data files could be ignored by the user, but at least for geochemistry, initial equilibriums can, and should, be checked analysing phreeqC outputs before explicitly launching a coupled process.
At the opposite, in some cases, it can become the duty of the user to make "add ons" to the Python data model to handle within the coupling some specific parameters of the software.

A last point worth to mention, associated to Python, is the ability for the user to setup "user defined" functionalities within the coupling. The user has just to have a good knowledge of the data model and some insight in the Python programming to introduce its own physical models within the coupling algorithm, this without modifying the initial standard version.
As an example, a function enabling to drive permeability as a function of porosity variations over time can be introduced without modifying the coupling algorithm.
We will now analyse the way functionalities of Elmer and phreeqC can be made part of Python modules.

Wrapping process

Looking at the chemical transport algorithm, the main requirement is to "wrap" the legacy codes in Python, these ones becoming Python extensions modules, so called shared objects on Linux (analogous to DLLs on Windows). The wrapping code process could be to some extent automated via the use of SWIG [15], the Simple Wrapper Interface Generator. Nevertheless, most of the necessary methods being developed from scratch, we decided to avoid SWIG to improve the readability of that wrapping. That way, we have to create C wrapping functions; these functions will enable data manipulation between the tool itself and the Python interpreter. These functions have as basis the specific structures of each tool.
The first method to be created concerns the initialisation. Once the specific data files have been created via Python, this method will enable to launch the soft, read the data file and generate the suitable structures. As an example, for Elmer, we have a function bounded to initialisation which looks like:

```
PyObject * py_elmer_initialize (PyObject *self,
PyObject * args)
  {
  extern unknownAnz, activeCellsAnz;
  int ok,ierror;
  ierror = 0;

  ok = !c_elmer_initialize(ierror);

  if(!ok){
    PyErr_SetString(PyExc_RuntimeError,"Error in
Elmer initialize");
    return NULL;}

  return Py_BuildValue("");
```

Here PyErr_SetString and Py_BuildValue ensure communication between C and Python. c_elmer_initialise is a C function enabling to wrap the fortran subroutine ensuring Elmer initialisation. The elmer_initialise.f90 subroutine is part of the ElmerSolver.f90 standard one. The time stepping is managed by Python, Elmer methods being called sequentially over time stepping for each managed phenomenology to be involved, eventually with various time steppings.
PhreeqC is managed in a similar manner. Exchange of concentration fields are made at a memory level through set and get methods. We won't detail here all necessary methods for the coupling algorithm.

Mesh generation

As already mentioned, associated to the finite element code, we needed to choose a mesh generator. The open source requirement leaded us to the choice of Gmsh [11]. We just had to define a Python module to manipulate mesh elements, bodies,…, retrieved from the mesh file. Via the Python interface, mesh bodies become objects which can be associated to material properties or aqueous states characterising the problem to be modelled. The Elmer data file being generated in that way, the correspondence between bodies, materials and initial conditions is managed through Python.
Gmsh presents some restrictions about the geometries to be handled. Elmer being interfaced with major commercial mesh generators, depending on user's needs, meshes from different sources can be used.

Postprocessing

As a postprocessor, apart from ASCII files, we use gnuplot [12] and paraview [13], the "legacy" and "xml" file format of vtk being handled through Python modules.

Equations

As already mentioned, the core of studies on CO_2 storage is the geochemical-transport process. It combines chemical reactions and transport processes, fluid/rock interactions being the two binding phenomena. Considering the monophasic geochemistry/ transport coupling as an example, we have to consider the following transport equations:

$$\frac{\partial \omega C_i}{\partial t} = -\nabla(\omega v C_i) + \nabla.(\omega D \nabla C_i) - \frac{\partial q_i}{\partial t} \quad (1)$$

C_i stands for mobile concentrations of species i, v for the seepage velocity, D is the hydrodynamic dispersion tensor, $\frac{\partial q_i}{\partial t}$ represents the concentration change in the solid phase due to fluid/solid interactions and ω the porosity of the medium. Porosity can vary over time, these variations being induced by precipitation/ dissolution phenomena.
The seepage velocity is related to the Darcy velocity, U, which can be obtained from the Darcy law:

$$U = -\frac{k}{\mu}\nabla(p + \rho g z) \quad (2)$$

k being the permeability of the medium and μ the dynamic viscosity of the fluid.
Transport equations (1) can also be expressed as:

$$\frac{\partial \omega C_i}{\partial t} = L(C_i) - \frac{\partial q_i}{\partial t} \quad (3)$$

L represents here the advection-diffusion-dispersion operator.
The model being defined, various ways to solve it numerically exist. A detailed analysis of these methods can be found in the work of Yeh and Tripathi [10].The first historical way uses operator splitting [9]. In such an algorithm, the transport equations being linear, we have to solve N_a algebraic systems, N_a being the number of aqueous master species of the problem, each of these systems having N_v unknowns, N_y being the number of vertices in the mesh. The aqueous master unknown's formulation enables to limit the dimension of the algebraic system to be solved.
The chemical part is made of a non-linear speciation problem on each node with the aqueous master species as unknowns, the chemical speciation enabling to determine the repartition between species.
It is to notice that one direct constraint of that formulation is that aqueous master species have to share the same diffusion coefficient.
The main interest of the operator-splitting approach is that the method is easy to implement using different codes for chemistry and transport, the coupling algorithm allowing a direct and efficient parallelisation of chemistry. The main drawback is the occurrence of "operator-splitting errors", see [9], these one can be partially avoided through iterative splitting algorithms. Dependant from the unknowns of the problem, for the iterative algorithm, three formulations can be used. Here we will use the "CC" formulation which is based on the standard formulation of advection-diffusion species transport equations, cf. (1) .
The "CC" algorithm relies on a fixed point, Picard like, algorithm:

$$\begin{aligned} \frac{C_i^{n+1,k} - C_i^n}{\Delta t} &= L(C_i^{n+1,k}) - R_i^{k-1} \\ \frac{C_i^{n+1,k+1} - C_i^{n,k}}{\Delta t} &= f(C_i^{n,k}) \end{aligned} \quad (4)$$

where R_i^{k-1} stands for source terms issued from aqueous/solid interactions, see Fig. 4.
For temperature, we have the following equation:

$$\frac{\partial(\alpha_h T)}{\partial t} + \frac{\partial(\beta_h V_i T)}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\lambda^{cond_T}\frac{\partial T}{\partial x_j}\right) + S_T \quad (5)$$

where $\lambda^{cond_T} = (1-\omega)\lambda^s + \omega\lambda^f$ is the thermal conductivity, α_k and β_k depend on specific heat coefficients:

$$\begin{aligned} \alpha_h &= (1-\omega)\rho^s c^s + \omega\rho^f c^f \\ \beta_k &= \rho^f c^f \end{aligned} \quad (6)$$

Validation and applications

The tool has been validated over numerous analytical and practical cases issued from the literature and respective involved software's libraries.
To illustrate the application field, we give here two examples.
The first one is issued from the geothermal industry. Considering a hot reservoir, we simulate here the evolution of anhydrite over time in the surrounding of an injection well. The initial reservoir temperature is of 100°C while the injected brine has a temperature of 20°C. Due to the high brine salinity (1.6 mol/l), the Pitzer database is used. The temperature controls the anhydrite solubility, anhydrite being more soluble in cold waters. In the simulation, at the injection level and 2.5m away from that one, we see, Fig. 1, the evolution of anhydrite over time. Initially slightly increasing, the dissolution process occurs over about 0.2 days.
The second case considered here is a synthetic benchmark to demonstrate the ability to model porosity evolution induced by the percolation of a Calcite plug through a CO_2 brine. Porosity evolution is modelled by considering the following invariant:

$$\omega(1 + f_{mv}) = 1 \quad (7),$$

where:

$$f_{mv} = \sum_{i=1} V_{m_i} p_i$$

p_i (mol/m^3) represents here the concentration of mineral i, and V_{m_i} (m^3/mol) its molar volume; the summation being made over minerals present in the system.
The test case is supposed to be axisymmetric. The 3D dimensional plug is made of Calcite and Quartz. A kinetic law has been associated to Calcite dissolution with a standard parametrisation issued from the standard phreeqC geochemical data file.
The Darcy velocity is constant over time. The initial porosity field presents a radial dependence, see Fig. 2. Here, due to dissolution phenomena, the porosity field is affected over time, see Fig. 3.

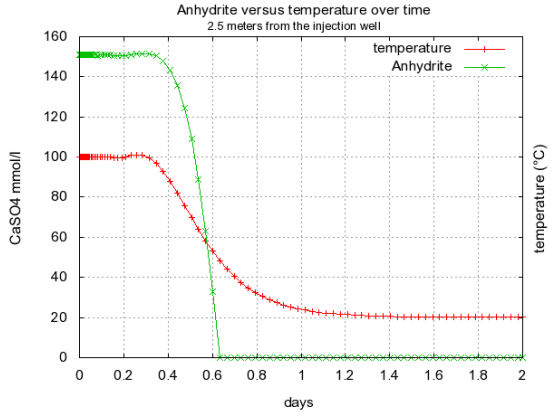

Fig. 1 Evolution over time of the anhydrite concentration near the injection well

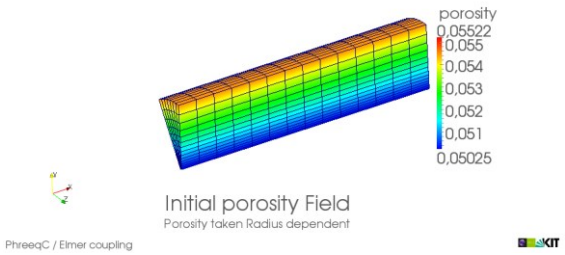

Fig. 2 Initial, radially dependant, porosity distribution field

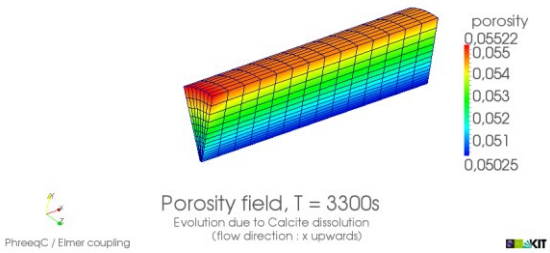

Fig. 3 Porosity field evolution, elapsed time: T=3300s

Further developments

Considering the actual state of the platform, it covers a wide spectrum of the saturated geochemical flow-transport phenomena and additionally the so called Richards flow model in the unsaturated field. As already mentioned, a temperature field can also be studied and, via user's functions, a mechanical stress analysis can be pursued. However, apart from Elmer, considering mechanics, no specific software has been introduced; Elmer being probably, within that frame, the easiest way to handle mechanics.
The next step in that direction should be the modelling of geothermal fields where thermal stresses and fluid pressure in the fractures can vary when considering operating or production variations; these are likely to induce in turn a change of the aperture and conductivity of the fractures due to precipitation/dissolution processes. To model these effects, a mechanical approach is required for allowing determining the deformations of the fractured medium under hydro-thermo-mechanical loading.
Apart from mechanics, the thorough validation of the treatment of the vadose zone is actually under development.
PhreeqC enables a mass balance on gases and seems to be suitable to handle a fully coupled two-phase flow which could be applied to CO_2 storage studies.

REFERENCES

[1] Elmer Model Manual CSC IT Centre for Science 2010

[2] Extending and Embedding the Python Interpreter, Release 2.5.2 Guido van Rossum, 2008

[3] http://www.csc.fi/english

[4] http://www.usgs.gov

[5] User's guide to phreeqC (version 2)
A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations

[6] http://wiki.Python.org/moin/LanguageComparisons

[7] http://www.csc.fi/english

[8] http://dirac.cnrs-orleans.fr/plone/software

[9] On the construction and comparison of difference schemes, SIAM J. Numerical Analysis, vol. 5, 1968, p. 506-517.
Strang G.

[10] Critical evaluation of recent developments in hydrogeochemical transport models of reactive multi-chemical components
Water Resources Research 25, 1989, 93.108
Yeh, G.T., Tripathi, V.A

[11] http://www.geuz.org/gmsh

[12] http://www.gnuplot.info

[13] http://www.paraview.org

[14] http://sourceforge.net/projects/wxPython

[15] http://www.swig.org

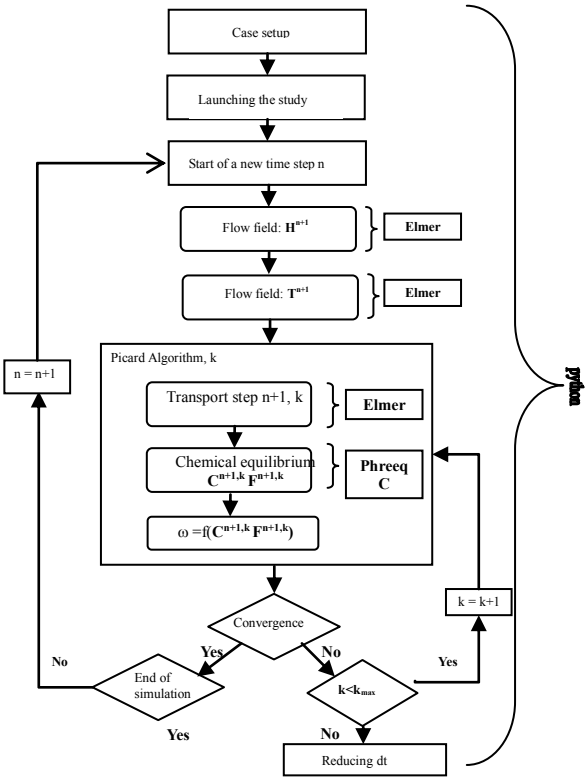

Fig. 4 Schematic flow chart of the operator splitting algorithm

AUTHOR BIOGRAPHY

After a PhD. in numerical analysis at the university of Lyon in France, **Dimier A.** worked first at the D.L.R in Germany in aerodynamics for a couple of years. Then after 9 years as a researcher in the chemical industry he moved to the nuclear industry in the field of multiphysics analysis. Now, working as senior scientist in the Karlsruhe University at the Eifer institute, he is working in the field of CCS modelling.