EUROVOLC

European Network of Observatories and Research Infrastructure for Volcanology

Deliverable Report

D10.3 Modeling of volcano geothermal systems

Work Package:	*Integration and modelling of geophysical, geochemical and petrological monitoring data*
Work Package Number:	*WP10*
Work Package Leader:	*UNILEEDS*
Task (Activity) Name:	*Integrated modeling of volcano geothermal systems*
Task Number:	*10.1.3*
Responsible Activity Leader:	*INGV*
Lead Beneficiary:	*INGV*
Author(s):	*Gilda Currenti and Santina Chiara Stissi*
Type of Deliverable:	*Report*
Dissemination level:	*Public*

Programme: H2020
Project number: 731070

SUMMARY

We present a hydro-geophysical model to compute surface ground deformation and gravity changes in a volcanic edifice caused by variations in fluid temperature, pore pressure and density due to perturbation of a geothermal system. The hot fluid circulation in the volcano is simulated by the well-known MUFITS software, which provides the temperature, pressure and density changes and gas saturation, then used by the thermo-poro-elasticity model to compute the surface ground deformation and gravity changes. Two numerical methods have been investigated for solving thermo-poro-elasticity equations: (i) a numerical finite element method, implemented within the COMSOL environment; (ii) a semi-analytical method useful for fast computation in simplified geometries. Comparison between the two methods shows results in good agreement.

Codes have been implemented as MATLAB scripts for the automation of the calculation and the pre- and post-processing of input data and solutions.

1. INTRODUCTION

The continuous monitoring of geophysical observables and their quantitative assessment through mathematical models play an important role in advancing the comprehension of volcano geothermal system. These models, reproducing the evolution in space and in time of the ground deformation and gravity changes, compared with the observed data allow to quantify the underlying processes and provide hints about possible scenarios in the volcanic edifice (Currenti and Napoli 2017; Currenti et al. 2017; Del Negro et al. 2013). Ground deformation and gravity changes in fact represent the superficial manifestations of the perturbation of the volcano geothermal system at depth (Troiano et al. 2011). Particularly, in a volcano geothermal system, the fluid circulation causes temperature and pore pressure changes, which in turn engender fluid density, stress and strain changes. Stress and strain variations are mirrored in ground deformation, while density variations are reflected in gravity field changes (Currenti and Napoli 2017; Currenti et al. 2017). Usually, these geophysical observables are modelled in terms of volume or pressure sources embedded in an elastic medium (Juncu et al. 2019; Drouin et al. 2017) using standard analytical solutions (Mogi 1958; Okada 1985; Yang et al. 1988). Indeed, this approach appears at odds with the complex processes in which hot fluids from magmatic origin interact with hydrothermal system producing heating and pressurization. A quantitative evaluation of this interaction is fundamental for a correct assessment of the geophysical observations and their interpretation in volcanic geothermal areas. A thermo-poro-elastic numerical model is here proposed to jointly evaluate ground deformation and gravity changes caused by hydrothermal fluid circulation in complex media.

To evaluate numerically how the temperature, pore pressure and density changes affect the geophysical observables, we implemented a **hydro-geophysical model** (Figure 1), which exploit the use of the MUFITS (Multiphase Filtration Transport Simulator) *reservoir simulation model* (Afanasyev 2012; Afanasyev 2013a, b) and *thermo-poroelasticity model*:

1. The MUFITS **reservoir simulation software** simulates the circulation of a non-isothermal multiphase fluid of two components (H_2O and CO_2) in a porous medium. The software solves, by using a finite volume discretization, the mass and energy balance equations in the temperature and pressure range 0-1000 °C and 0-350 MPa, respectively. It provides the solutions for gas saturation and temperature, pore pressure and density changes, which are used as input values in the thermo-poroelasticity model.

2. The **thermo-poroelasticity model** is implemented to evaluate the geophysical observables, in particular the ground deformation and gravity changes. We compared two codes. The first one uses the finite element COMSOL Multiphysics software (COMSOL 2012) for solving the elastostatic equation and to evaluate ground deformation and the Poisson equation to evaluate gravity changes. The second code enables to fast compute the observables using semi-analytical solutions for simple homogeneous half-space models.

The implemented hydro-geophysical code (Stissi et al. 2021) provides the opportunity to explore different model configurations that cannot be taken into account using standard analytical models. Since the physics of the modelled hydrothermal system is similar to any geothermal system, the approach is generally applicable to study geothermal areas, such as Krafla.

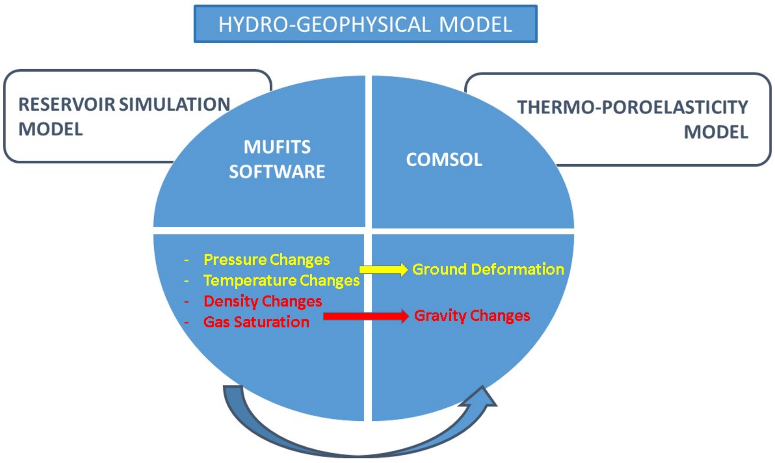


Figure 1: Operation Map of the Hydro-Geophysical Model. The hydro-geophysical model includes: (i) the **reservoir simulation model** which simulates the circulation of a fluid in a porous medium using the MUFITS code and allows to evaluate the variations of temperature, pressure and density, (ii) the **thermo-poro-elastic model**, developed within the COMSOL software, which evaluates the ground deformation and gravity changes using MUFITS solutions (temperature, pressure and density changes). The two models were linked by implementing codes in MATLAB for data transfer.

2. GRAVITY AND GROUND DEFORMATION FORMULATIONS

We here describe the models and the governing equations. A brief overview of the MUFITS code is given to illustrate the input data to be provided to run properly the code.

2.1 Reservoir Simulation Model

MUFITS software

The circulation of a non-isothermal multiphase mixture of water and carbon dioxide, under sub- and super-critical conditions, is simulated exploiting the BINMIXT module of the MUFITS software. In addition to the MUFITS, also the EOS2 module of the TOUGH2 software (Pruess et al. 1999) simulates the flows of binary mixture of water and carbon dioxide in porous media. However, we have chosen to use the MUFITS software mainly for two reasons:

1. To allow a more realistic modeling of geophysical observables. In TOUGH2 in fact the injection of the mixture is limited to the most superficial area as the software works up to a temperature of 350 °C. The presence of supercritical fluids, generally present due to the high temperatures reached in the deeper areas of a volcanic edifice, is thus neglected.

2. For a greater numerical stability in the vicinity of critical conditions (Afanasyev 2013b).

Hydrodynamic simulations

MUFITS uses a different technique to evaluate the properties of the mixture. They are specified by means of a thermodynamic potential $\bar{\sigma}(P, h, x)$ as a function of the primary variables (pressure P, total enthalpy h and total composition x of the mixture), which is calculated with an iterative procedure only once before the hydrodynamic simulations. The calculated potential is used, in a first step, to define, in a non-iterative manner, the properties of the mixture, and in the next step, in the hydrodynamic simulations for the resolution, in every cell, of a maximum conditional problem to evaluate the multiphase thermodynamic equilibrium.
The flow of the mixture is then simulated by solving the hydrodynamic equations (Afanasyev 2013a; Coco et al. 2016), i.e. the mass balance equations (1), in equal number $k = 2$ of components of the mixture, the energy balance equation (2), and the Darcy's law (3) at which the mass flow rates $F_\beta = \rho_\beta \mathbf{w}_\beta$ obey:

$$\frac{\partial\left(\phi \sum_\beta \left(\rho_\beta S_\beta \chi_\beta^k\right)\right)}{\partial t} + \nabla \cdot \left(\sum_\beta \chi_\beta^k \rho_\beta \mathbf{w}_\beta\right) = q^{M^k}, k = 1, 2, \tag{1}$$

$$\frac{\partial\left(\phi \sum_\beta \left(\rho_\beta S_\beta e_\beta\right) + (1-\phi)\, \rho_r e_r\right)}{\partial t} + \nabla \cdot \left(-\tilde{\lambda} \nabla T + \sum_\beta \left(h_\beta \rho_\beta \mathbf{w}_\beta\right)\right) = q^E, \tag{2}$$

$$\mathbf{w}_\beta = -K \frac{k_\beta}{\mu_\beta} \left(\nabla P - \rho_\beta \mathbf{g}\right). \tag{3}$$

While $\mathbf{w}_\beta$ is the Darcy velocity at phase β, the equations (1-3) allow to evaluate the fluid phases composition, gas saturation distribution and temperature, pore pressure, and density variations. The equations are solved for the primary variables (pressure, total enthalpy and total composition), which are calculated with the iterative procedure to evaluate the fluid phase equilibria (Afanasyev 2012, 2013a). The other quantities appearing in equations (1-3) (Table 1) can be user-supplied para-
meters or non-linear functions of the primary variables:

Primary Variables	**Variables**	**Parameters**
h Total enthalpy P Pressure x Total composition	T Temperature S_β Saturation of the phase β e_β Internal energy of the phase β χ_β^k Mass fraction of the k-th component of the phase β ρ_β Density of the phase β h_β Enthalpy of the phase β k_β Permeability of the phase β μ_β Viscosity of the phase β e_r Rock grain internal energy	ϕ Porosity $\tilde{\lambda}$ Thermal conductivity K Absolute permeability ρ_r Rock grain density q^{M^k}, q^E Source terms $\mathbf{g}$ Gravity acceleration

Table 1: Parameters and variables of the governing equations (1-3)

The equations (1-3) does not present any singularities in the vicinity of the critical thermodynamic conditions since these primary variables are used, ensuring greater stability and thus avoiding a reduction of the time step. As a result, MUFITS also ensures a CPU time saving since the maximum conditional problem and the hydrodynamic equations (1-3) are solved for the same thermodynamic variables, which therefore do not need to be recalculated, as there is no transition to a different set of variables at phase transitions.

The MUFITS code allows to simulate three phases β: a liquid phase consisting mostly of H_2O, a gaseous phase consisting of vapor H_2O and gaseous CO_2, and a phase consisting of liquid CO_2 present only at high pressures and temperatures below to the critical temperature of CO_2.

Two different periods are generally simulated:

(1) the steady state phase (simulation 1) which leads to the achievement of the stationary conditions of pressure and temperature;

(2) the unrest phase (simulation 2) in which the geothermal system is perturbed, with a change in the flow rate and chemical composition of the injected mixture and/or the permeability of the porous medium.

The aim is to evaluate, at the end of the unrest phase, the variations in temperature, pore pressure and density with respect to the stationary conditions, to modeling the geophysical observables.

2.2 Thermo-poroelastic model

The thermo-poroelastic model allows to evaluate the ground deformation and the gravity changes due to temperature, density and pressure variations which are provided by the MUFITS simulations.
To evaluate them, we have adopted two different methods: the first uses a semi-analytical solution and the second uses the COMSOL Multiphysics software which solves a set of PDEs that describe the geophysical problem.

2.2.1 Semi-Analytical Solution

Gravity changes

We start with the description of the semi-analytical solution in which the variation of gravity at an observation point P is calculated using the point mass solution (Hemmings 2014).

The approach used by Hemmings (2014) to give a gravitational formulation takes into account the contribution of the gravitational force F, calculated according to Newton's law of universal gravitation, that each cell element (x_i, z_i) in the 2D domain (plane x-z) exerts on an observation point P(x_P, z_P) placed on the surface (see Figure 2):

$$F = \frac{Gm_i}{s^2}, \tag{4}$$

with $G = 6.67\ 10^{-11} \frac{m^3}{kgs^2}$ the universal gravitational constant, m_i the mass of the element (x_i, z_i) and $s = \sqrt{(x_i - x_P)^2 + (z_i - z_P)^2}$ the distance between the element (x_i, z_i) and the point P(x_P, z_P).

The gravimeters record the vertical component of this force, which is given by:

$$F_z = \frac{Gm_i}{s^2} \sin\alpha = \frac{Gm_i}{s^3}(z_P - z_i). \tag{5}$$

By integrating over the entire domain the gravitational signal is generated by density variation $\Delta\rho_i$ of all elements in 2D, at time Δt on the point P:

$$\Delta g(\mathrm{P}, \Delta t) = \int_x \int_z \frac{G\Delta\rho_i}{s^3}(z_P - z_i)dxdz. \tag{6}$$

Clearly, since a 2D domain has been considered, while the gravity measurements depend on the density variations in all directions, it is appropriate to extend these results into a 3D domain. The approach presented by Hemmings (2014) is to consider an axisymmetrical mass distribution.

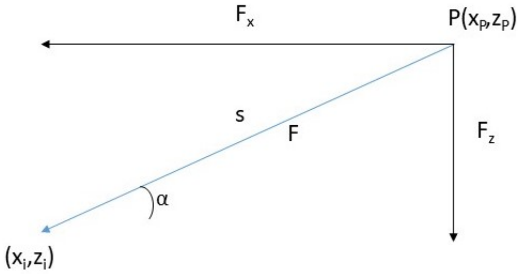


Figure 2: Gravitational force that the element (x_i, z_i) exerts on the survey point P (after Hemmings 2014)

We consider an axisymmetric domain with respect to $x = 0$ and an element (x_i, z_i) in the 2D domain (plane x-z), which is an element of a ring centered at $x = 0$ and radius x_i. We indicate with $m_i = \rho_i x_i dxdzd\vartheta$ the mass of the element (x_i, z_i), with $dxdz$ the area and $d\vartheta$ the length (see Figure 3).

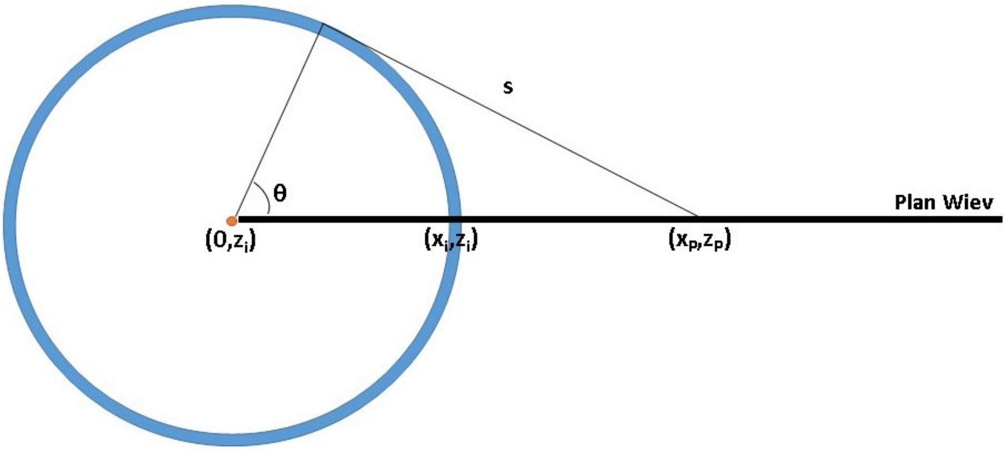


Figure 3: Elemental ring Scheme (after Hemmings 2014)

In cylindrical coordinates the vertical gradient of the gravitational potential due to a single ring detected on the point P is obtained by integrating around the ring:

$$g_i(\mathrm{P}) = G\rho_i x_i dx dz \int_0^{2\pi} \frac{z_P - z_i}{(x_i^2 - 2x_P x_i \cos\vartheta + x_P^2 + (z_P - z_i)^2)^{\frac{3}{2}}} \mathrm{d}\vartheta, \tag{7}$$

where $0 < \vartheta < 2\pi$, and, by adding on the contributions given by each ring, we obtain the gravity change due to a 2D axisymmetrical mass distribution detected on the point P:

$$\Delta g(\mathrm{P}, \Delta t) = G \sum_{x_i} \sum_{z_i} x_i \Delta\rho_i dx_i dz_i (z_P - z_i) \int_0^{2\pi} \frac{1}{(x_i^2 - 2x_P x_i \cos\vartheta + x_P^2 + (z_P - z_i)^2)^{\frac{3}{2}}} \mathrm{d}\vartheta. \tag{8}$$

Ground deformation

We now illustrate the vertical ground deformation, considering the case of a single dipole without a pre-existing cavity in the medium (Afanasyev and Utkin 2020; Davies 2003; Rinaldi et al. 2010; Rinaldi et al. 2011). We first consider the displacement $Uz_i(\mathrm{P})$ due to a single dipole and then we sum the single contributions to obtain the total displacement at the observation point.

The variation of the relative volume (dilatation or compression) of each grid block centered in the element (x_i, z_i) is given by (Rinaldi et al. 2010):

$$\varepsilon_i = \frac{\Delta V_i}{V_i} = \frac{\beta}{K_d}\Delta P + \alpha_T \Delta T,$$

where α_T is the volumetric thermal expansion coefficient and $\beta = 1 - \frac{K_d}{K_s}$ is the Biot-Willis coefficient, with K_d and K_s the drained and solid bulk moduli, respectively.

From Rinaldi et al. (2010) we have:

$$Uz_i(\mathrm{P}) = -\frac{1+\nu}{12\pi(1-\nu)}\varepsilon_i V_i \left[\frac{-h}{((x_P - x_i)^2 + (z_P - z_i)^2)^{\frac{3}{2}}} - (3-4\nu)\frac{h}{((x_P - x_i)^2 + (z_P - z_i)^2)^{\frac{3}{2}}} \right], \tag{9}$$

from which:

$$Uz_i(\mathrm{P}) = \frac{1+\nu}{12\pi(1-\nu)}\varepsilon_i V_i \left[4(1-\nu)\frac{h}{((x_P - x_i)^2 + (z_P - z_i)^2)^{\frac{3}{2}}} \right] = \frac{1+\nu}{3\pi}\varepsilon_i V_i \frac{h}{((x_P - x_i)^2 + (z_P - z_i)^2)^{\frac{3}{2}}}, \tag{10}$$

where ν is the Poisson's ratio and h is the arm of the dipole.

As in the case of the calculation of gravity changes, in cylindrical coordinates, we have:

$$Uz_i(\mathrm{P}) = \frac{1+\nu}{3\pi}\varepsilon_i x_i dx dz \int_0^{2\pi} \frac{z_P - z_i}{(x_i^2 - 2x_P x_i \cos\vartheta + x_P^2 + (z_P - z_i)^2)^{\frac{3}{2}}} \mathrm{d}\vartheta. \tag{11}$$

The vertical displacement of an axisymmetrical mass distribution detected on the point P is obtained summing on the single contributions.

The integral in formulas (7) and (11) is computed numerically using the **quad** function of MATLAB, which solves the integrals with the recursive adaptive Simpson quadrature formula.

2.2.2 COMSOL Solution

The method just described has a limited range of applications, as the solutions have been derived under the assumption of an homogeneous half-space model. To overcome this limit we considered a numerical finite element method, implemented within the COMSOL software, which evaluates ground deformation and changes in gravity, even on heterogeneous and with topography domains.
The modules used in COMSOL are the Solid Mechanics and the classical PDE for the calculation of the ground deformation and gravity changes.

Ground deformation

Variations in temperature and in pore pressure are responsible for the ground deformation. The model solves the elastostatic equation coupled with thermo-poroelastic extension of the Hooke's law (Fung 1965; Jaeger et al. 2007), which takes into account of changes in temperature ΔT and pressure ΔP:

$$\nabla \cdot \sigma = 0, \tag{12}$$

$$\sigma = \lambda \mathrm{tr}(\varepsilon) + 2\mu\varepsilon + \alpha_T K_d \Delta T\mathbf{I} + \beta\Delta P\mathbf{I}, \tag{13}$$

$$\varepsilon = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right). \tag{14}$$

In the equations, σ and ε are the stress and strain tensors, respectively, λ and μ are the Lame's elastic medium parameters and $\mathbf{u}$ is the unknown deformation vector.
If $\mathbf{n}$ is the normal vector to the ground surface, to close the problem zero displacements are imposed at infinity and stress-free boundary condition $\sigma \cdot \mathbf{n} = 0$ on the ground surface.

Moreover, in the model it is assumed that the deformation occurs very slowly over time, to allow the pressure equilibrium and the quasi-static equilibrium of the rock.

Gravity Changes

Variations in density are responsible for the gravity changes. The model solves a Poisson equation for the gravitational potential φ_g (Currenti and Napoli 2017):

$$\nabla^2\varphi_g = -4\pi G\Delta\rho, \tag{15}$$

$$\Delta g = -\frac{\partial\varphi_g}{\partial z}. \tag{16}$$

In the equation, the total density variation $\Delta\rho$ of the fluid, which occupies the porous volume fraction ϕ, is calculated as follows: $\Delta\rho = \phi\sum_\beta \rho_\beta S_\beta$.

To close the problem the gravitational potential is set to zero at infinity.

We developed the codes useful to simulate the injection of hot fluid in an axi-symmetric domain. The inlet is placed at the bottom, near the axis of symmetry. Below, we describe the codes implemented in MATLAB scripts for: (i) automating the generation of the input file of MUFITS; (ii) the transferring of MUFITS solutions for each time step to COMSOL; (iii) solving iteratively the elastostatic equation and the gravity Poisson PDE under COMSOL software for each time step.

2.3 Numerical codes

We implemented the codes distinguishing two different phases:

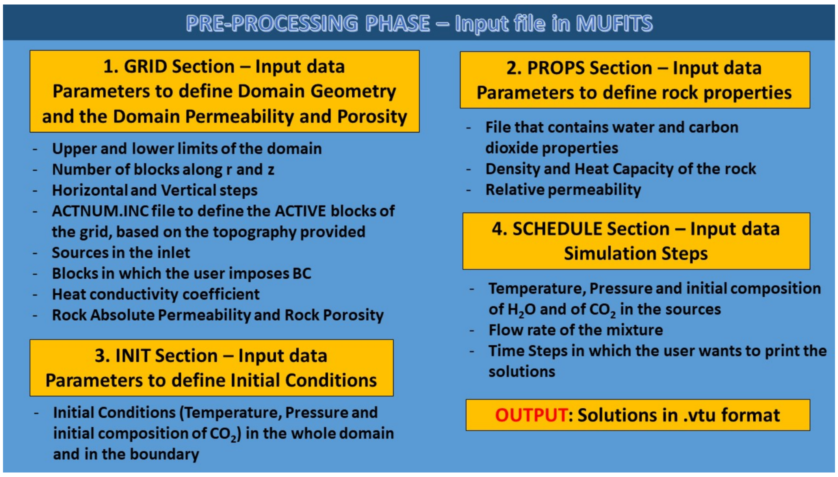
- *Pre-Processing Phase*, which concerns the generation of the input file used by MUFITS for the simulation (simulation 1) at the end of which the solutions, including pore pressure, temperature, fluid composition, gas saturation and density, are released for each grid block;
- *Post-Processing Phase*, in which we developed two different codes:
  1. a code that allows to write the input file for a further simulation (# 2) in MUFITS that extends the previous simulation (# 1) and in which the initial conditions are set up from the temperature, pore pressure and mixture composition solutions released by the previous simulation (# 1);
  2. a code that reads the values of temperature, pressure, density and gas saturation released at the end of the simulation 1 and those released by the simulation 2 for each time step, to evaluate their variations over time beetwen the two simulations and provide them to the COMSOL software in order to compute the ground deformation and gravity changes.

The subroutines we have implemented and which we will describe in detail below are summarized in Table 2:

PRE-PROCESSING PHASE subroutines	**POST-PROCESSING PHASE subroutines**
Geometry.m: to define the geometry of the problem. Releases the thicknesses and the number of blocks along the r axis and the number of blocks along the k axis	**READVTK.m:** to read the .vtk file solutions released by the simulation in MUFITS
act.m: to define the active and inactive blocks of the grid	**Init_MUFITS.m:** to write the initial conditions for a new simulation in MUFITS (# 2) starting from the values released of a previous simulation (# 1)
SampleRocks.m: to define the different regions with certain permeability and porosity values	**Solution_def.m:** to compute ground deformation
Generate_INPUT_MUFITS.m: calling previous subroutines, generates the MUFITS input file	**Solution_grav.m:** to compute gravity changes

Table 2: Subroutines developed for pre- and post-processing phases

a. PRE-PROCESSING PHASE - MUFITS input file



PRE-PROCESSING PHASE – Input file in MUFITS

1. GRID Section – Input data
Parameters to define Domain Geometry and the Domain Permeability and Porosity

- Upper and lower limits of the domain
- Number of blocks along r and z
- Horizontal and Vertical steps
- ACTNUM.INC file to define the ACTIVE blocks of the grid, based on the topography provided
- Sources in the inlet
- Blocks in which the user imposes BC
- Heat conductivity coefficient
- Rock Absolute Permeability and Rock Porosity

3. INIT Section – Input data
Parameters to define Initial Conditions

- Initial Conditions (Temperature, Pressure and initial composition of CO_2) in the whole domain and in the boundary

2. PROPS Section – Input data
Parameters to define rock properties

- File that contains water and carbon dioxide properties
- Density and Heat Capacity of the rock
- Relative permeability

4. SCHEDULE Section – Input data
Simulation Steps

- Temperature, Pressure and initial composition of H_2O and of CO_2 in the sources
- Flow rate of the mixture
- Time Steps in which the user wants to print the solutions

OUTPUT: Solutions in .vtu format

Figure 4: Pre-Processing Scheme

We have first implemented a script (**Generate_INPUT_MUFITS.m**) that automatically releases, on the basis of some data provided by the user as input to the code, the INPUT file **SPE1.RUN** for an axisymmetric domain, necessary for the MUFITS software to start the hydrodynamic simulation. The script writes the file SPE1.RUN, which contains several data organized in SECTIONS, according to MUFITS input file, which are listed below:

- **Section GRID** in which the user inserts the data necessary to set up the axi-symmetric domain: the number of blocks along r and k directions and their thicknesses, the geometry of the domain, the blocks in which the user defines the point sources, the blocks in which the user imposes the Dirichlet boundary conditions, the heat conduction coefficient $\tilde{\lambda}$, the rock absolute permeability K and the rock porosity ϕ.

- **Section PROPS** in which the user inserts the properties of the rocks: the rock grain density ρ_r and heat capacity, the relative permeability of each phase k_β.

- **Section INIT** in which the user inserts the initial conditions in the whole domain. The user enters also the values that the temperature, pressure and initial composition assume in the boundary.

- **Section SCHEDULE** in which the user establishes the sequence of steps to simulate: define the temperature, pressure and initial composition values of the mixture to be injected in the inlet (in the point sources) (this procedure can also be inserted in the INIT section), define the total flow rate to be injected, define the duration of the simulation (in days) and the time steps (in days) in which the user wants to print the solution.

- **Section POST** in which the .SUM solutions (one for each time step established by the user in the previous section) are converted into the .vtu format.

During the simulation, together with the input file, the EOS-file **CO2H2O_V3.0.EOS** for the prediction of the thermophysical properties of water and carbon dioxide mixture, of the sub-critical and super-critical conditions for water and pure carbon dioxide, of the mono-, bi- and tri-phase states of the mixture, must be provided. The code **Generate_INPUT_MUFITS.m** automatically loads it in the PROPS section.

We describe in detail the various sections and in particular the data that the user must provide as input to the code so that the file SPE1.RUN is automatically created.

Section GRID

The input values that the user must provide are:

- **par__ss**: this parameter can assume two values: it assumes value 1 if the user is writing the input file for the simulation 1, it assumes value 2 if the user is writing the input file for the simulation 2, which is a continuation of the previous simulation (post-processing phase). In the pre-processing phase, therefore, the parameter must have value 1.

- **L** [m], **lower** [m], **top** [m]: these parameters define respectively the length of the domain (**L**) and the lower (**lower**, < 0 if the domain is below sea level) and upper (**top**, < 0 if the domain is below sea level) limit along the z axis. With these parameters provided by the user, the code automatically writes to the file SPE1.RUN a domain with these length and height limits:

  RTZBOUND
  0 L -0.0314159 0.0314159 -top -lower /

  To specify the grid, in an axisymmetric domain, MUFITS considers only the hundredth part of the entire circle. After defining all the parameters, always in the Section GRID, the code automatically writes to the file SPE1.RUN the keyword that allows to multiply by 100 the cell volumes and interface areas (both in r and k directions).

- **stepV** [m], **l** [m], **N**: through these parameters supplied by the user, the horizontal and vertical steps of the grid blocks are established. In particulare, **stepV** defines the thickness of the blocks along the k direction, assuming the grid is uniform. The parameters **l** and **N** allow to define the thickness (which increase in geometric scale) of the blocks along the r direction. These parameters are used to solve the following equation: $lx^{N+1} - Lx + (L - l) = 0$, of which we consider the largest root q among those greater than 1. The thickness are then defined by the cumulative sum of the vector $[l, lq, lq^2, ..., lq^N]$. The function **Geometry.m** called by the code **Generate__INPUT__MUFITS.m**, allows to calculate these roots and automatically releases the thickness and the number of blocks (**Col**) along the r direction and the number of blocks (**Row**) along the z direction, necessary to set the file input.

  The code automatically writes to the file SPE1.RUN:

  RADIAL Col 1 Row /,
  to define the hundredth part of an axi-symmetric domain (RADIAL) with **Row** blocks along the k direction and **Col** blocks along the r direction,

  DRV
  block 1 thickness block 2 thickness ... /,
  to define (from left to right) the thickness (in meters) of each grid block along the r direction,

  DZV
  Row*stepV /,
  to define (from top to bottom) the thickness (in meters) of each grid block along the k direction assuming the grid is uniform. Otherwise it is necessary to report the thickness of each grid block as seen above.

- **act_val**: through this parameter the user defines if the domain is flat (value 0) or has a topography (value 1). In this second case the topography must be supplied to the code (providing the coordinates along the r axis and along the z axis of the boundaries of the chosen geometry). The code automatically prints, by calling the function **act.m**, a file **ACTNUM.INC** that is supplied during the simulation together with the file SPE1.RUN. The file ACTNUM.INC defines the active (i.e. it is part of the computational domain and it is marked with value 1) and inactive (marked with value 0) blocks of the grid based on the topography of the domain. At the same time the code releases a matrix **Dom** whose number of rows coincides with the number of blocks along k and the number of columns coincides with the number of blocks along r. Each element of the matrix indicates a block of the grid which takes the value 1 if it is an active block, value -1 if it is an inactive block and value 0 if it is an active block in the upper limit. The blocks in the matrix are sorted from left to right and from top to bottom.

- **inlet** [m]: the length of the inlet from which the injection of the CO_2 and H_2O flow rates is simulated. In the code the inlet is placed at the base of the domain near the axis of symmetry. The code automatically computes the number n of point sources in the inlet and the blocks (along r and k) in which to define those sources. The code writes to the file SPE1.RUN:

  ```
  SRCSPECG
        INJE1 1 1 Row /
        INJE2 2 1 Row /
        INJE3 3 1 Row /
        INJE4 4 1 Row /
        ⋮
        INJEn n 1 Row /
  /,
  ```

  where **n 1 Row** indicate respectively the blocks along r, y and k in which the n-th source (**INJEn** - 8-byte character) is placed.

- **par_TP** and **par_heat**: The code Generate_INPUT_MUFITS.m automatically allows the user to impose the boundary conditions through the keyword **BOUNDARY**. With the keyword BOUNDARY, MUFITS creates additional grid blocks in the direction assigned by the user. In particular, the code was designed to impose Dirichlet conditions (through the **INFTHIN** keyword within BOUNDARY keyword) on the upper boundary of the domain. Boundary conditions other than those upper BC must be entered manually by the user in the input file.

  For example the code writes:

  ```
  BOUNDARY
        111 1   58  2* 1  21 I- I+ K- 3* INFTHIN ACTBASED 3* 1 1* CELL /
        222 59 171 2* 22 75 I- I+ K- 3* INFTHIN ACTBASED 3* 1 1* CELL /
  /
  ```

  to impose Dirichlet conditions in the upper boundary (**I- I+ K-**) for grid blocks characterized by coordinate values $z \geq 0$ (111) and for grid blocks characterized by coordinate values $z < 0$ (222). The code automatically compute and writes the blocks along r and k directions where to impose the conditions. In the example, the BC 111 are imposed on blocks that have index r from 1 to 58 and index k from 1 to 21.

In general, to impose the Dirichlet conditions in the upper boundary, it must be written on the input file:

BOUNDARY
  frn $\mathbf{r}_{min}$ $\mathbf{r}_{max}$ **2*** $\mathbf{k}_{min}$ $\mathbf{k}_{max}$ **I- I+ K- 3* INFTHIN ACTBASED 3* 1 1* CELL /**
/,

where **frn** is the fluxnum region number.
The number **1** indicates the TYPENUM number, which assumes the default value 1. It assumes value 2 for impermeable blocks in which only the heat conduction equation (2) is solved.

The grid blocks that represent boundary conditions are marked with the keyword **ACTNUM** equal to 2 in the case in which the pressure and temperature values are to be fixed in these blocks. For the upper BC the code does it automatically. For example the code writes:

REGALL
EQUALREG
  ACTNUM 2 FLUXNUM 111 /
  ACTNUM 2 FLUXNUM 222 /
/

to fix pressure and temperature in the upper BC.

The only parameters that the user must provide to impose the BC in the most appropriate way are: **par_TP** and **par_heat**. The parameter **par_TP** allows to fix the pressure and the temperature also in the blocks of the inlet (where there are the point sources), while **par_heat** allows to fix only the temperature also in the blocks of the inlet. In this second case the code automatically assigns a high value to the heat capacity in those blocks. **par_TP** and **par_heat** assume a value of 1 if the user wants to set the parameters, otherwise they must be set equal to 0.

The keyword **REGALL** must be inserted at the beginning of each section (except POST section) where arithmetic operations are performed on grid blocks created with BOUNDARY keyword.

- **typenum**: this parameter, provided by the user, allows to define permeable boundaries (value 1) and impermeable boundaries (value 2). The value is inserted by the code within the BOUNDARY keyword. In the example given above it is the value 1 which we highlighted in blue.
- **cond**: to define the heat conductivity coefficient [W/m/K].
- **include_perm**: The model can have homogeneous or heterogeneous permeability and porosity configurations. The parameter **inlude_perm** takes the value 0 if the configurations are homogeneous, 1 if the configurations are heterogeneous. In the first case, the user assigns the permeability and porosity values of the entire domain through the parameters **permability** and **porosity**. In the second case, the code calls the function **SampleRocks.m** to define the different regions of the model with the same permeability and porosity values and assigns the values using the parameters **perm** and **pore** given by the user in the blocks of the regions. In this second case, the code automatically prints a file **PERM.INC** that assigns the permeability and porosity values given by the user to each grid block and that must be supplied at simulation together with the input file.

Section PROPS

The input values that the user must provide are:

- **rock_density**: to define the rock grain density [Kg/m^3].
- **heat_capacity**: to define the heat capacity [kJ/Kg/K].
- The relative permeability of the rock must also be entered in this section, through the keyword **SATTAB**. The code was designed following the Corey's law:

  $$k_l = \widehat{S}^4, \; k_g = (1 - \widehat{S}^2)(1 - \widehat{S})^2,$$

  where $\widehat{S} = \frac{S_l - S_{lr}}{1 - S_{lr} - S_{gr}}$, with $S_{lr} = 0.33$ and $S_{gr} = 0.05$ the irriducible liquid saturation and irriducible gas saturation, respectively.
- In this section, the phases for output are also defined using the **PHASES** keyword. With this keyword is possible to define the phases of the mixture, with the aim of obtaining, at the end of the simulation, the quantities related to the phases, for example, the saturation and density of the liquid and gas phases. It does not affect the solutions of temperature, pressure and total density during and at the end of the simulation. For each phase the user must enter the characteristic parameters of the phase: the name of the phase (4-byte character), the pressure, the total molar enthalpy, and the quantities of H_2O and CO_2 in the phase as a function of the molar composition:

  ```
  PHASES
      phase name 1   pressure 1   molar enthalpy 1   comp1t 1   comp2t 1 /
      phase name 2   pressure 2   molar enthalpy 2   comp1t 2   comp2t 2 /
  ...
  / ,
  ```

  where **comp1t** and **comp2t** are the molar composition of the first (CO_2) and second (H_2O) component, respectively.

  In particular, in the code we have defined two phases: one rich of H_2O (**LH2O**) and one rich of CO_2 (**SCO2**). The user must provide the pressures (**pres_h2o** and **pres_co2**) and total molar enthalpy (**ent_h2o** and **ent_co2**) characteristic of the phases. The two enthalpies at a given temperature can be obtained from a simulation for the calculation of the properties of the mixture (the input file in this case is **PVT.RUN**; you can download an example on the website of the author of the MUFITS code).

Section INIT

The code was designed to automatically assign the initial conditions considering the geothermal gradient in the whole domain. The input values that the user must provide are:

- **Ttop**: to define the temperature in the upper boundaries [°C].
- **Ptop**: to define the pressure in the upper boundaries [MPa].
- **gradT**: to define the temperature gradient [°C/m].
- **gradP**: to define the pressure gradient [MPa/m].

At the upper boundary the temperature and pressure are respectively **Ttop** and **Ptop**, while in the domain, for each grid block the pressure and the temperature values are assigned in this way:

$$\mathbf{T = Ttop + hvalT*gradT,}$$
$$\mathbf{P = Ptop + hvalP*gradP,}$$

where **hvalT** and **hvalP** are the quantities represented in figure 8. In particular, **hvalT** is the distance of the grid block from the upper limit, **hvalP** indicates the same distance only for grid blocks that fall within the perimeter of the domain extending above sea level. Otherwise it is the distance from the grid block to the free surface of the water. **hvalT** [m] and **hvalP** [m] values for each grid block are gived by two files **HVALT.INC** and **HVALP.INC** which are automatically printed by the code and must be provided during the hydrodynamic simulation together with the input file SPE1.RUN.

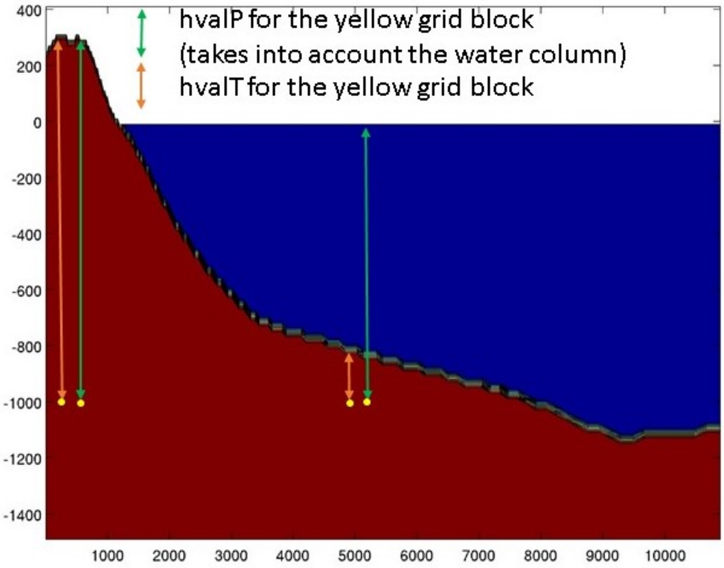


Figure 5: **hvalT** and **hvalP** values

The molar composition in the whole domain and in the upper boundary is obtained from the Dalton's law on partial pressures, assuming that the partial pressure of CO_2 is derived from the total pressure by dividing by 4 10^6.

In this section it is also possible to enter the pressure, temperature and molar composition of the mixture of H_2O and CO_2 to be injected. Otherwise it can also be inserted in the next section.

Section SCHEDULE

The input values that the user must provide are:

- **Temp_inlet** [°C] and **Pres_inlet** [MPa]: to define temperature and pressure of the mixture being injected. For example, the code write in the input file:

```
RESETNAM
        INJE* /
/
```

```
 EQUALNAM
        PRES 19.0 INJE* /
        TEMPC 350.0 /
        COMP1T 0.166698 /
        COMP2T 0.833302 /
/

UPDATNAM
        INJE* /
/
```

to inject in the point sources (INJEi) a mixture of H_2O (about 83%) and CO_2 (about 17%) at a temperature of 350 °C and a pressure of 19 MPa. The keywords **RESETNAM** and **UPDATNAM** must be entered if the pressure, temperature and molar composition of the mixture are defined in this section.

- **H2O_rate** and **CO2_rate**: to define the total flow rate of water and of carbon dioxide (in ton / day) to be injected through the sources. The code automatically calculates the flow to be supplied to each point source of the inlet, and the initial compositions **comp2t_inlet** and **comp1t_inlet** of water and of carbon dioxide in the inlet blocks inserted with **EQUALNAM** keyword.
- **pres_out** [MPa]: to define the maximum pressure at the pump outlet. The user must enter the maximum pressure high enough to ensure that it is not reached in the simulation.
- In this section also provides information on the data the user want to output:

```
RPTSOL
        ASCII FLUXNUM COMP1T PHST PRES TEMP SAT#LH2O SAT#SCO2 ENTHT
        COMP2T K-IJKRES FLUXK#T FLUXI#T /

RPTSUM
        PRES TEMPC COMP1T COMP2T ENTHT SAT#LH2O SAT#SCO2 CP1#LH2O
        CP1#SCO2 CP2#LH2O CP2#SCO2 DEN#LH2O DEN#SCO2 DENT PERMI ASCII
/
```

In the example, in output, from RPTSUM, we will have: pressure, temperature, composition of carbon dioxide, composition of water, total enthalpy, saturation of liquid phase (phase defined with the keyword PHASES), saturation of gas phase (phase defined with the keyword PHASES), molar fraction of carbon dioxide in liquid phase, molar fraction of carbon dioxide in gas phase, molar fraction of water in liquid phase, molar fraction of water in gas phase, density of liquid phase, density of gas phase, total density and permeability. While from RPTSOL we have also **FLUXK#T** [ton/day] and **FLUXI#T** [ton/day] which indicate for each grid block the total mass flow along k and r directions, respectively.

- Finally, information on the time step must be given:

```
TUNING
        dtlimit dtmax dtexp dtmin /,
```

where
dtexp and **dtlimit** indicate the next time step and its limit, respectively,
dtmax and **dtmin** indicate the maximal and the minimal time step, respectively,

and

```
TSTEP
        tstep1 tstep2 tstep3 ... /,
```

where **tstep#** indicates the time step [day]. For each **tstep#** the simulation proceeds and a new .SUM solution file is printed.

- If we insert the keyword EXPORT at the end of this section, at the end of the simulation the values of temperature, pressure and composition (of CO_2 and/or H_2O) can be released, which can be used as initial conditions for a subsequent simulation (# 2):

```
EXPORT
        PRES 2* PRESSURE.INC /
EXPORT
        TEMPC 2* TEMPERATURE.INC /
EXPORT
        COMP1T 2* COMPT.INC /
```

Section POST

For each time step, the simulation releases a .SUM solution file. Only at the end of the simulation, the MUFITS, through the POST section in the file input, converts these solutions in the .vtu format.

OUTPUT of the code Generate_INPUT_MUFITS.m:

- Input file **SPE1.RUN**;
- the matrix that gives information on active and inactive blocks by means of a file **Dom.txt**;
- **Blocks_bottom.txt** which contains the indexes of active blocks from bottom to top, **blocks.txt** which contains the indexes of active blocks from top to bottom;
- files **HVALT.INC** and **HVALP.INC** to set the initial conditions;
- files **ACTNUM.INC** and **PERM.INC** if requested by the user to define, respectively, the active blocks and the rock permeability and porosity values.

OUTPUT of the MUFITS simulation:

- **TEMPERATURE.INC**, **PRESSURE.INC**, **COMPT.INC** files, if requested by the user.
- Solutions in .vtu format. The solutions in the .vtu format released by the simulation in MUFITS are displayed through the **ParaView** platform, which converts them into the .vtk format. We have used the **version 3.12**.

ParaView

MUFITS: Solution in .vtu format $\Longrightarrow$ Solution in .vtk format

b. POST-PROCESSING PHASE

b.1 Input file of a simulation that is a continuation of a previous simulation

We now describe the codes we implemented to write the file input **SPE1.RUN** of a simulation which is the continuation of a previous simulation (# 2). The peculiarity of this new simulation is that the values of the initial conditions (temperature, pressure, mixture composition) are provided by the output of the previous simulation (# 1).

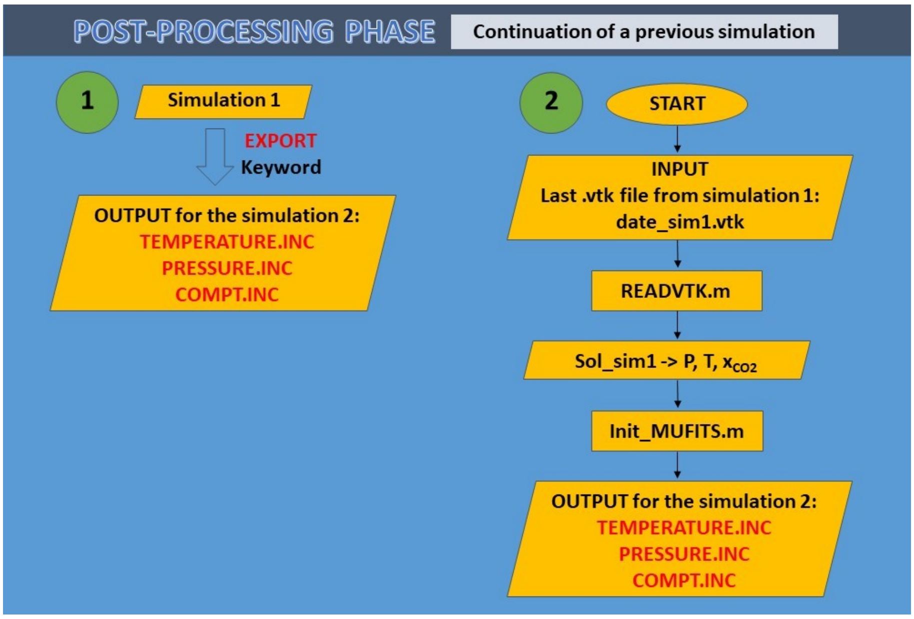


Figure 6: Post-Processing Scheme: Continuation of a previous simulation

The code to write the file input for the simulation 2 is always **Generate_INPUT_MUFITS.m**. The parameters that the user must supply are those already described above. It is possible that the parameters remain unchanged, or they must be changed because, for example, the second simulation is characterized by a greater flow rates of water and carbon dioxide, and/or by a change in permeability and porosity values.

The parameter that must certainly be changed by the user is **par_ss**, which must have value 2. There are, now, two different methods to impose the initial conditions for simulation 2 (Figure 6).

1. The user already has the pressure, temperature and composition values of the mixture through the TEMPERATURE.INC, PRESSURE.INC and COMPT.INC files released by the previous simulation. Otherwise,

2. Generate_INPUT_MUFITS.m calls the following two functions before writing the input file:

   - **READVTK.m**:

     Input:

     (a) Dom.txt released by the code Generate_INPUT_MUFITS.m in the previous simulation (# 1);

     (b) Last .vtk file of the first simulation provided from ParaView (date_sim1.vtk).

     The code reads the file .vtk and creates a table **Sol_sim1** which contains for each active grid block the values of the parameters that have been specified in the MUFITS input file using the RPTSOL and RPTSUM keywords. In particularthe code was written to release automatically: the values of the coordinates along r and along k directions, pressure, temperature, gas saturation, density of the liquid phase, density of the gas phase, total density, molar composition of carbon dioxide, mass flow rate along r and k directions. If the user want solutions other than these, he must enter the parameters as output parameters in RPTSUM, and modify the code READVTK.m.

     Output: Sol_sim1

$$\begin{array}{ccccccccccc} x_1 & z_1 & P_1 & T_1 & S_{g_1} & \rho_{l_1} & \rho_{g_1} & \rho_1 & x_{\mathrm{CO_2}_1} & f_{r_1} & f_{k_1} \\ x_2 & z_2 & P_2 & T_2 & S_{g_2} & \rho_{l_2} & \rho_{g_2} & \rho_2 & x_{\mathrm{CO_2}_2} & f_{r_2} & f_{k_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

   - **Init_MUFITS.m**:

     Input:

     (a) Sol_sim1;

     (b) File **blocks.txt** released by the code Generate_INPUT_MUFITS.m in the previous simulation (# 1).

     The code reads, from the file **Sol_sim1**, the values of temperature, pressure and carbon dioxide composition for each grid block. Automatically prints the files **TEMPERATURE.INC**, **PRESSURE.INC** and **COMPT.INC**.

     Output: TEMPERATURE.INC, PRESSURE.INC and COMPT.INC. These files must be provided together with the SPE1.RUN input file to the hydrodynamic simulation.
     The code Generate_INPUT_MUFITS.m automatically loads them into the INIT section when the parameter par_ss is equal to 2. In this way the initial conditions of the new simulation (# 2) are set through these files and not through the geothermal gradient. At the end of this second simulation, all files released in .vtu format are converted to .vtk format through the ParaView platform.

b.2 Codes to evaluate ground deformation and gravity changes within COMSOL software

After completing the hydrodynamic simulations in MUFITS (# 1 and # 2), we can compute the surface ground deformation and gravity changes. We describe the codes we have implemented for data transfer between MUFITS and COMSOL (Currenti and Napoli 2017, Currenti et al. 2017): **Solution_def.m** and **Solution_grav.m** (Figure 7).

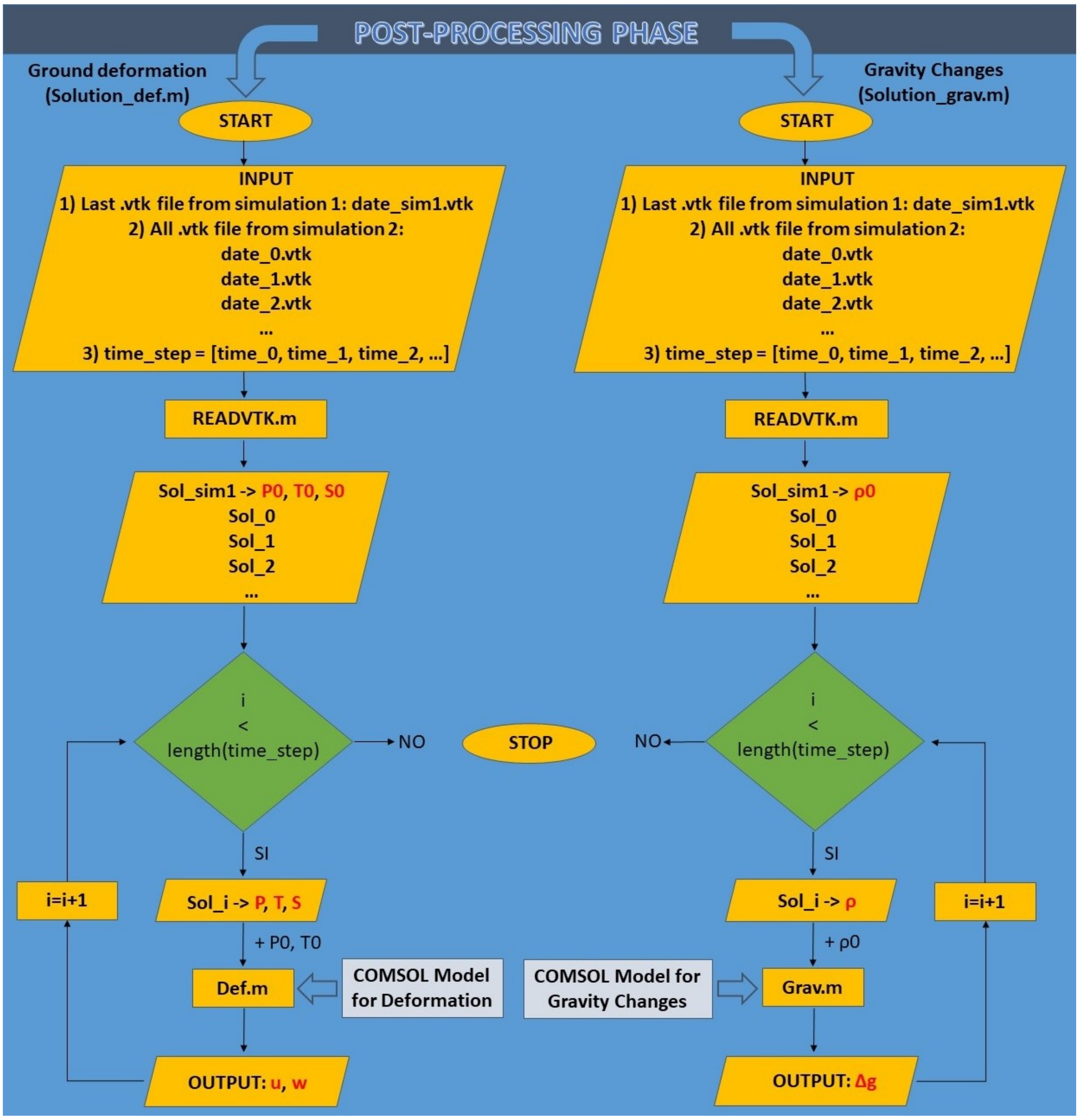


Figure 7: Post-Processing Scheme to evaluate surface ground deformation and gravity changes

Input to the matlab codes Solution_def.m and Solution_grav.m:

The data that must be provided in INPUT for both codes are:

- the last .vtk file from simulation 1: date_sim1.vtk;
- all .vtk files from simulation 2: date_i.vtk (note that the values of pressure, temperature and mixture composition between date_sim1.vtk and date_0.vtk are the same);
- the matrix Dom.txt released from the code Generate_INPUT_MUFITS.m;
- the parameters **Row** and **Col**: to define the number of blocks along k and r directions, that we can get either from the input file SPE1.RUN of one of the two simulations or from the code Generate_INPUT_MUFITS.m;
- the vector **time_step** [s]: in which the user enters the times in which the solutions (date_i.vtk) of simulation 2 were printed. So for each time step we have a .vtk file of simulation 2 (date_i.vtk) and it will therefore be possible, for each time step, to compute the variations in temperature, pressure and total density between the values of simulation 2 and the values of simulation 1 (date_sim1.vtk).

Code steps:

For each time step, the scripts call the function **READVTK.m** to create, given the .vtk files from the two simulations, the tables of solutions **Sol_i**. Therefore, the code **Solution_def.m** computes the pressure and temperature variations between the two simulations, while the code **Solution_grav.m** computes the total density variation and gas saturation between the two simulations. The values of pressure, temperature and density, for that time, are stored in the files **PT.txt** and **Rho.txt** for simulation 2 and in the files **PT0.txt** and **Rho0.txt** for simulation 1. The files **PT.txt** and **Rho.txt** change for each time step.

After the first time step, the COMSOL software must be set up to solve the equations (12-14) to evaluate vertical and horizontal ground deformations and separately to solve the equations (15-16) to evaluate gravity changes.

To compute ground deformation, COMSOL:

- Requests the topography provided through a text file (the same file provided to the code Generate_INPUT_MUFITS.m) to build the model geometry.
- Requests the values of the temperature and pressure **PT0.txt** and **PT.txt**. These values are interpolated and adapted by COMSOL on the new grid.
- Requests the values of the parameters needed to solve the equations (12-14):
  1. Lame's elastic medium parameter μ;
  2. volumetric thermal expansion coefficient α_T;
  3. bulk K_s and drained K_d moduli;
  4. Poisson's ratio ν and Young's modulus E.
- Sets the free boundary conditions along the surface of the computational domain.
- The computational domain is bounded by infinite mapped elements, that with appropriate transformation functions map the finite domain into an infinite domain, and therefore the vertical and horizontal deformations vanish toward infinity (Currenti 2014; Currenti and Napoli 2017).

To compute the gravity changes is instead necessary:

- Provide the topography.
- Provide the porosity values of the rock.
- Provide the universal gravitational constant G.
- Provide the values of density **Rho0.txt** and **Rho.txt**. These values are interpolated and adapted by COMSOL on the new grid;
- Impose that the gravitational potential is zero at infinity.

After having set the two models, COMSOL releases two scripts in MATLAB, **Def.m** and **Grav.m**, which at each time step, are recalled respectively by the scripts Solution_def.m and Solution_grav.m, allowing to automate the data transfer, since at each time step, the temperature, pressure and density files are automatically loaded, the equations are solved and the solutions released.

The boundaries of the domain in COMSOL are marked by a numerical value, which is assigned by the software during the setting phase of the model. To compute the vertical and horizontal deformations (**dat_u** and **dat_w**) and the changes in gravity (**dat**), in the codes Solution_def.m and Solution_grav.m, respectively, it is necessary to provide such values (which can be read in COMSOL in the setting phase of the model) as input. Example:

dat_u = mpheval(model, { 'u' } , 'edim', 'boundary', 'section', [15:16])

dat_w = mpheval(model, { 'w' } , 'edim', 'boundary', 'section', [15:16])

dat = mpheval(model, { 'u2z * le8' } , 'edim', 'boundary', 'section', [49:54]).

We report below the results obtained for vertical deformation and gravity changes both with the semi-analytical solution and by means of the COMSOL software, for a geothermal system in a flat and homogeneous domain. The results are in good agreement.

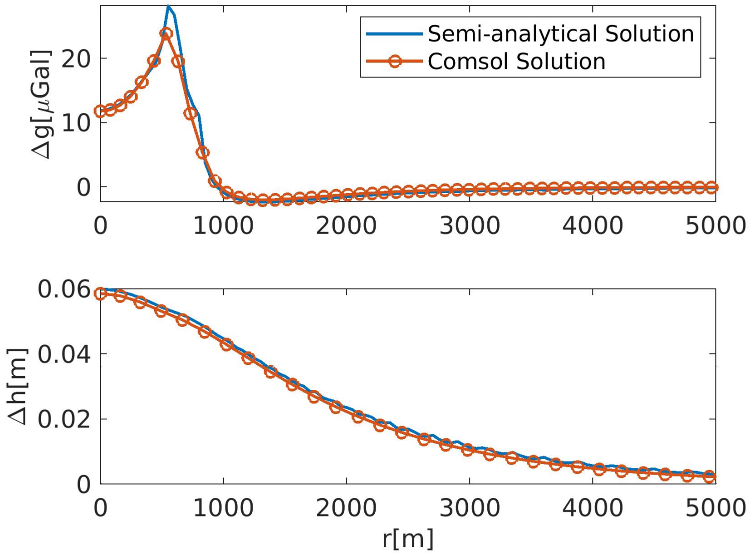


Figure 8: Comparison between semi-analytical solution and numerical solution for gravity (top) and vertical ground deformation (bottom). To derive the numerical solution in COMSOL, we have reproduced the results reported in Rinaldi et al. (2010), where the domain is flat and homogeneous. After the steady state phase in which a mixture of H_2O (2400 t/d) and CO_2 (1000 t/d) was injected, the unrest phase, in which the flow rate of the mixture and CO_2 content were increased (6000 t/d of CO_2 and 6100 t/d of H_2O), was simulated for 1 year. For both models the parameters assume the following values: $\phi = 0.2$, $\nu = 0.25$, $K_d = 5\ 10^9$ Pa, $K_s = 30\ 10^9$ Pa and $\alpha_T = 10^{-5}$ K^{-1}. For COMSOL software we have set $\mu = 2\ 10^9$ Pa and $E = 7.5\ 10^9$ Pa.

Code availability

The code is fully available upon request to the developers. Applications of the code can be found in Stissi et al. (2021).

Bibliography

Afanasyev, A. (2012). Simulation of the Properties of a Binary Carbon Dioxide-Water Mixture under Sub- and Supercritical Conditions. High Temperature 50:340-347. https://doi.org/10.1134/S0018151X12030017

Afanasyev, A. (2013a). Application of the Reservoir Simulator MUFITS for 3D Modelling of CO2 Storage in Geological Formations. Energy Procedia 40:365-374. https://doi.org/10.1016/j.egypro.2013.08.042

Afanasyev, A. (2013b). Multiphase compositional modelling of CO2 injection under subcritical conditions: The impact of dissolution and phase transitions between liquid and gaseous CO2on reservoir temperature. Int. J. Greenh. Gas Control 19, 731-742. https://doi.org/10.1016/j.ijggc.2013.01.042

Afanasyev, A., Utkin, I. (2020). Modelling the ground displacement and gravity changes with the MUFITS simulator. Adv. Geosci. 54, 89-98. https://doi.org/10.5194/adgeo-54-89-2020

Coco, A., Currenti, G., Gottsmann, J., Russo, G., and Del Negro, C. (2016). A hydro-geophysical simulator for fluid and mechanical processes in volcanic areas. J. Math. Ind. 6:6. https://doi.org/10.1186/s13362-016-0020-x

COMSOL (2012). Comsol Multiphysics 4,3. Stockholm: Comsol Ab

Currenti, G. (2014). Numerical Evidences Enabling Reconciliation Gravity And Height Changes In Volcanic Areas. Geophysical Journal International. 197(1):164-173. http://doi.org/10.1093/Gji/Ggt507

Currenti, G. M., Napoli, R. (2017). Learning about Hydrothermal Volcanic Activity by Modeling Induced Geophysical Changes. Front. Earth Sci. 5:41. https://doi.org/10.3389/feart.2017.00041

Currenti, G., Napoli, R., Coco, A., and Privitera, E. (2017). Effects of hydrothermal unrest on stress and deformation: insights from numerical modeling and application to Vulcano Island (Italy). Bull. Volcanol. 79:28. https://doi.org/10.1007/s00445-017-1110-3

Davies, J. H. (2003). Elastic field in a semi-infinite solid due to thermal expansion or a coherently misfitting inclusion. J. Appl. Mech. 70(5): 655-660. https://doi.org/10.1115/1.1602481

Del Negro, C., Currenti, G., Solaro, G., Greco, F., Pepe, A., Napoli, R., Pepe, S., Casu, F., Sansosti, E. (2013). Capturing the fingerprint of Etna volcano activity in gravity and satellite radar data. Sci Rep. 3:3089. https://doi.org/10.1038/srep03089

Drouin, V., Sigmundsson, F., Verhagen, S., Ófeigsson, B. G., Spaans, K. and Hreinsdóttir, S. (2017). Deformation at Krafla and Bjarnarflag geothermal areas, Northern Volcanic Zone of Iceland, 1993-2015. J. Volc. Geotherm. Res. 344, 92-105. https://doi.org/10.1016/j.jvolgeores.2017.06.013

Fung, Y. (1965). Foundations of Solid Mechanics. Englewood Cliffs, NJ: PrenticeHall.

Hemmings, B. J. C. (2014). Field and numerical studies of the hydrogeology of an island arc volcano: Insights from Montserrat, BWI. PhD thesis University of Bristol

Jaeger, J., Cook, N., and Zimmerman, R. (2007). Fundamentals of Rock Mechanics, 4th Edn. Oxford: Blackwell Publishing

Juncu, D., Árnadóttir, Th., Geirsson, H., Gunnarsson, G. (2019). The effect of fluid compressibility and elastic rock properties on deformation of geothermal reservoirs. Geophys. J. Int. 217, 122-134. https://doi.org/10.1093/gji/ggz011

Kearey, P., and Brooks, M. (1991). An Introduction to Geophysical Exploration, 2nd Edn. Oxford: Blackwell Scientific Publications

Mogi, K. (1958). Relations between the Eruptions of Various Volcanoes and the Deformations of the Ground Surfaces around them. Bulletin of the Earthquake Research Institute 36:99-134

Okada, Y. (1985). Surface deformation due to shear and tensile faults in a half-space. Bulletin of the seismological society of America, 75(4), 1135-1154. https://doi.org/10.1785/BSSA0750041135

Pruess, K., Oldenburg, C., Moridis, G. (1999). TOUGH2 User's Guide, Version 2.0. Berkeley, CA: Lawrence Berkeley National Laboratory

Rinaldi, A. P., Todesco, M., Bonafede, M. (2010). Hydrothermal instability and ground displacement at the Campi Flegrei caldera. Phys. Earth Planet. Inter. 178(3-4):155-161. https://doi.org/10.1016/j.pepi.2009.09.005

Rinaldi, A., Todesco, M., Bonafede, M., Vandemeulebrouck, M.J., Revil, A. (2011). Electrical conductivity, ground displacement, gravity changes, and gas flow at Solfatara crater (Campi Flegrei caldera, Italy): Results from numerical modeling. J. Volcanol. Geother. Res. 207(3-4):93-105. https://doi.org/10.1016/j.jvolgeores.2011.07.008

Stissi, S. C., Napoli, R., Currenti, G., Afanasyev, A., Montegrossi, G. (2021). Influence of permeability on the hydrothermal system at Vulcano Island (Italy): inferences from numerical simulations. Accepted to Earth, Planets and Space

Troiano, A., Di Giuseppe, M., Petrillo, Z., Troise, C., and De Natale, G. (2011). Ground deformation at calderas driven by fluid injection: modelling unrest episodes at Campi Flegrei (Italy). Geophys. J. Int. 187(2):833-847. https://doi.org/10.1111/j.1365-246X.2011.05149.x

Yang, X. M., Davis, P. M., and Dieterich, J. H. (1988). Deformation from inflation of a dipping finite prolate spheroid in an elastic half-space as a model for volcanic stressing. Journal of Geophysical Research: Solid Earth, 93(B5), 4249-4257. https://doi.org/10.1029/JB093iB05p04249