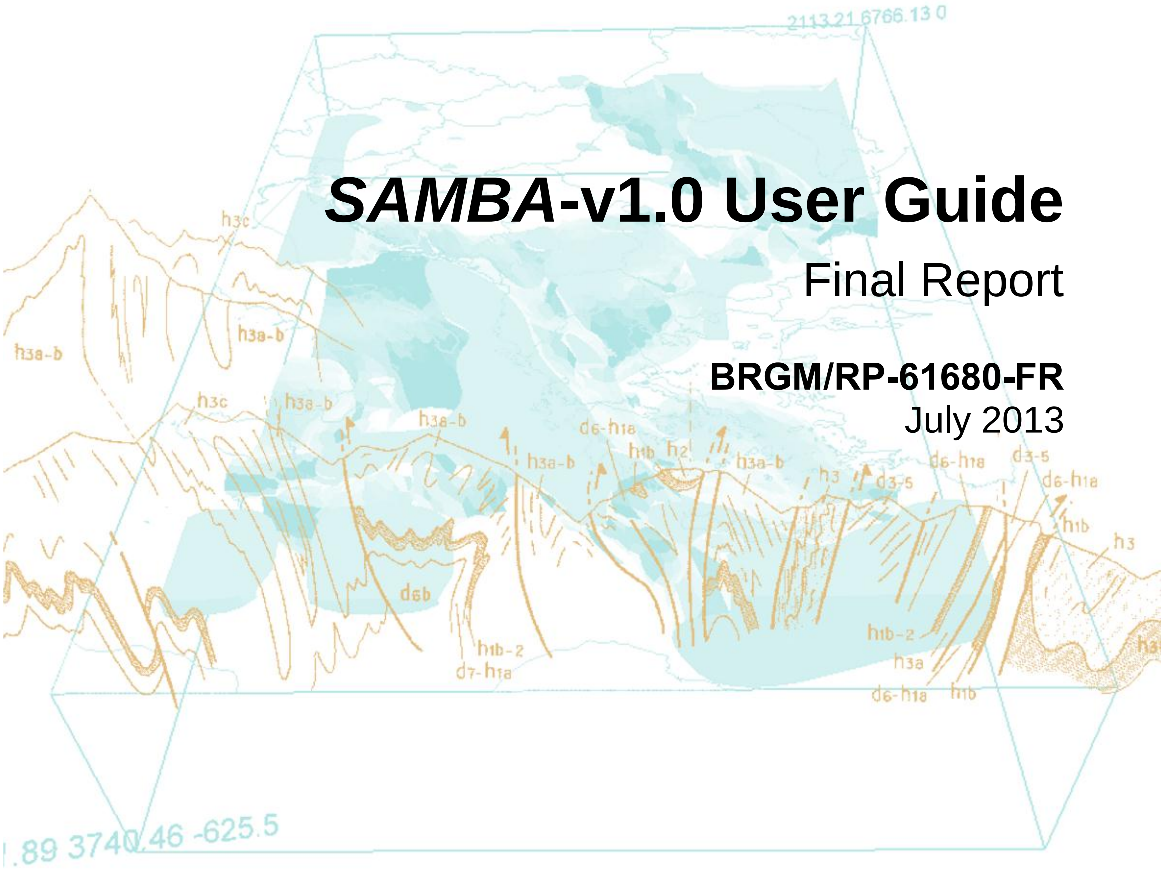


SAMBA-v1.0 User Guide

Final Report

BRGM/RP-61680-FR
July 2013

Geoscience for a sustainable Earth
brgm

SAMBA- v1.0 User Guide

Final Report

BRGM/RP-61680-FR
July 2013

Study carried out as part of
BRGM Research project on CO_2 storage risks management

Arnaud Réveillère

Checked by:	**Approved by**:
Name: J.C. Manceau	Name: H. Fabriol
Date: 17/12/2012	Date: 07/01/2014
Signature:	Signature:

If the present report has not been signed in its digital form, a signed original
of this document will be available at the information and documentation Unit (STI).

The quality management system of BRGM is certified according to AFAQ ISO 9001:2008.

Geoscience for a sustainable Earth
brgm

Keywords:geological storage, brine, leakage, semi-analytical model, deep aquifer, density effect

In bibliography, this report should be cited as follows:
Réveillère, A. (2013) *SAMBA* v1.0 – User Guide. BRGM Report RP-61680-FR, 64 p., 14 fig.

Synopsis

This document is intended to:

- Define the problem of brine leakage between connected aquifers taking into account the effect of brine density difference (Section 1).
- Recall the approximate solution to this problem proposed in Réveillère, 2013 (Section 2).
- Present the semi-analytical method used for computing this solution (Section 3).
- Detail the input parameters of the functions available in the *SAMBA* code, which is the implementation of the semi-analytic method presented in Section 3. *SAMBA* is a collection of Python functions grouped into the 3 modules (BrineProperties.py, BrinePropertiesLinearized.py, LeakageModels.py) plus an optional one (TOUGH2PreAndPostProcessing.py). It requires Python-2.7.1, numpy-1.6.0, scipy-0.9.0, and is *a priori* compatible with later versions (Section 4).
- Present examples of results obtained using the *SAMBA* code. These examples consist in scripts describing a problem and calling the *SAMBA* functions. For plotting the results, Matplotlib-1.0.1 (or later) is required (Section 5).

Contents

List of illustrations

1. Introduction and problem definition

As presented in Réveillère (2013), the injection of waste fluid or CO_2 in deep saline aquifers will create a pressurization that tends to displace brine upwards if a connection (e.g. an abandoned well) exists, raising issues of shallow potable water pollution by saline brine intrusion. However, this upwards displacement of brine is countered by the progressive weight increase of the fluid filling the leak that occurs when the dense lifting brine from the deep saline aquifer replaces less saline brine initially in the leak.

This effect should be included in models when tackling this issue. Réveillère (2013) proposes a semi-analytical model that accounts for this phenomenon. In the following of this section, we recall the model.

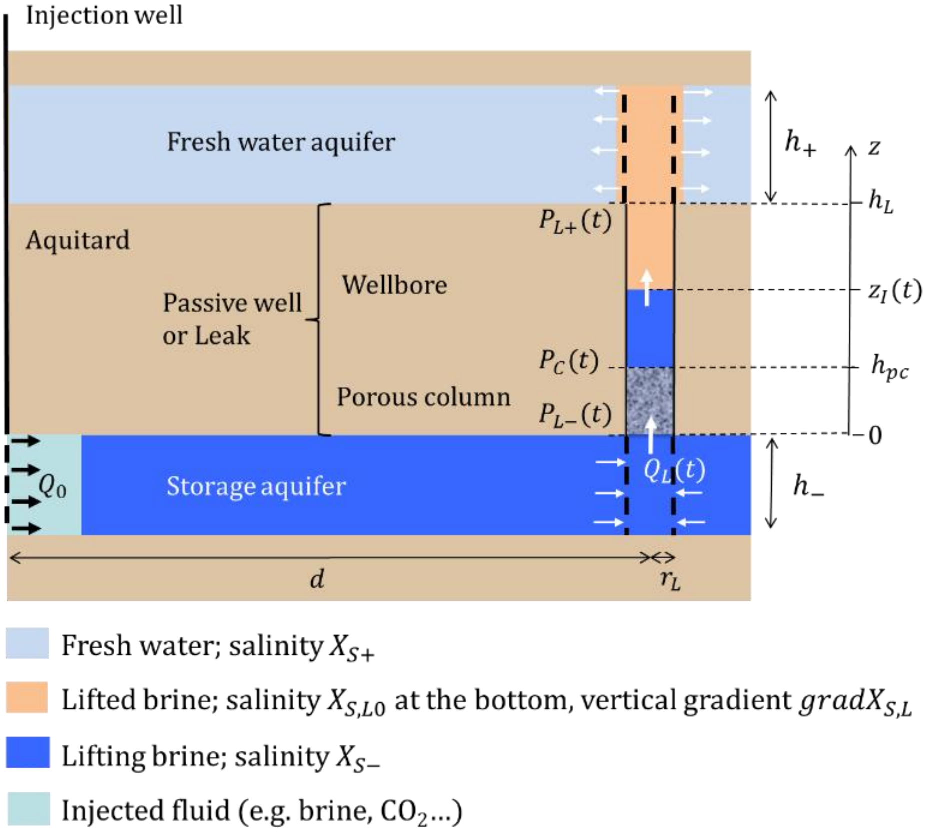


Figure 1: Schematic diagram of the leakage model. Typically, the lifting brine from the storage aquifer is a dense saline solution; the drinking water aquifer has very low salinity. The lifted brine initially fills the leak (porous column and wellbore). It may have either constant salinity or a vertical salinity gradient.

We model the subsurface as a stack of alternating permeable layers (aquifers) and impermeable layers (aquitards), which is typical of sedimentary basins. We assume that all aquifers are homogeneous, horizontal and have infinite area and constant thickness. One of the deep saline aquifers is used for geological storage (CO_2 or liquid waste) with a constant volumetric flow rate Q_0 injected from an active injection well at a location $\boldsymbol{x_I}$, from time $t_0 = 0$ to t_{inj}. This aquifer is connected to an overlying aquifer via a passive pathway (e.g., abandoned well) of radius r_L, referred to from now onwards as the leak, at location $\boldsymbol{x_L}$. The distance between the injection and the leak is termed $d = \|\boldsymbol{x_I} - \boldsymbol{x_L}\|$. The leak is composed of a porous column of length h_{pc}, where Darcy's law is applicable (representing, for instance,a degraded cement plug or annular cement), under an open wellbore. Length of both parts can vary, so the leak may represent a pure wellbore (as modeled in the static approaches), a pure porous column (similar to the connection model used in Nordbotten et al., 2004) or a combination of both.

Initially, the system is assumed to be at hydrostatic equilibrium. When injection starts at $t_0 = 0$, the pressure builds up in the storage aquifer. As this occurs, the increase in pressure under the leak P_{L-} drives an upward flow through the leak, the fluid from the storage aquifer lifting and replacing the fluid initially present in the leak (porous column and wellbore). The passive leakage well is therefore acting as a time-varying pumping well with a flow rate into the storage aquifer Q_L ($Q_L > 0$ for the upward flow), and time-varying injection into the overlying aquifer at a rate $\frac{\rho_+}{\rho_-} Q_L$, $\frac{\rho_+}{\rho_-}$ being the ratio of top aquifer density to bottom aquifer density. When injection stops, at t_{inj}, the overpressure in the storage aquifer decreases towards its long term equilibrium value, 0 in the case in point (infinite aquifer). The pressure of the bottom aquifer will therefore not be sufficient to withstand the increased weight of the column of fluid filling the leak (after replacement of the light, lifted brine by the denser lifting brine), and downward flow will start and will continue until return to the initial hydrostatic equilibrium.

During flow in the leak, the lifted brine (initially filling the leak) is lifted without change in its salinity, and its temperature either equilibrates instantaneously with its surroundings (Thermal equilibrium case), or flows along the leak at its constant initial temperature (Adiabatic flow case).

The model does not include molecular diffusion or convective mixing in the top aquifer; the brine pushed out from the leak during upward leakage therefore accumulates in the top aquifer in a cylindrical shaped plume centered on the top of the leak, the cylinder spreading out from its center during leakage. During the downward flow, the cylinder retracts from its center until it disappears. This determines the salinity of the brine entering the leak from the top.

2. Leakage model

2.1. PRESSURE EQUATIONS FOR THE AQUIFER

We consider a homogeneous aquifer of constant thickness and infinite extent, with injection at a constant volumetric flow rate $Q[\mathrm{L}^3\mathrm{T}^{-1}]$ starting at time $t = 0$ at location $\boldsymbol{x_I}$. The governing equation of the pressure P$[\mathrm{M\,L}^{-1}\mathrm{T}^{-2}]$ in the aquifer is (see e.g., De Marsily 1986):

$$S\frac{\partial P}{\partial t} - T\nabla^2 P = \rho g Q \delta(\boldsymbol{x} - \boldsymbol{x_I})$$

$$P(\boldsymbol{x}, t = 0) = P^0$$

Equation **1**

Where δ is the Dirac function, $t[T]$ the time, $x[L]$ the location, $T[\mathrm{L}^2\mathrm{T}^{-1}]$ the transmissivity and $S[-]$ the storativity, expressed by:

$T = \frac{k\rho g h}{\mu}$ Equation **2**

$S = \rho g \omega h(C_p + C_b)$ Equation **3**

$\omega[-]$, $k\,[L^2]$ and $h\,[L]$ are respectively the aquifer's porosity, permeability and thickness. $\mu\,[\mathrm{ML}^{-1}\mathrm{T}^{-1}]$ is the formation fluid's dynamic viscosity and $\rho[\mathrm{ML}^{-3}]$ its density. $g[\mathrm{L\,T}^{-2}]$ is the gravity constant andC_p and C_b are respectively pore and brine compressibility $[\mathrm{M}^{-1}\mathrm{LT}^2]$.

The solution of Equation 1 has been given by Theis (1935):

$$P(\boldsymbol{x}, t) = P^0 + \frac{Q\mu}{4\pi kh} E_1\left(\frac{\|\boldsymbol{x} - \boldsymbol{x_I}\|^2 S}{4Tt}\right)$$

Equation **4**

Therefore, Equation 4 expresses the pressure at any point of the aquifer for a constant injection (or pumping) volumetric flow rate. Since Equation 1 is linear, the pressure perturbation created by a time varying flow rate $Q(t)$ starting at t=0 at the location $\boldsymbol{x_L}$ can be deduced using the superposition principle, which introduces a convolution:

$$P(\boldsymbol{x}, t) = P^0 + \frac{\mu}{4\pi kh}\int_0^t \frac{dQ}{dt'} E_1\left(\frac{\|\boldsymbol{x} - \boldsymbol{x_L}\|^2 S}{4T(t - t')}\right) dt'$$

Equation **5**

These general equations Equation 4 and Equation 5 can be applied to the two connected aquifers system described in Section 1. We use the subscripts $_+$, $_-$ and $_L$ respectively to denote a property of the top aquifer, of the bottom aquifer and of the leak.The pressure above the leak P_{L+} is given by:

$$P_{L+}(t) = P_{L+}^0 + \frac{\mu_+}{4\pi k_+ h_+} \frac{\rho_+}{\rho_-} \int_0^t \frac{dQ_L}{dt'} E_1 \left(\frac{r_L^2 S_+}{4T_+(t-t')} \right) dt'$$

Equation **6**

The pressure under the leak P_{L-} is obtained using the superposition principle to consider the effects of the injection flow rate Q_0 and transient pumping Q_L (Equation 5). Before the end of injection at t_{inj}, the pressure at the leak is given by:

$$P_{L-}(t) = P_{L-}^0 + \frac{\mu_-}{4\pi k_- h_-} \left[Q_0 E_1 \left(\frac{d^2 S_-}{4T_- t} \right) - \int_0^t \frac{dQ_L}{dt'} E_1 \left(\frac{r_L^2 S_-}{4T_-(t-t')} \right) dt' \right]$$

Equation **7**

After the end of injection, at $t > t_{inj}$, the pressure field is obtained by considering a production flow rate Q_0 starting at t_{inj} at the same location as that where the injection started at $t = 0$. The pressure field is the sum of both:

$$\begin{aligned} P_{L-}(t) = P_{L-}^0 + \frac{\mu_-}{4\pi k_- h_-} \left[Q_0 \left[E_1 \left(\frac{d^2 S_-}{4T_- t} \right) - E_1 \left(\frac{d^2 S_-}{4T_-(t - t_{inj})} \right) \right] \right. \\ \left. - \int_0^t \frac{dQ_L}{dt'} E_1 \left(\frac{r_L^2 S_-}{4T_-(t-t')} \right) dt' \right] \end{aligned}$$

Equation **8**

We assume that brine viscosity is a property associated with the storage and overlying aquifers (resp. μ_- and μ_+), and we neglect the rather small effects of warmer or more saline lifting brine that may flow in the top aquifer.

2.2. BRINE PROPERTIES IN THE LEAK

Salinity, pressure, and temperature all increase with depth in sedimentary basins. With the two other parameters being constant, brine density increases with pressure or salinity increase, and decreases with temperature increase. Due to the low compressibility of water, the effect of temperature dominates that of pressure, and density therefore decreases with depth for a given constant salinity (see Figure 2). This effect is balanced in case of salinity increase with depth, and the density may even increase with depth for high salinity gradients.

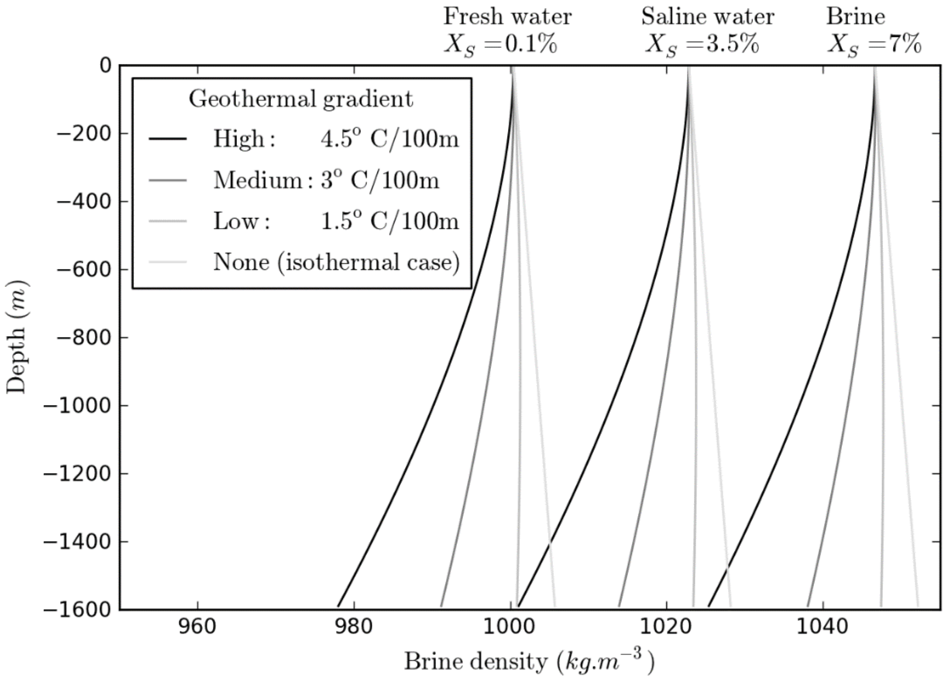


Figure 2: Brine density against depth for several cases of geothermal gradients and salt mass fractions X_S. A surface temperature of 10°C, pressure of 10^5 Pa and a hydrostatic pressure gradient are assumed.

Similarly to the static approach taken by Nicot et al. (2009), we use a linear approximation of these brine density profiles $\rho(z)$ for the region of interest, i.e., for a given geothermal gradient and for the appropriate salinity and pressure conditions. We introduce the parameters $\rho_{0,Xs}$ (density at depth $z = 0$) and ξas follows:

Equation **9** $$\rho(z) = \rho_{0,Xs} + \xi z$$

In the case of constant salinity X_S, ξ is written ξ_{Xs}, and in case of constant temperature and salinity, it is written $\xi_{T,Xs}$. We assume that ξ_{Xs} and ξ_{TXs} do not depend on the value of the salt mass fraction Xs (see Figure 2).

2.3. HYDROSTATIC EQUATIONS FOR THE LEAK

Figure 3 illustrates the initial linearized brine density profile in the leak and its evolution during thermal equilibrium or adiabatic leakage. Since we consider that there is no diffusion and no mass transfer between the inside and outside of the leak, the interface between lifting and lifted brines remains sharp and its position is termed z_I.

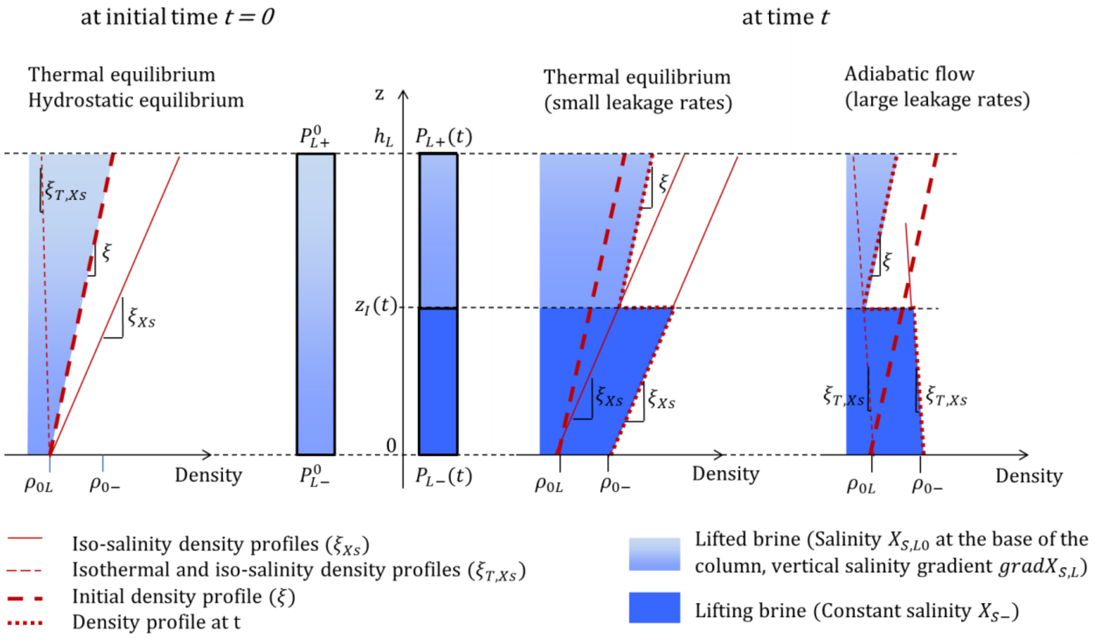


Figure 3: Schematic view of density profile evolution in the leak due to replacement of the lifted brine initially filling the leak by the lifting brine from the bottom aquifer.

We then express the hydrostatic pressure difference between the bottom and the top of the porous column ΔP^{static}_{pc} orof the wellbore ΔP^{static}_{wb}in terms of two components: a constant term corresponding to the initial situation (superscript ^{0}) and a first order term based on the density differences between the lifting and lifted brines (superscript ^{1}):

$$\Delta P^{static}_{pc} = \Delta P^0_{pc} + \Delta P^1_{pc}(z_I), \qquad with\Delta P^0_{pc} = P^0_{L-} - P^0_C$$

$$\Delta P^{static}_{wb} = \Delta P^0_{wb} + \Delta P^1_{wb}(z_I), \qquad with\Delta P^0_{wb} = P^0_C - P^0_{L+}$$

Equation **10**

The first order terms are deduced from the integration of Equation 9:

$$\Delta P_{pc}^{1}(z_I) = \begin{cases} \Delta\rho_0 g z_I + \Delta\xi g z_I \left(h_{pc} - \frac{z_I}{2}\right) & for z_I \leq h_{pc} \\ \Delta\rho_0 g h_{pc} + \Delta\xi g \frac{{h_{pc}}^2}{2} & for h_{pc} < z_I \leq h_L \end{cases}$$

$$\Delta P_{wb}^{1}(z_I) = \begin{cases} 0 & for z_I \leq h_{pc} \\ \Delta\rho_0 g\left(z_I - h_{pc}\right) + \Delta\xi g\left(z_I - h_{pc}\right)\left(h_L - \frac{z_I - h_{pc}}{2}\right) & for h_{pc} < z_I \leq h_L \end{cases}$$

$$with \Delta\rho_0 = \rho_{0-} - \rho_{0L}; \quad \Delta\xi = \begin{cases} \xi - \xi_{XS} for thermal equilibrium flow \\ \xi - \xi_{TXS} for adiabatic flow \end{cases}$$

Equation **11**

Where $\rho_{0-}\mathrm{and}\rho_{0\mathrm{L}}$ are the densities at z = 0 of the lifting and lifted brines respectively. $g\ [LT^{-2}]$ is the acceleration due to gravity.

Once the lifting brine has reached the top aquifer, the first order term no longer evolves since the porous column and/or the wellbore are completely filled with the lifting brine from the deep reservoir. The density may still change due to pressure variations in the column, but this brine compressibility effect is negligible compared to its thermal expansion and variation due to salinity.

2.4. LEAKAGE FLOW IN THE POROUS COLUMN

We apply a macroscopic statement of Darcy's law to the porous column, assuming that the well is vertical and has a constant circular section of radius r_{L}:

$$Q_L = \pi r_L^2 \frac{k_{pc}}{\mu_{pc}(z_I) h_{pc}} \left(P_{L-} - P_C - \Delta P_{pc}^0 - \Delta P_{pc}^1(z_I)\right)$$

Equation **12**

The gravity terms ΔP_{pc}^0 and ΔP_{pc}^1 have been presented in the previous section. $\mu_{pc}(z_I(t))$ is the reference viscosity of the brine at the conditions of temperature, pressure and salinity in the leak at time t.

2.5. LEAKAGE FLOW IN AN OPEN WELLBORE

Above the plug, the open wellbore is modeled as the corroded tubing of an abandoned well. The pressure gradient in a flowing well can be represented by a superposition of gravity, frictional, and acceleration gradients.

The gravity term corresponds to the hydrostatic equations in the wellbore (Equation 11). The other two terms are null in absence of flow, and are significant for large leakage rates only. We can therefore assume that the flowing brine is at the temperature of the bottom aquifer (adiabatic leakage case); its density and viscosity are therefore those of the bottom aquifer, respectively ρ_- and μ_-. The pressure loss due to friction forces $\Delta P_{wb}^{fric}$is given by the Darcy-Weisbach equation:

$$\Delta P_{wb}^{fric} = \frac{\lambda(Q_L)\rho_{0-}}{4}\frac{h_{wb}}{r_L}\left(\frac{Q_L}{\pi r_L^2}\right)^2$$

Equation **13**

λ is the friction factor and depends on the Reynolds number $Re = \frac{2Q_L\rho_{0-}}{\pi r_L \mu_-}$, which characterizes the flow regime. It satisfies the following equations:

$$\lambda = \frac{64}{Re} for Re < 2500$$

$$\frac{1}{\sqrt{\lambda}} = -0.87\ \log\left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{0.005}{7.42\ r_L}\right) for Re \geq 2500$$

Equation **14**

The equation for turbulent flow Re>2500, known as the “Colebrook equation”, is based on experimental data and is given here for very highly corroded tubings (roughness height of $5\ mm$). As it appears in Equation 13, the pressure loss forces are proportional to Q_L^2 and to r_L^{-5}. They are therefore significant for large leakage rates (i.e., wellbore only, no porous column) and small tubing diameters only.

The acceleration term is negligible. The pressure difference in the wellbore is therefore given by:

$$P_C - P_{L+} = \Delta P_{wb}^0 + \Delta P_{wb}^1(z_I) + sign(Q_L)\Delta P_{wb}^{fric}$$

Equation **15**

2.6. CONSERVATION OF THE LEAKING MASS

We deduce the height of the interface by assuming that the mass of reservoir brine that has leaked is equal to the mass of reservoir brine in the leak, using linear approximation of the density (Equation 9):

$$\int_o^t Q_L(t')\,dt' = \pi r_L^2 \cdot \begin{cases} \omega_{pc} z_I + \dfrac{\xi_{Xs}\omega_{pc}{z_I}^2}{2\rho_{0-}} & for z_{L-} \leq\ z_I \leq z_C \\ \omega_{pc} h_{pc} + \omega_{pc}\dfrac{\xi_{Xs}{h_{pc}}^2}{2\rho_{0-}} + z_I - h_{pc} + \dfrac{\xi_{Xs}}{2\rho_{0-}}\left(z_I - h_{pc}\right)^2 & for z_C < z_I \leq\ z_{L+} \end{cases}$$

Equation **16**

This equation is given for the thermal equilibrium leakage case. For adiabatic flow, ξ_{Xs} should be replaced by ξ_{TXs}.

3. Leakage solution

3.1. APPROXIMATION OF THE CONVOLUTION PRODUCT FOR A MONOTONOUS LEAKAGE RATE

The convolution products in equations 6 and 7 would normally need to be solved in the Laplace domain, numerically or analytically for simple problems. Alternatively, Nordbotten et al. (2004) propose to approximate the leakage history Q by $\bar{Q}$, a step function changing from 0 to $Q(t)$ at time $[1-\gamma]t$:

$$\bar{Q}(t') = H(t' - [1-\gamma]t)Q(t)$$

Equation **17**

H is the Heaviside step function. The authors use a constant value γ=0.92 for a case of brine leakage through a porous column by comparing this approximate solution to a case solved exactly analytically in the Laplace domain. Similarly, Nordbotten et al. (2005) use two constant parameters for successive leakage of brine and CO_2. In this study, we propose use of a fluctuating $\gamma(t)$ parameter set in order to ensure that the approximate leakage $\bar{Q}$ and the real leakage Q have the same integrals (i.e., the same leakage volume), as presented in Figure 4. The parameter is therefore set by:

$$\int_0^t Q(t')dt' = \gamma(t)t\bar{Q}(t)$$

Equation **18**

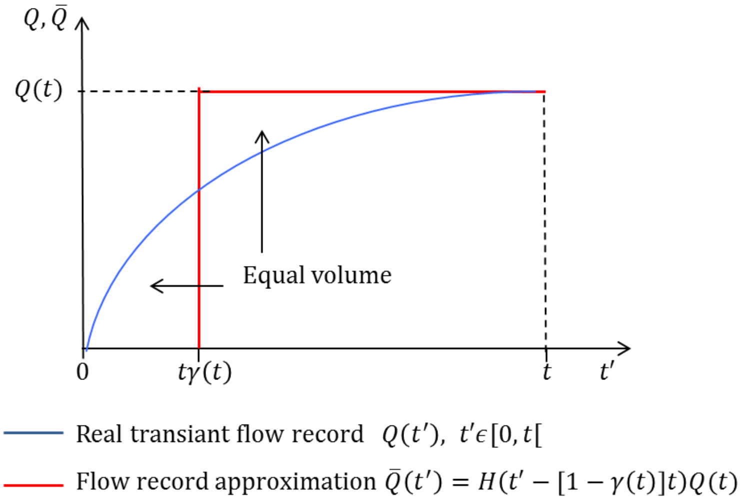

$Q, \bar{Q}$
$Q(t)$
Equal volume
0
$t\gamma(t)$
t
t'
Real transiant flow record $Q(t'),\ t'\epsilon[0,t[$
Flow record approximation $\bar{Q}(t') = H(t' - [1-\gamma(t)]t)Q(t)$

Figure 4: Real leakage record Q and its approximation $\bar{Q}$ introduced in Nordbotten et al., 2004 and modified by using a time-varying $\gamma(t)$ parameter set in order to ensure that both the integrals (i.e., the leakage volume) are equal.

3.2. APPROXIMATION OF NON-MONOTONIC LEAKAGES

As stated in Nordbotten et al. (2005), this approximation works very well when the leakage has a self-similar form. This not the case for leakage rates when the effects of salinity and density differences between lifted and lifting brines are included. For instance, the numerical simulations by Birkholzer et al. (2011) present results where leakage rates increase, reach a maximum and decrease due to the greater weight of the column of fluid in the leak. The leakage is therefore clearly not self-similar, and the leakage approximation presented in Section 3.1 cannot apply.

We therefore propose to develop non-monotonic leakages into series of monotonic functions, a function being added at every inflexion point. For a piecewise monotonic leakage rate that admits, from time 0 to t, m+2 local extrema at times [t_0=0, t_1,…, t_k,…, t_m, t], we construct m+1 strictly monotonic functions $Q_{L,k}, k \in [0,m]$, whose sum equalsQ_L. $Q_{L,k}$ is a continuous function, null from t_0=0 to t_k, varying monotonously from t_k to t_{k+1}, and constant after t_{k+1}, as presented in Figure 5. Using this construction, the convolution $\frac{\partial Q_L}{\partial t} * W$ can be written:

$$\int_0^t \frac{\partial Q_L(t')}{\partial t'} W(t-t')dt' = \sum_{k=0}^{m} \int_0^t \frac{\partial Q_{L,k}(t')}{\partial t'} W(t-t')dt'$$

Equation **19**

We apply the step function approximation for monotonic functions as described in Section 3.1 to each of these functions $Q_{L,k}$. The convolution $\frac{\partial Q_{L,k}}{\partial t} * W$ can then be approximated by an algebraic expression as follows:

$$\begin{aligned}\int_0^t \frac{\partial \overline{Q_{L,k}}(t')}{\partial t'} W(t-t')dt' &= \int_0^t \frac{\partial H(t'-[1-\gamma_k(t)]t)Q_{L,k}(t)}{\partial t'} W(t-t')\,dt' \\ &= Q_{L,k}(t)\int_0^t \delta([1-\gamma_k(t)]t)W(t-t')\,dt' \\ &= Q_{L,k}(t)\cdot W(\gamma_k(t)\cdot[t-t_k])\end{aligned}$$

Equation **20**

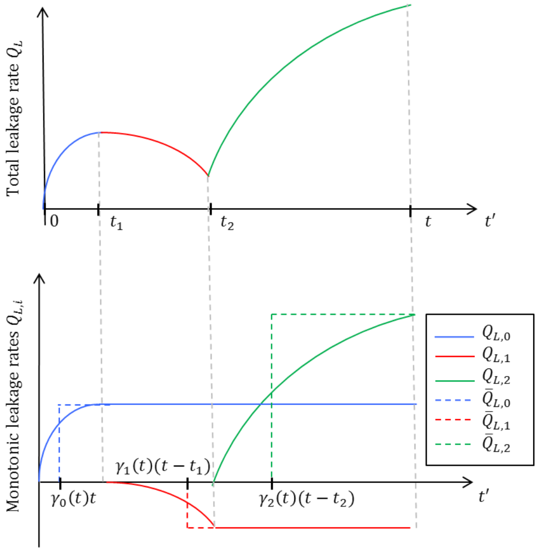


Figure 5: Top panel: schematic representation of a leakage rate function Q_L that admits 4 local extrema at times 0, t_1, t_2 and t. Bottom panel: construction of monotonic functions Q_{Lk}, whose sum equals Q_L. Each Q_{Lk} record is approximated by a Heaviside step function $\overline{Q_{L,k}}$, as presented in Section 3.1.

To sum up, the convolution products with time-varying leakage rates Q_L are evaluated as follows:

- $Q_L$is broken down into a series of self-similar monotonic functions $Q_{L,k}$
- Every $Q_{L,k}$ function is approximated by a Heaviside step function, which uses a dephasing parameter γ_k. Every γ_k is time-varying and is evaluated using Equation 18.

Below, we no longer use the notation $\bar{Q}$. We apply this approximation to Equation 6, Equation 7 and Equation 8, which leads respectively to Equation 21, Equation 22 ($fort \leq t_{inj}$) and Equation 23$fort > t_{inj}$.

$$P_{L+}(t) = P_{L+}^{0} + \frac{\mu_+}{4\pi k_+ h_+}\frac{\rho_+}{\rho_-}\sum_{k=0}^{m} Q_{L,k}\,(t) E_1\left(\frac{r_L^2 S_+}{4T_+\gamma_k(t)[t-t_k]}\right)$$

Equation **21**

$$P_{L-}(t) = P_{L-}^{0} + \frac{\mu_-}{4\pi k_- h_-}\left[Q_0 E_1\left(\frac{d^2 S_-}{4T_- t}\right) - \sum_{k=0}^{m} Q_{L,k}\,(t) E_1\left(\frac{r_L^2 S_-}{4T_-\gamma_k(t)[t-t_k]}\right)\right]$$

Equation **22**

$$P_{L-} = P_{L-}^{0} + \frac{\mu_-}{4\pi k_- h_-}\left[Q_0\left[E_1\left(\frac{d^2 S_-}{4T_- t}\right) - E_1\left(\frac{d^2 S_-}{4T_-(t-t_{inj})}\right)\right] - \sum_{k=0}^{m} Q_{L,k}\, E_1\left(\frac{r_L^2 S_-}{4T_-\gamma_k(t)[t-t_k]}\right)\right]$$

Equation **23**

3.3. DIMENSIONLESS FORMULATION OF THE PROBLEM AND SOLUTION

The problem described above can be simplified by considering the dimensionless groupings or functions in Table 2. In the table, the formulations for depth are generic: z can be replaced by any depth or length, e.g., $\mathrm{z_I}, \mathrm{h_{pc}}$.The formulation for pressures is also generic, P can be replaced by$P_{L-}^0, P_{L+}^0, \Delta P_{cp}^0, \Delta P_{wb}^0$, ΔP_{cp}^1, ΔP_{wb}^1. Also note that time is not dimensionless since that does not simplify any equation.

$\widetilde{Q_L} = \frac{Q_L}{Q_0}$	$\tilde{z} = \frac{z}{h_L}$	$\tilde{P} = \frac{4\pi h_- k_- P}{Q_0 \mu_-}$
$\widetilde{\Delta\rho_0} = \beta \frac{\Delta\rho_0}{\rho_{0-}}$	$\widetilde{\Delta\xi} = \beta \frac{\Delta\xi h_L}{\rho_{0-}}$	$\beta = \frac{4\pi h_- k g \rho_{0-} h_L}{Q_0 \mu_-}$
$\tau\ [T] = \frac{\pi r_L^2 h_L}{Q_0}$		$\varphi = \frac{h_L}{2\rho_{0-}} \cdot \begin{cases} \xi_{Xs} for thermal eq. flow \\ \xi_{TXs} for adiabatic flow \end{cases}$
$\widetilde{\mu_{pc}} = \frac{\mu_{pc}}{\mu_-}$	$\kappa = \frac{\mu_- r_L^2 k_{pc}}{4\mu_{pc} h_- k_- h_{pc}}$	$\Lambda = sign(Q_L)\lambda \frac{h_- k_- \rho_- h_L (1-\delta) Q_0}{\pi \mu_- r_L^5}$
$W_0(t) = E_1\left(\frac{d^2 S_-}{4T_- t}\right)$	$W_{L-}(t) = E_1\left(\frac{r^2 S_-}{4T_- t}\right)$	$W_{L+}(t) = \frac{\mu_+ \rho_+ h_- k_-}{\mu_- \rho_- h_+ k_+} E_1\left(\frac{r^2 S_-}{4T_- t}\right)$

Table 1: Definition of groupings and functions (dimensionless except time)

The following table presents the dimensionless formulations of the equations describing the problem.

Equation number	Dimensionless formulation	Dimensionless eq. number
Equation 10	$\widetilde{P_{L-}^{0}} = \widetilde{P_{L+}^{0}} + \widetilde{\mathit{\Delta} P_{cp}^{0}} + \widetilde{\mathit{\Delta} P_{wb}^{0}}$	Equation 24
Equation 11	$\widetilde{\Delta\, P_{pc}^{1}} = \begin{cases} \widetilde{\Delta\,\rho_0}\widetilde{z_I} + \widetilde{\Delta\xi}\widetilde{z_I}\left(\widetilde{h_{pc}} - \frac{\widetilde{z_I}}{2}\right) & for\ 0 \le \widetilde{z_I} \le \widetilde{h_{pc}} \\ \widetilde{\Delta\,\rho_0}\ \delta\ + \widetilde{\Delta\xi}\,\frac{\widetilde{h_{pc}}^{2}}{2} & for \widetilde{h_{pc}} < \widetilde{z_I} \le 1 \end{cases}$ $\widetilde{\Delta\, P_{wb}^{1}} = \begin{cases} 0 & for\ \ 0 \le \widetilde{z_I} \le \widetilde{h_{pc}} \\ \widetilde{\Delta\,\rho_0}\left(\widetilde{z_I} - \widetilde{h_{pc}}\right) + \widetilde{\Delta\xi}\left(\widetilde{z_I} - \widetilde{h_{pc}}\right)\left(1 - \frac{\widetilde{z_I} - \widetilde{h_{pc}}}{2}\right) & for \widetilde{h_{pc}} < \widetilde{z_I} \le 1 \end{cases}$	Equation 25
Equation 12	$\widetilde{Q_L} = \frac{\kappa}{\widetilde{\mu_{pc}}}\left(\widetilde{P_{L-}} - \widetilde{P_C} - \widetilde{\mathit{\Delta} P_{cp}^{0}} - \widetilde{\mathit{\Delta} P_{cp}^{1}}\right)$	Equation 26
Equation 13 Equation 15	$\widetilde{P_C} = \widetilde{P_{L+}} + \widetilde{\mathit{\Delta} P_{wb}^{0}} + \widetilde{\mathit{\Delta} P_{wb}^{1}} +\ \Lambda\,\widetilde{Q_L}^{2}$	Equation 27
Equation 16	$\int_o^t \widetilde{Q_L}(t')\frac{dt'}{\tau} = \begin{cases} \omega_C\widetilde{z_I}(1 +\ \phi\,\widetilde{z_I}) & for\ 0 \le \widetilde{z_I} \le \widetilde{h_{pc}} \\ \omega_C\widetilde{h_{pc}}\left(1 +\ \phi\,\widetilde{h_{pc}}\right) + \left(1 +\ \phi\left(\widetilde{z_I} + \widetilde{h_{pc}}\right)\right)\left(\widetilde{z_I} - \widetilde{h_{pc}}\right) & for \widetilde{h_{pc}} < \widetilde{z_I} \le 1 \end{cases}$	Equation 28
Equation 21	$\widetilde{\mathrm{P}_{\mathrm{L+}}}(\mathrm{t}) = \widetilde{\mathrm{P}_{\mathrm{L+}}^{0}} + \sum_{k=0}^{m} \widetilde{\mathrm{Q}_{\mathrm{L,k}}}(\mathrm{t})\mathrm{W}_{L+}\left(\gamma_{\mathrm{k}}(\mathrm{t})[t - t_k]\right)$	Equation 29
Equation 22, Equation 23	$\widetilde{P_{L-}}(\mathrm{t}) = \widetilde{P_{L-}^{0}} - \sum_{k=0}^{m} \widetilde{\mathrm{Q}_{\mathrm{L,k}}}(\mathrm{t})\mathrm{W}_{L-}\left(\gamma_{\mathrm{k}}(\mathrm{t})[t - t_k]\right) + \begin{cases} W_0(t) fort \le t_{inj} \\ W_0(t) - W_0\left(t - t_{inj}\right) fort > t_{inj} \end{cases}$	Equation 30
Equation 18	$\gamma_k(t) = \frac{\int_0^t \widetilde{Q_{L,k}}(t')dt'}{t\widetilde{Q_{L,k}}(t)} forkin[0, m]$	Equation 31

Table 2: Dimensionless formulation of the problem

4. *SAMBA* semi-analytic resolution

This resolution is based on the dimensionless formulation of the problem obtained in the previous section.

We note that according to the construction of the leakage rates monotonic components, the only one which is varying at that time $t \in [t_i,\ t_{i+1}[, i \leq m$ is $\widetilde{\mathrm{Q}_{\mathrm{L},}}_i$. It can besolved in the linear system made of Equation **24**, Equation 26, Equation 27, Equation 29 and Equation 30:

$$\widetilde{\mathrm{Q}_{\mathrm{L},}}_i = \frac{\mathrm{W}_0 - \sum_{k=0}^{i-1} \widetilde{\mathrm{Q}_{\mathrm{L,k}}}\left[\tilde{\mu}/\kappa + \mathrm{W}_{L-}\left(\gamma_{\mathrm{k}}[t - t_k]\right) + \mathrm{W}_{L+}\left(\gamma_{\mathrm{k}}[t - t_k]\right)\right] - \widetilde{\Delta \mathrm{P}^1_{\mathrm{wb}}} - \widetilde{\Delta \mathrm{P}^1_{\mathrm{cp}}} - \Lambda\left(\widetilde{\mathrm{Q}_{\mathrm{L}}}\right)\widetilde{\mathrm{Q}_{\mathrm{L}}}^2}{\tilde{\mu}/\kappa + \ \mathrm{W}_{L-}\left(\gamma_i[t - t_i]\right) + \mathrm{W}_{L+}\left(\gamma_m[t - t_i]\right)}$$

Equation **32**

Note that in this expression,$\widetilde{\mathrm{Q}_{\mathrm{L},}}_i$ depends on itself through the dependence on $\widetilde{\mathrm{Q}_{\mathrm{L}}} = \sum_{\mathrm{k}=0}^{\mathrm{i}} \widetilde{\mathrm{Q}_{\mathrm{L,k}}}$, and that Λ also depends on $\widetilde{\mathrm{Q}_{\mathrm{L}}}$. This dependency corresponds to tubing pressure losses and it is treated in an explicit manner in the present time discretization, i.e. tubing pressure losses are calculated based on the flow rate of the previous time step. It will therefore work well as long as tubing pressure losses are not the dominant phenomena countering the leakage.

In order to solve the leakage rate time evolution, we have to solve equations 19, 21, 22 and 24 over time, which is done using a time discretization. However, the model remains analytical for the space dimensions, it does not use any spatial grid and the computation is therefore almost immediate.

Initially, at time $t^0 = 0$, we set:$\widetilde{Q_L}^0 = \widetilde{z_I}^0 = \widetilde{\Delta P^1_{pc}}^0 = \widetilde{\Delta P^1_{wb}}^0 = \gamma^0_k = 0$; $\tilde{\mu}^0 = \tilde{\mu}_{lifted}$

From the state of the system at time t^n, we successively compute the 3 following steps in order to deduce the state of the system at time t^{n+1}:

- **1st step: Computation of the total leakage rate**

First, we note that according the construction of the leakage rates monotonic components, the only one which is varying at that time $t^{n+1} \in [t_i,\ t_{i+1}[$ is $\widetilde{\mathrm{Q}_{\mathrm{L},}}_i$. Others are therefore immediately computed:

$$\widetilde{\mathrm{Q}_{\mathrm{L,k}}}^{n+1} = \widetilde{\mathrm{Q}_{\mathrm{L,k}}}^{n} for\ k \in [0, m]\ and\ k \neq i$$

The varying component at $t^{n+1}$is computed using Equation **32**:

$$\widetilde{\mathrm{Q}_{\mathrm{L},_{i}}}^{n+1} = \frac{\mathrm{W}_0(t^n) - \widetilde{\Delta\mathrm{P}^1_{\mathrm{wb}}}^{n} - \widetilde{\Delta\mathrm{P}^1_{\mathrm{cp}}}^{n} - \Lambda\left(\widetilde{\mathrm{Q}_{\mathrm{L}}}^{n}\right)\widetilde{\mathrm{Q}_{\mathrm{L}}}^{n^2} - \sum_{k=0}^{i-1}\widetilde{\mathrm{Q}_{\mathrm{L,k}}}^{n}[\tilde{\mu}^n/\kappa + \mathrm{W}_{L-}(\gamma_k^n[t^{n+1}-t_k]) + \mathrm{W}_{L+}(\gamma_k^n[t^{n+1}-t_k])]}{\tilde{\mu}^n/\kappa + \mathrm{W}_{L-}\left(\gamma_i^n[t^{n+1}-t_i]\right) + \mathrm{W}_{L+}\left(\gamma_i^n[t^{n+1}-t_i]\right)}$$

The total leakage is the sum of these monotonic components: $\widetilde{\mathrm{Q}_{\mathrm{L}}}^{n+1} = \sum_{k=0}^{m}\widetilde{\mathrm{Q}_{\mathrm{L,k}}}^{n+1}$

- **2nd step: Compute lifting-lifted brine interface position $\widetilde{\mathbf{z}_{\mathbf{I}}}^{\mathbf{n+1}}$**

The position of the interface is deduced from the sum of the leakage rate records from $t^0 = 0$ to t^{n+1}. The brine average viscosity in the porous column $\tilde{\mu}$ and pressure increases due to brine density differences in the leak, which are functions $\widetilde{\mathrm{z}_{\mathrm{I}}}$ only, are then updated.

- $for\ \widetilde{z_I}^{n+1} = 0$, i.e. when the lifting brine has started getting in the leak:
  - $\widetilde{\Delta\mathrm{P}^1_{\mathrm{cp}}}^{\mathrm{n+1}} = 0$
  - $\widetilde{\Delta P^1_{wb}}^{n+1} = 0$
  - $\tilde{\mu}^{n+1} = \tilde{\mu}_{lifted}$
- $for\ 0 < \widetilde{z_I}^{n+1} \leq \delta$, i.e. while the interface is in the porous column:
  - solve $\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}$ in:

$$\omega_{\mathrm{pc}}\sum_{\mathrm{j=0}}^{\mathrm{n+1}}\widetilde{\mathrm{Q}_{\mathrm{L}}}^{\mathrm{j}}\left(t^{\mathrm{j+1}} - t^j\right)/\tau = \widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}\left(1 + \varphi\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}\right)$$

  - $\widetilde{\Delta\mathrm{P}^1_{pc}}^{\mathrm{n+1}} = \widetilde{\Delta\rho_0}\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}} + \widetilde{\Delta\xi}\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}\left(\delta - \frac{\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}}{2}\right)$
  - $\widetilde{\Delta P^1_{wb}}^{n+1} = 0$
  - $\tilde{\mu}^{n+1} = \tilde{\mu}_{lifting} \cdot \frac{\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}}{\delta} + \tilde{\mu}_{lifted} \cdot \left(1 - \frac{\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}}{\delta}\right)$
- $for\ \delta < \widetilde{z_I}^{n+1} < 1$, i.e. while the interface is in the open wellbore
  - solve $\widetilde{\mathrm{z}_{\mathrm{I}}}^{\mathrm{n+1}}$ in:

$$\sum_{j=0}^{n+1} \widetilde{Q_L}^{j} \left(t^{j+1} - t^{j}\right)/\tau = \widetilde{z_I}^{n+1} + \frac{\delta}{\omega_{pc}} - \delta + \varphi\left(\widetilde{z_I}^{n+1} + \frac{\delta}{\omega_{pc}} - \delta\right)^2$$

- ○ $\widetilde{\Delta P^1_{pc}}^{n+1} = \widetilde{\Delta P^{1,bk}_{pc}}$
- ○ $\widetilde{\Delta P^1_{wb}}^{n+1} = \widetilde{\Delta\rho_0}\left(\widetilde{z_I}^{n+1} - \delta\right) + \Delta\xi\left(\widetilde{z_I}^{n+1} - \delta\right)\left(1 - \frac{\widetilde{z_I}^{n+1} - \delta}{2}\right)$
- ○ $\tilde{\mu}^{n+1} = \tilde{\mu}_{lifting}$

- $for\ \widetilde{z_I}^{n} = 1$, i.e. when lifting brine has reached the top aquifer
  - ○ $\widetilde{\Delta P^1_{cp}}^{n+1} = \widetilde{\Delta P^{1,bk}_{cp}}$
  - ○ $\widetilde{\Delta P^1_{wb}}^{n+1} = \widetilde{\Delta P^{1,bk}_{wb}}$
  - ○ $\tilde{\mu}^{n+1} = \tilde{\mu}_{lifting}$

- **3rdstep: Update the γ parameter of the approximated evaluation of the convolution integrals**

This is done for each leakage rate monotonic component.

$$\gamma_k^{n+1} = \frac{\sum_0^{n+1}\left(t^{i+1} - t^{i}\right)\widetilde{Q_L}^{i}}{\left(t^{n+1} - t_k\right)\widetilde{Q_{L,k}}^{n+1}}\ for\ k \in [0, i]$$

$$\gamma_k^{n+1} = 0\ for\ k \in [i + 1, m]$$

- **Compute addition information (optional)**

The pressure at different positions in the leak can then be post-computed if needed:

- The pressure at the top of the leak:

$$\widetilde{P_{L+}} = \widetilde{P^0_{L+}} + \sum_{k=0}^{m} \widetilde{Q_{L,k}}(t) \cdot W_{L+}(\gamma_k(t)[t - t_k])$$

- The pressure in between the porous column and the open wellbore:

$$\widetilde{P_C} = \widetilde{P_{L+}} + \widetilde{\Delta P^0_{wb}} + \widetilde{\Delta P^1_{wb}}$$

- The pressure at the bottom of the leak:

$$\widetilde{P_{L-}} = \widetilde{P_{L-}^{0}} + W_0(t) - \sum_{k=0}^{m} \widetilde{Q_{L,k}}(t) \cdot W_{L-}(\gamma_k(t)[t - t_k])$$

The pressures driving the leakage flow and the ones countering it that appear in Equation **32** can similarly be post-computed if needed.

Note that in order to simplify the present explanation of the numerical scheme, the following elements have been intentionally omitted:

- Input variables of functions are not always written. The reader is referred to the first appearance of this function
- The *SAMBA* model implementation also enables to consider that a fraction of the lifting brine that has entered the top aquifer will not flow back down if the flow reverses (denoted *MixingCoef*). It results in a modification of Step 2.
- The construction of the monotonic functions is not detailed. Since this resolution is essentially using an explicit method, the leakage rate oscillates around the exact solution. The distinction of “real local maxima”, i.e. the ones due to flow regime change, from the ones due to these oscillations must therefore be made. Also note that major flow regime changes do not necessarily create a “real local maxima”: for instance in the case of an already slowly decreasing upwards leakage rate when the storage injection stops, the decrease pace gets much steeper without creating a new local maxima (see e.g. Réveillère, 2013, Figure 7 case $X_{S-} = 14\%$ at 5 years). A new monotonic component is therefore also created in that specific case, even if there is, strictly speaking, no local maxima.

5. Technical manual

This section presents the input parameters of the *SAMBA* Python functions. Input parameters written in blue are optional, and that their default value is mentioned in the description. The column named “symbol” in the parameters tables refers to a notation used in the presentation of the model (Sections 1 to 3 of this report).

5.1. BRINES PROPERTIES MODULE

This module has been developed as part of the CAMELOTPY code (Bandilla et al., 2011), which is available on code.google.com/camelotpy. It consists in the Python implementation of the brine density and viscosity laws presented respectively in sections 2.2 and 2.4 of the present report.

Readers are referred to the documentation provided on the CAMELOTPY website for additional information.

5.2. BRINE PROPERTIES LINEARIZED MODULE

This module essentially consists in the linearization of the brine properties from the Brine Properties module, or in the utilization of brine properties for deriving other parameters.

GetBrineDensityLinearFit(h, T0, gradT, P0, Xs0, gradXs, ReturnAll, discretization)

This command computes ρ_0 and ξ, the linearization parameters of the density of a column of brine as presented in Equation **9**. It is based on the following parameters:

Parameter	Symbol	Unit	Signification
h	h_L	m	Height of the brine column over which the fit is applied
T0		°C	Temperature at the bottom of the column of brine
gradT		°C.m^{-1}	Vertical temperature gradient (>0 for increasing temperature with depth)
P0		Pa	Pressure at the bottom of the column
Xs0	$X_{S,L0}$		Salt (NaCl) mass fraction at the bottom of the column
gradXs	$gradX_{S,L}$	m^{-1}	Salt mass fraction gradient in the column (>0 for increasing salinity with depth). Default = 0
ReturnAll		binary	If True, the function also returns the non-approximated density against depth. Default = True
Discretization		integer	Number of depth increments. Default = 100

GetAverageViscosity(h, T0, gradT, P0, Xs0, gradXs, ReturnAll, discretization)

This command computes the average viscosity in a vertical column of brine based on the following parameters:

Parameter	Symbol	Unit	Signification
h	h_{pc}	m	Height of the brine column over which the average is applied
T0		°C	Temperature at the bottom of the leak
gradT		$°C.m^{-1}$	Vertical temperature gradient (>0 for increasing temperature with depth)
P0		Pa	Pressure at the bottom of the leak
Xs0	$X_{S,L0}$		Salt (NaCl) mass fraction at the bottom of the leak
gradXs	$gradX_{S,L}$	m^{-1}	Salt mass fraction gradient in the leak (>0 for increasing salinity with depth). Default = 0
ReturnAll		binary	If True, the function also returns the viscosity against depth
Discretization		integer	Number of depth increments. Default = 100

GetBrineCompressibility(T, P, Xs, dP)

This command computes the brine compressibility based on the following parameters:

Parameter	Symbol	Unit	Signification
T		°C	Brine temperature
P		Pa	Brine pressure
Xs	X_S		Brine salt (NaCl) mass fraction
dP		Pa	Typical pressure increase. Defaut = 10^5 Pa

GetTSratio(w, k, Cr, mu, Cb)

This command computes the aquifer Transmissivity/Storability ratio based on the following parameters:

Parameter	Symbol	Unit	Signification
w	ω	$\in [0,1]$	Aquifer porosity
k		m^2	Aquifer permeability
Cr		Pa^{-1}	Pore (not rocks) compressibility
mu	μ	Pa.s	Brine viscosity
Cb		Pa^{-1}	Brine compressibility

GetTransmissivity(h, k, rho, mu, g)

This command computes the aquifer transmissivity based on the following parameters:

Parameter	Symbol	Unit	Signification
h		m	Aquifer thickness
k		m^2	Aquifer permeability
rho	ρ	$kg.m^{-3}$	Brine density
mu	μ	Pa.s	Brine viscosity
g		$m.s^{-2}$	Gravity acceleration. Default = 9.81 $m.s^{-2}$

GetStorability(h, w, k, rho, mu, g)

This command computes the aquifer storability based on the following parameters:

Parameter	Symbol	Unit	Signification
h		m	Aquifer thickness
w	ω	$\in [0,1]$	Aquifer porosity
k		m^2	Aquifer permeability
rho	ρ	$kg.m^{-3}$	Brine density
mu	μ	Pa.s	Brine viscosity
g		$m.s^{-2}$	Gravity acceleration. Default = 9.81 $m.s^{-2}$

5.3. LEAKAGE MODELS MODULE

The present module includes the leakage solution proposed in Réveillère (2013) and intermediate functions, as well as the solution proposed in Nordbotten et al. (2004).

```
LeakageSolution(h_b, w_b, k_b, Cr_b,                  # Bottom aquifer properties
h_t, w_t, k_t, Cr_t, MixingCoef,             # Top aquiferproperties
T_lb, gradT, P_lb,                           # T,P initial conditions
r_l, h_l,h_pc, w_pc, k_pc,               # leak properties
Xs_b, Xs_t, Xs0_lifted, gradXs_lifted# Brines salt mass fractions
d, Q0, t_inj, t_sim,                         # Injection parameters
IsThermalEq,
IsTubingPLossIncluded,
ComputePressures,
ComputeDrivingAndCounteringForces,
ExportInTextFile,
discretization)
```

This command computes the time (s), leakage flow rate (m^3/s) and position of the lifting/lifted brines interface (m). If ComputePressures is True, it also returns the pressures (in Pa) at the bottom of the leak, at the top of the porous column and at the top of the leak. If ComputeDrivingAndCounteringForces is True, it also returns the pressures (in Pa) of the mechanisms driving the leakage that appear in Equation **32**. The input parameters are:

Parameter	Symbol	Unit	Signification
h_b	h_-	m	Bottom aquifer thickness
w_b	ω_-	$\in [0,1]$	Bottom aquifer porosity
k_b	k_-	m^2	Bottom aquifer permeability
Cr_b		Pa^{-1}	Bottom aquifer pore compressibility
h_t	h_+	m	Top aquifer thickness
w_t	ω_+	$\in [0,1]$	Top aquifer porosity
k_t	k_+	m^2	Top aquifer permeability
Cr_t		Pa^{-1}	Top aquifer pore compressibility

MixingCoef		$\in [0,1]$	Part of the lifting brine mixing in the top aquifer and not flowing back down if the flow reverses.
T_lb		°C	Temperature at the bottom of the leak
gradT		$°C.m^{-1}$	Vertical temperature gradient (>0 for increasing temperature with depth)
P_lb		Pa	Pressure at the bottom of the leak
r_l	r_L	m	Leak radius
h_l	h_L	m	Leak height
h_pc	h_{pc}	m	Porous column height. Set to 0 if there is no porous col.
w_pc	ω_{pc}	$\in [0,1]$	Porous column porosity. Set to 1 if there is no porous col.
k_pc	k_{pc}	m^2	Porous column permeability. Set to 1 if there no porous column
Xs_b	X_{S-}		Salt mass fraction of the bottom aquifer brine
Xs_t	X_{S+}		Salt mass fraction of the top aquifer brine
Xs0_lifted	$X_{S,L0}$		Salt mass fraction of the brine initially filling the leak at z=0
gradXs_lifted	$gradX_{S,L}$	m^{-1}	Salt mass fraction gradient of the lifted brine (>0 for increasing salinity with depth). Default = 0
d		m	Leak to injection distance in the bottom aquifer
Q0	Q_0	$m^3.s^{-1}$	Injection flow rate in the bottom aquifer
t_inj	t_{inj}	s	Injection duration
t_sim	t_{sim}	s	Simulation duration $t_{sim} \geq t_{inj}$
IsThermalEq		Binary	Leakage flow at thermal equilibrium if True, in adiabatic conditions if False. Default = True

IsTubingPressure LossIncluded		Binary	Default = False
ComputePressures		Binary	Default = False
ComputeDrivingAnd CounteringForces		Binary	Default = False
ExportInTextFile		False or text	Default = False. Otherwise, input a recording text file name in string format (e.g. ‘MySimulation.txt’)
discretization		integer	Number of time steps. Default = 500

```
GetDimensionlessGroupings(h_b, w_b, k_b, Cr_b,                    # Bottom aquifer properties
h_t, w_t, k_t, Cr_t, MixingCoef,            # Top aquiferproperties
T_lb, gradT, P_lb,                          # T,P initial conditions
r_l, h_l,h_pc, w_pc, k_pc,              # leak properties.
Xs_b, Xs_t, Xs0_lifted, gradXs_lifted# Brines salinities
    d, Q0,                       # Injection parameters
IsThermalEq,
IsTubingPLossInclude)
```

This command computes the dimensionless formulation of the problem:

$\beta, \kappa, \varphi, \tau, \delta, \widetilde{\Delta\xi}, \widetilde{\Delta\rho_0}, \widetilde{\Delta P_{pc}^{1,bk}}, \widetilde{\Delta P_{wb}^{1,bk}}, \tilde{\mu}_{lifting}, \tilde{\mu}_{lifted}, W_0, W_{L-}, W_{L+}, \Lambda$. Note that τ has the dimension of a time. It is based on the following parameters:

Parameter	Symbol	Unit	Signification
h_b	h_-	m	Bottom aquifer thickness
w_b	ω_-	$\in [0,1]$	Bottom aquifer porosity
k_b	k_-	m^2	Bottom aquifer permeability
Cr_b		Pa^{-1}	Bottom aquifer pore compressibility
h_t	h_+	m	Top aquifer thickness
w_t	ω_+	$\in [0,1]$	Top aquifer porosity

k_t	k_+	m^2	Top aquifer permeability
Cr_t		Pa^{-1}	Top aquifer pore compressibility
MixingCoef		$\in [0,1]$	Part of the lifting brine mixing in the top aquifer and not flowing back down if the flow reverses.
T_lb		°C	Temperature at the bottom of the leak
gradT		$°C.m^{-1}$	Vertical temperature gradient (>0 for increasing temperature with depth)
P_lb		Pa	Pressure at the bottom of the leak
r_l	r_L	m	Leak radius
h_l	h_L	m	Leak height
h_pc	h_{pc}	m	Porous column height. Set to 0 if there is no porous col.
w_pc	ω_{pc}	$\in [0,1]$	Porous column porosity. Set to 1 if there is no porous column
k_pc	k_{pc}	m^2	Porous column permeability
Xs_b	X_{S-}		Salt mass fraction of the bottom aquifer brine
Xs_t	X_{S+}		Salt mass fraction of the top aquifer brine
Xs0_lifted	$X_{S,L0}$		Salt mass fraction of the brine initially filling the leak at z=0
gradXs_lifted	$gradX_{S,L}$	m^{-1}	Salt mass fraction gradient of the brine initially filling the leak (>0 for increasing salinity with depth). Default = 0
d		m	Leak to injection distance in the bottom aquifer
Q0	Q_0	$m^3.s^{-1}$	Injection flow rate in the bottom aquifer
IsThermalEq		Binary	Leakage flow at thermal equilibrium if True; in adiabatic conditions if False. Default =

			True
IsTubingPressure LossIncluded		Binary	Default = False

SemiAnalyticalResolution(beta, kappa, phi, tau, delta, w_pc, Dxi, Drho0, DP1bk_pc, DP1bk_wb, mu_pc_lifting, mu_pc_lifted, MixingCoef, W0, W_lb, W_lt, t_inj, t_sim, LambdaTable, ReturnLocalVar, discretization)

This command computes the time (s), the dimensionless leakage flow rate and position of the lifting/lifted brines interface. If ReturnLocalVar is True, it also returns: $\widetilde{\Delta P^1_{pc}}, \widetilde{\Delta P^1_{wb}}, \Lambda\widetilde{\mathrm{Q_L}}^2, \tilde{\mu}_{pc}$ and $[\widetilde{Q_{L,k}}], [\gamma_k], [t_k]\ for\ k \in [0, m]$. All these elements are dimensionless time series except the local maxima times $[t_k]$, in seconds. It is based on the following parameters:

Parameter	Symbol	Signification
beta	β	Dimensionless parameter representing the bottom aquifer injectivity
kappa	κ	Dimensionless parameter describing the Darcy flow in the porous column (permeability divided by column height)
phi	φ	Dimensionless parameter representing the lifting brine density decrease with depth
tau	τ	Characteristic time in seconds
delta	δ	Dimensionless porous column height
w_pc	ω_{pc}	Porous column porosity
Dxi	$\widetilde{\Delta\xi}$	Dimensionless parameter representing the lifting-lifted brines density decrease with depth difference
Drho0	$\widetilde{\Delta\rho_0}$	Dimensionless parameter representing the lifting-lifted brines density difference
DP1bk_pc	$\widetilde{\Delta P^{1,bk}_{pc}}$	Dimensionless pressure difference in the porous column fully filled with lifting brine compared to the initial situation (filled with lifted brine)

DP1bk_wb	$\Delta\widetilde{P_{wb}^{1,bk}}$	Dimensionless pressure difference in the wellbore fully filled with lifting brine compared to the initial situation (filled with lifted brine)
ViscosityXs_lifting	$\tilde{\mu}_{lifting}$	Dimensionless viscosity of the lifting brine
ViscosityXs_lifted	$\tilde{\mu}_{lifted}$	Dimensionless viscosity of the lifted brine
W0	W_0	Dimensionless well function giving the bottom aquifer pressurization created by the storage injection at the leak distance d
W_lb	W_{L-}	Dimensionless well function giving the pressure impact of the leakage flow in the bottom aquifer at the passive well
W_lt	W_{L+}	Dimensionless well function giving the pressure impact of the leakage flow in the top aquifer at the passive well
t_inj	t_{inj}	Injection duration in seconds
t_sim	t_{sim}	Simulation time in seconds
LambdaTable	$\Lambda(\widetilde{Q_L})$	Table of the dimensionless λparameter of the tubing pressure losses
ReturnLocalVar	Binary	If True, returns the local variables used for the computation. Default = False
discretization		Number of time steps

GetPressures(t, DP1_wb, TubingPLoss, MonotonicQl, ExtremaTimes, gamma, W0, W_lb, W_lt, t_inj)

This command computes the dimensionless pressure increase at the bottom of the leak, at the top of the porous column and at the top of the leak. It is based on parameters from the dimensionless formulation of the problem and on local variables used during the resolution:

Parameter	Symbol	Signification
t		The time array that has been used in the semi-analytical resolution, in seconds

DP1_wb	$\widetilde{\Delta P_{wb}^{1}}$	The dimensionless pressure increase in the wellbore due to lifted brine replacement by lifting one
TubingPLoss	$\Lambda\widetilde{Q_L}^{2}$	Dimensionless pressure loss in the tubings over time
MonotonicQl	$[\widetilde{Q_{L,k}}]$ $k \in [0,m]$	Monotonic leakage functions time records
ExtremaTimes	$[t_0, \ldots, t_m]$	Times of extrema in the leakage rates
gamma	$[\gamma_k]$ $k \in [0,m]$	Parameter of the approximate evaluation of the convolution integral
W0	W_0	Dimensionless well function giving the bottom aquifer pressurization created by the storage injection at the leak distance d
W_lb	W_{L-}	Dimensionless well function corresponding to the leakage from the bottom reservoir at the leak radius
W_lt	W_{L+}	Dimensionless well function corresponding to the leakage in the top reservoir at the leak radius
t_inj	t_{inj}	Injection duration in seconds

GetDrivingAndCounteringForces(t, DP1_pc, DP1_wb, TubingPLoss, mu_pc_av, MonotonicQl, ExtremaTimes, gamma, kappa, W0, W_lb, W_lt, t_inj)

This command computes the dimensionless driving and countering forces, as identified in Réveillère, 2013, Equation **32**): Injection pressurization, Brines density difference effect, Porous column flow resistance, Aquifers flow resistance, and Tubing pressure loss. It is based on the following parameters:

Parameter	Symbol	Signification
t		The time array that has been used in the semi-analytical resolution

DP1_pc	$\widetilde{\Delta P^1_{pc}}$	The dimensionless pressure increase in the porous column due to lifted brine replacement by lifting one
DP1_wb	$\widetilde{\Delta P^1_{wb}}$	The dimensionless pressure increase in the wellbore due to lifted brine replacement by lifting one
TubingPLoss	$\Lambda\widetilde{\mathrm{Q_L}}^2$	Dimensionless pressure loss in the open wellbore over time
mu_pc_av	$\widetilde{\mathrm{\mu_{pc}}}$	Dimensionless average viscosity in the porous column over time
MonotonicQl	$[\widetilde{Q_{L,k}}]$ $k \in [0, m]$	Monotonic leakage functions over time
ExtremaTimes	$[t_0, \ldots, t_m]$	List of extrema times of the leakage rates
gamma	$[\gamma_k]$ $k \in [0, m]$	Parameter of the approximate evaluation of the convolution integral over time
kappa	κ	Dimensionless parameter describing the Darcy flow in the porous column (permeability divided by column height)
W0	W_0	Dimensionless well function giving the bottom aquifer pressurization created by the storage injection at the leak distance d
W_lb	W_{L-}	Dimensionless well function giving the pressure impact of the leakage flow in the bottom aquifer at the passive well
W_lt	W_{L+}	Dimensionless well function giving the pressure impact of the leakage flow in the top aquifer at the passive well
t_inj	t_{inj}	Injection duration in seconds

GetDimensionalPressure(PD, Q0, h_b, k_b, P_lb, Xs_b, T_lb, gradT)

This command computes the dimensional pressure based on:

Parameter	Symbol	Unit	Signification
PD	$\tilde{\mathrm{P}}$		Dimensionless pressure
Q0	Q_0	$m^3.s^{-1}$	Injection flow rate in the bottom aquifer
h_b	h_-	m	Bottom aquifer thickness
k_b	k_-	m^2	Bottom aquifer permeability
P_lb		Pa	Pressure at the bottom of the leak
Xs_b	X_{S-}		Salt mass fraction of the bottom aquifer brine
T_lb		°C	Temperature at the bottom of the leak
gradT		$°C.m^{-1}$	Vertical temperature gradient (>0 for increasing temperature with depth)

GetDimensionalFlow(QD, Q0)

This command computes the dimensional flow rate based on:

Parameter	Symbol	Unit	Signification
QD	$\tilde{\mathrm{Q}}$		Dimensionless flow
Q0	Q_0	$m^3.s^{-1}$	Injection flow rate in the bottom aquifer

GetDimensionalZ(zD, h_l, z_lb)

This command computes the dimensional z-vertical coordinate based on:

Parameter	Symbol	Unit	Signification
zD	$\tilde{\mathrm{z}}$		Dimensionless z coordinate
h_l	h_L	m	Leak height

z_lb		m	z-coordinate value for the bottom of the leak. Default = 0.

Nordbotten_et_al_2004(kappa, ViscosityD, W0, W_lb, W_lt, gamma)

This command computes the dimensionless analytical solution in the case of a leak modeled by a porous column and of equal brines salinites (no density difference), i.e. the solution by Nordbotten et al. (2004). It is based on the following parameters:

Parameter	Symbol	Signification
kappa	κ	Dimensionless parameter describing the Darcy flow in the porous column (permeability divided by column height)
ViscosityD	$\tilde{\mu}$	Dimensionless average viscosity in the porous column
W0	W_0	Dimensionless well function giving the bottom aquifer pressurization created by the storage injection at the leak distance d
W_lb	W_{L-}	Dimensionless well function giving the pressure impact of the leakage flow in the bottom aquifer at the passive well
W_lt	W_{L+}	Dimensionless well function giving the pressure impact of the leakage flow in the top aquifer at the passive well
gamma	γ	Constant parameter of the approximate evaluation of the convolution integrals. Default = 0.92

Modified_Nordbotten_et_al_2004(kappa, ViscosityD, W0, W_lb, W_lt,tinj, discretization)

This command computes the dimensionless leakage rate in the case of a leak modeled by a porous column and of equal brines salinities (no density difference), i.e. almost similarly to the solution by Nordbotten et al. (2004). The difference is that the gamma parameter is fluctuating as presented in section 3.1. It is based on the following parameters

Parameter	Symbol	Signification
kappa	κ	Dimensionless parameter describing the Darcy flow in the porous column (permeability divided by column height)
ViscosityD	$\tilde{\mu}$	Dimensionless average viscosity in the porous column
W0	W_0	Dimensionless well function giving the bottom aquifer pressurization created by the storage injection at the leak distance d
W_lb	W_{L-}	Dimensionless well function giving the pressure impact of the leakage flow in the bottom aquifer at the passive well
W_lt	W_{L+}	Dimensionless well function giving the pressure impact of the leakage flow in the top aquifer at the passive well
t_inj	t_{inj}	Injection duration in seconds
discretization		Number of time steps. Default = 500

GetPressureLossInTubing(length, radius, rugosity, FlowRate, density, viscosity)

This command computes the pressure loss (in Pa) in a flowing well (personal communication of Herve Lesueur, geothermal energy department, BRGM). It is based on:

Parameter	Symbol	Unit	Signification
length	h_{wb}	m	Tubing length
radius	r_L	m	Well internal radius
rugosity		m	Average rugosity. Use for instance: New tubing: 0.05 mm Corroded tubing: 0.5mm

			Very corroded tubing: 5 mm
FlowRate		$m^3.s^{-1}$	Volumetric brine flow rate
density		$kg.m^{-3}$	Brine density
viscosity		Pa.s	Brine viscosity

GetPressureLossCoefficient (radius, rugosity, FlowRate, density, viscosity, negligible, nmax)

This command computes the λ coefficient of the Darcy-Weissbach equation according to equations developed for Paris basin geothermal applications (personal communication of Herve Lesueur, geothermal energy department, BRGM). It is based on:

Parameter	Symbol	Unit	Signification
radius	r_L	m	Well internal radius
rugosity		m	Average rugosity. Use for instance: New tubing: 0.05 mm Corroded tubing: 0.5mm Very corroded tubing: 5 mm
FlowRate		$m^3.s^{-1}$	Volumetric brine flow rate
density		$kg.m^{-3}$	Brine density
Viscosity		Pa.s	Brine viscosity

5.4. TOUGH2 PRE- AND POST- PROCESSING MODULES

These modules present the functions that have been useful for pre-processing TOUGH2 simulations or post-processing its results. It is only necessary when comparing the results to TOUGH2 simulations, which is done in Réveillère, 2013 for instance.

float2(s)

This command converts a Python string s to a Python float. It deals with '0.3149-124' (=0.3149e-124) format.

GetBrineMassFlowFromCOFT(path, dt_min)

This command returns a Python list of the brine mass flow recorded in the output TOUGH2 COFT file:

[[time(s)], [mass flow rate Connexion 1 (kg/s)], [mass flow rate Connexion 2]...]

The COFT file must be located in path (relative or absolute), and a minimum time interval dt_min (s) can be considered between two returned mass flow rates (default: dt_min =0, i.e. all simulated time steps are returned).This works with TOUGH2/eco2n output COFT files

GetMolality(Xs)

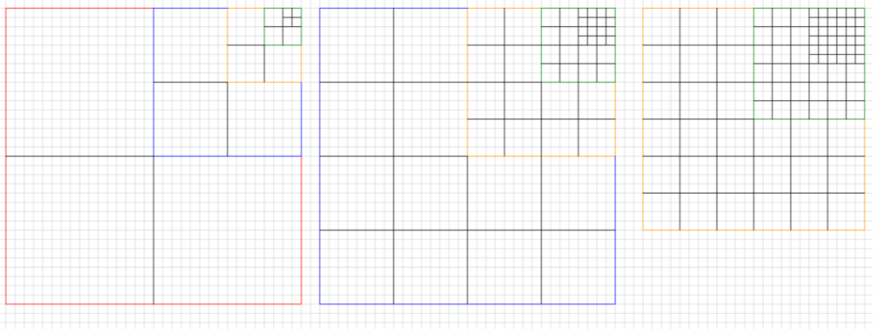
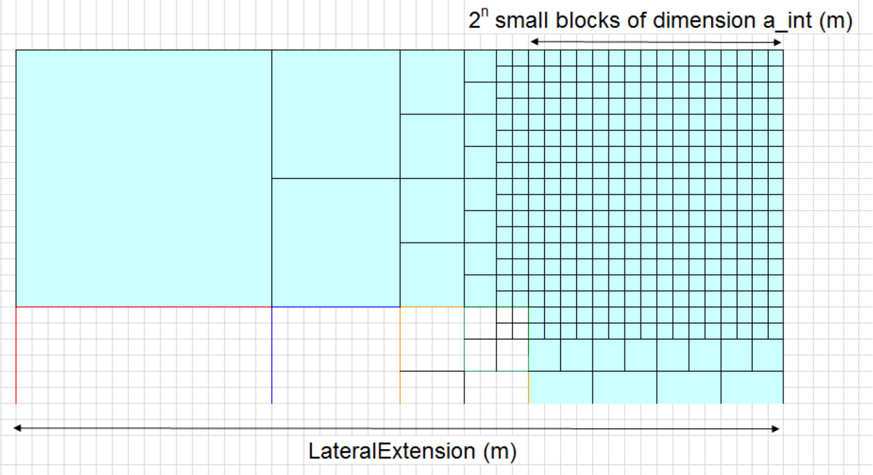
This command computes the brine molality= (solute amount)/(solvent mass) based on the brine salt mass fraction Xs= (solute mass)/(solution mass)

GetSaltMassFraction(Molality)

This command computes the brine salt mass fraction = (solute mass)/(solution mass) based on the brine molality= (solute amount)/(solvent mass)

BuildExponentialMesh(StripWidth, SymetryCase, LateralExtension, n, a_int, Layers)

This command writes Femesh2tough.itt and LGR.itt input files to Audigane et al. (2011) mesh-making tools in order to build a TOUGH2 MESH file with an exponential increase of grid blocks horizontal section. It is based on the following parameters:

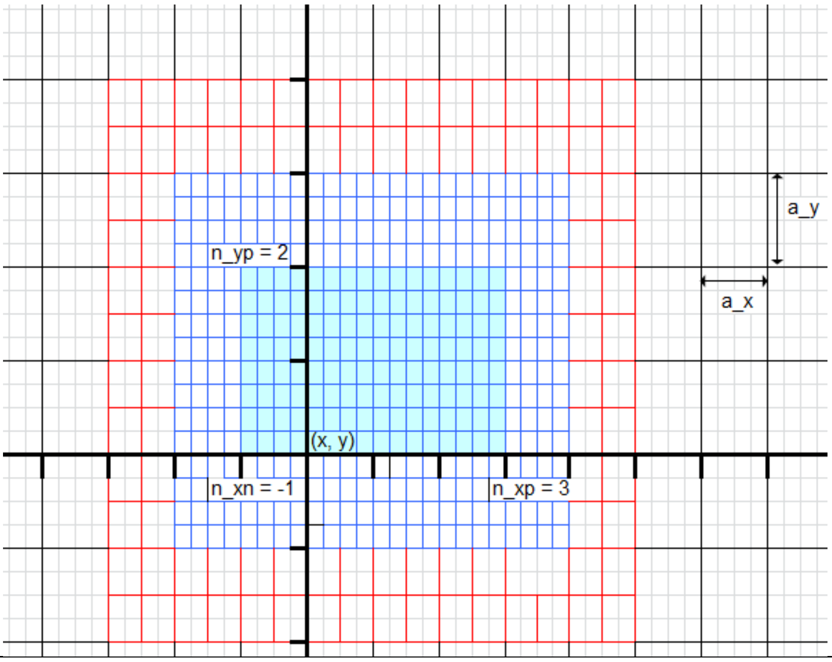
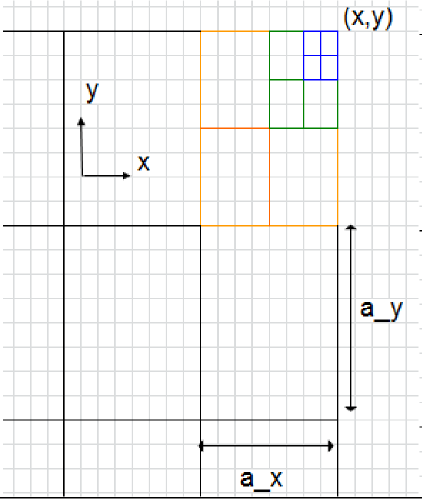
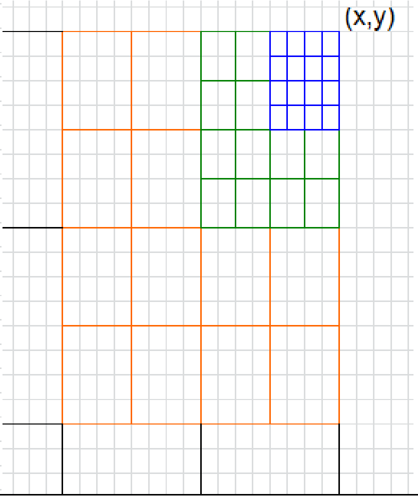
Parameter	**Signification**
StripWidth	Number of consecutive cells having the same dimensions. For instance: StripWidth=1 StripWidth=2 StripWidth=3 
SymetryCase	1:quarter of mesh x>0 and y>0; 2: half mesh y>0; 3: full square mesh
LateralExtension	Minimum Lateral extension demanded, in meters
n	Build a central area of $(2^n)^2$ grid blocks of equal dimensions (cf. the blue area below):  2^n small blocks of dimension a_int (m) LateralExtension (m)
a_int	Dimension of the internal area grid blocks in meters
Layers	List of layers boundaries, from the deepest to the shallowest, in meters.

```
BuildExpRefinement(x, y, Shape, a_x, a_y, z_min, z_max, DivisionNumber, CentralArea,StripWidth)
```

This command writes a LGR.itt (it stands for Local Grid Refinement) input files to Audigane et al. (2011) mesh-making tools. From a mesh of uniform horizontal grid blocks (a_x * a_y), it creates a logarithmic refinement centered in (x, y). This is defined by the following parameters:

<table>
<tr><th>Parameter</th><th>Signification</th></tr>
<tr><td>X</td><td>x-axis position of the corner where the division will be maximal</td></tr>
<tr><td>Y</td><td>y-axis position of the corner where the division will be maximal</td></tr>
<tr><td>Shape</td><td>List of 4 binaries [xn, xp, yn, yp] (for positive or negative x and y) defining on which direction the refinement should be made. Cf. the following view from the top and parameters for the dashed area:

^ y
  ////////////////// |
  // xn = 1 //// |
  // xp = 0 //// |
  // yn = 0 //// |
  // yp = 1 ////|
  ////////////////// |
_______________|_______________\
  |(x,y)   x /
 |
 |
 |
 |</td></tr>
<tr><td>a_x</td><td>Block dimension on the x-axis</td></tr>
<tr><td>a_y</td><td>Block dimension on the y-axis</td></tr>
<tr><td>z_min</td><td>Minimum depth at which the refinement should be done</td></tr>
<tr><td>z_max</td><td>Maximum depth at which the refinement should be done</td></tr>
<tr><td>DivisionNumber</td><td>The initial blocks dimensions will be divided by $2^{DivisionNumber}$</td></tr>
<tr><td>CentralArea</td><td>List of 4 integers [n_xn, n_xp, n_yn, n_yp] defining the number grid blocks of the initial MESH that will be divided in an area of equal</td></tr>
</table>

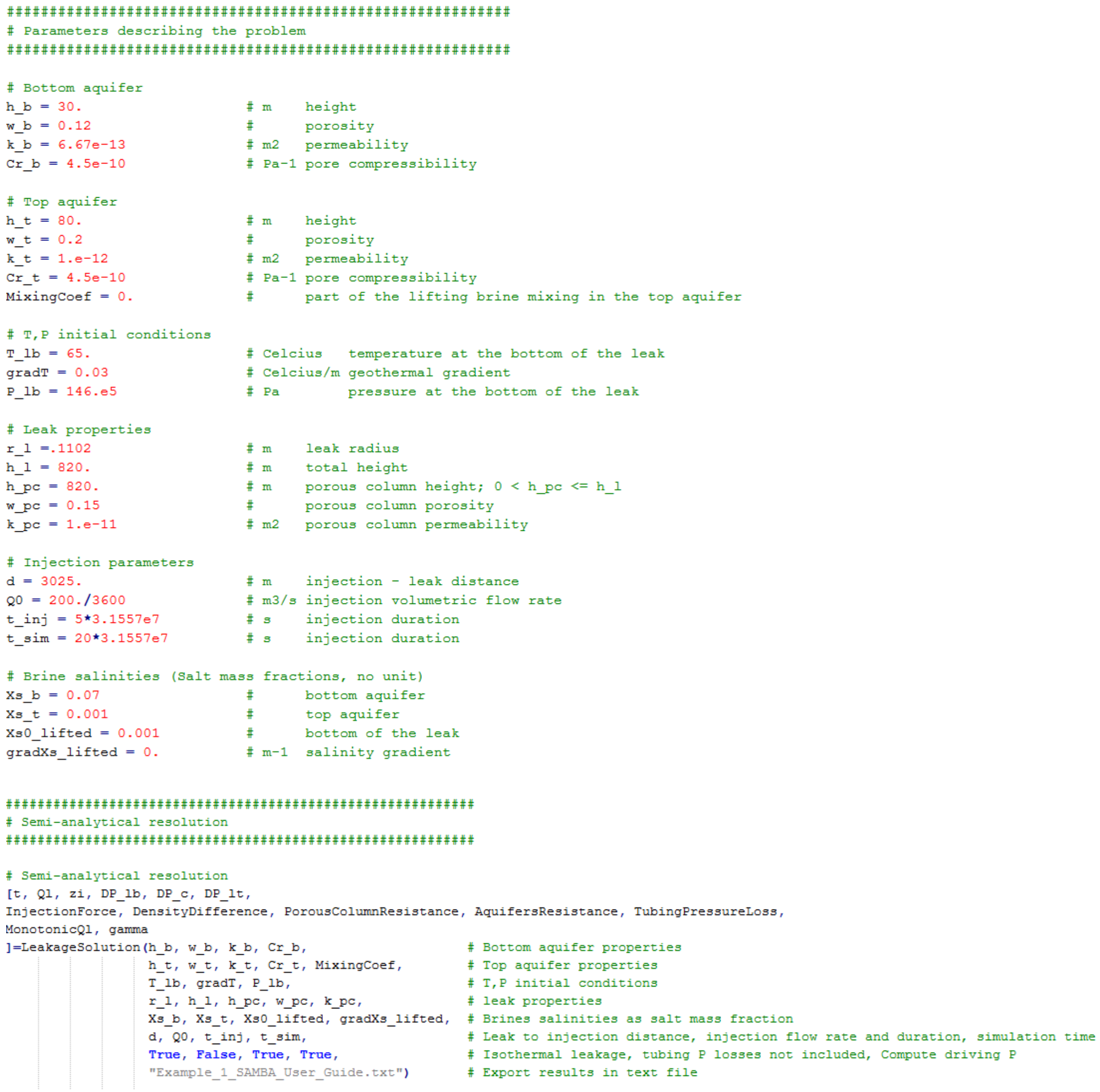
	dimensions grid blocks. Cf. the blue area below, it corresponds to CentralArea = [-1, 3, 0, 2] 
StripWidth	Number of consecutive cells having the same dimensions. For instance: StripWidth=1 StripWidth =2 etc.  

6. Examples

First, note that all input files used for the comparisons presented in Réveillère (2013) are available on the website http://github.com/arnaud-reveillere/SAMBA. The results of the examples given below are discussed in Réveillère (2013).

6.1. EXAMPLE 1: LEAK MODELED BY A POROUS COLUMN

This example presents all possible results for the Example 1 presented in Réveillère (2013) for the case $X_{s-} = 7\%$. It is available on http://github.com/arnaud-reveillere/SAMBA. t_sim is the simulation duration.



```
###########################################################
# Parameters describing the problem
###########################################################

# Bottom aquifer
h_b = 30.                   # m    height
w_b = 0.12                  #      porosity
k_b = 6.67e-13              # m2   permeability
Cr_b = 4.5e-10              # Pa-1 pore compressibility

# Top aquifer
h_t = 80.                   # m    height
w_t = 0.2                   #      porosity
k_t = 1.e-12                # m2   permeability
Cr_t = 4.5e-10              # Pa-1 pore compressibility
MixingCoef = 0.             #      part of the lifting brine mixing in the top aquifer

# T,P initial conditions
T_lb = 65.                  # Celcius   temperature at the bottom of the leak
gradT = 0.03                # Celcius/m geothermal gradient
P_lb = 146.e5               # Pa        pressure at the bottom of the leak

# Leak properties
r_l =.1102                  # m    leak radius
h_l = 820.                  # m    total height
h_pc = 820.                 # m    porous column height; 0 < h_pc <= h_l
w_pc = 0.15                 #      porous column porosity
k_pc = 1.e-11               # m2   porous column permeability

# Injection parameters
d = 3025.                   # m    injection - leak distance
Q0 = 200./3600              # m3/s injection volumetric flow rate
t_inj = 5*3.1557e7          # s    injection duration
t_sim = 20*3.1557e7         # s    injection duration

# Brine salinities (Salt mass fractions, no unit)
Xs_b = 0.07                 #      bottom aquifer
Xs_t = 0.001                #      top aquifer
Xs0_lifted = 0.001          #      bottom of the leak
gradXs_lifted = 0.          # m-1  salinity gradient


###########################################################
# Semi-analytical resolution
###########################################################

# Semi-analytical resolution
[t, Ql, zi, DP_lb, DP_c, DP_lt,
InjectionForce, DensityDifference, PorousColumnResistance, AquifersResistance, TubingPressureLoss,
MonotonicQl, gamma
]=LeakageSolution(h_b, w_b, k_b, Cr_b,                     # Bottom aquifer properties
                  h_t, w_t, k_t, Cr_t, MixingCoef,         # Top aquifer properties
                  T_lb, gradT, P_lb,                       # T,P initial conditions
                  r_l, h_l, h_pc, w_pc, k_pc,              # leak properties
                  Xs_b, Xs_t, Xs0_lifted, gradXs_lifted,   # Brines salinities as salt mass fraction
                  d, Q0, t_inj, t_sim,                     # Leak to injection distance, injection flow rate and duration, simulation time
                  True, False, True, True,                 # Isothermal leakage, tubing P losses not included, Compute driving P
                  "Example_1_SAMBA_User_Guide.txt")        # Export results in text file
```

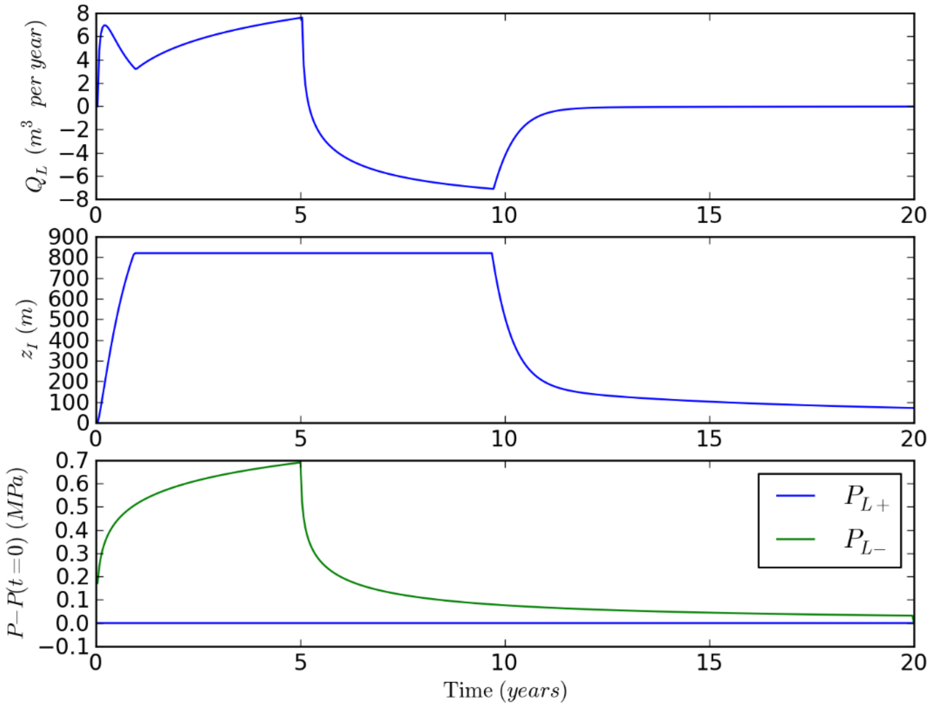


Figure 6: Flow rate, interface postion and over-pressures over time for Example 1.

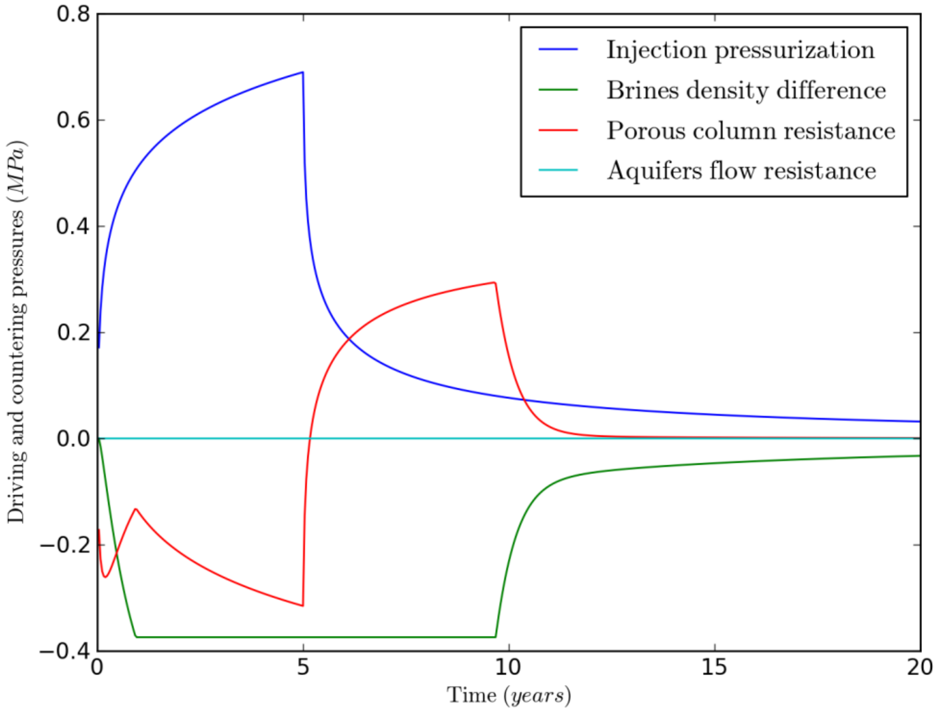


Figure 7: Driving upwards (when positive) or downwards (when negative) pressures

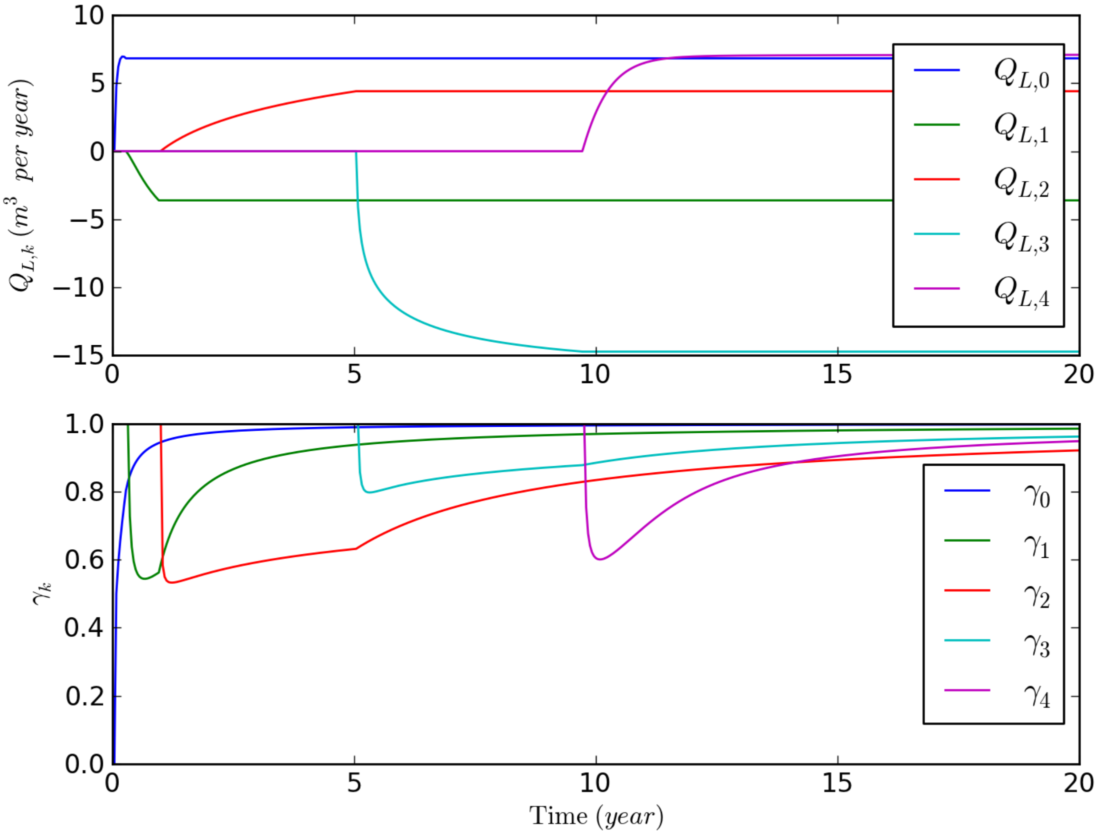
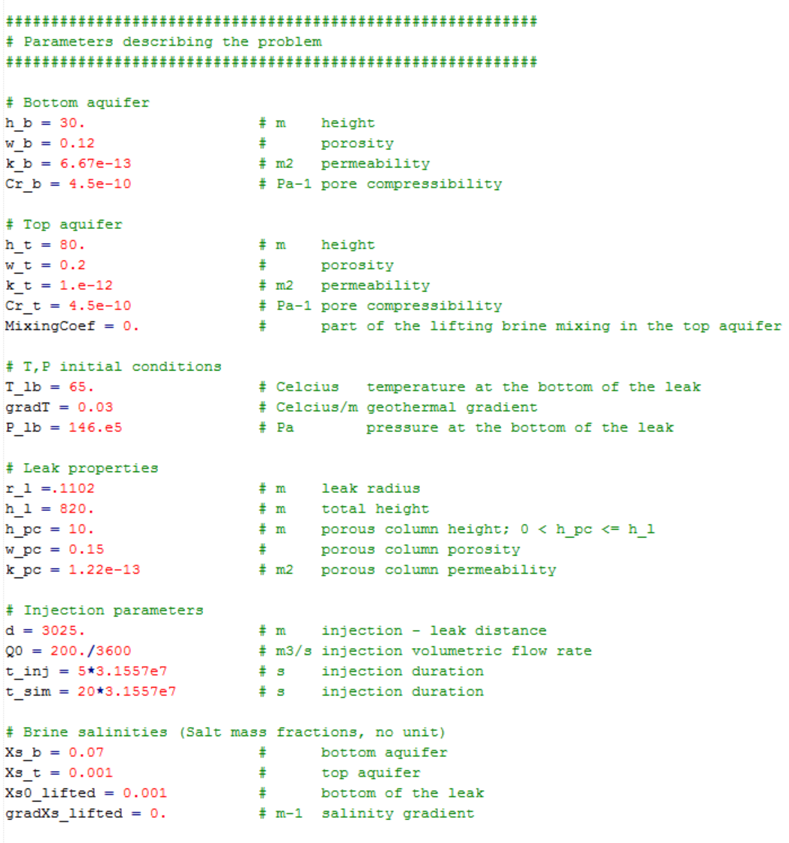


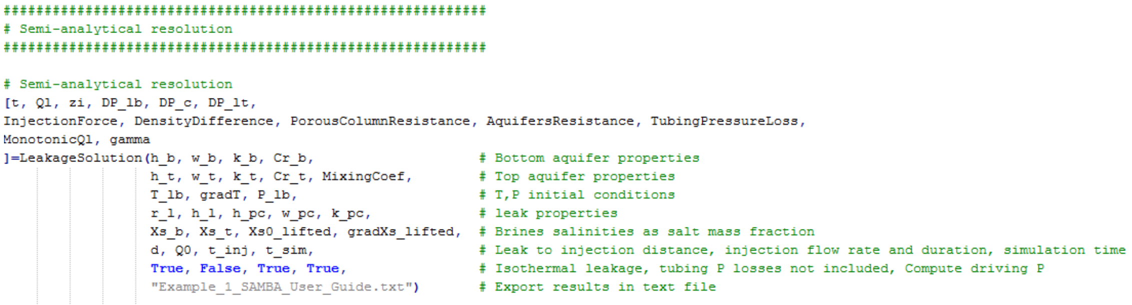
Figure 8: Construction of monotonic leakage rates $Q_{L,k}$*, evolution of* $Q_{L,k}$ *and* γ_k*over time*

6.2. EXAMPLE 2: LEAK MODELED BY BOTH A POROUS COLUMN AND AN OPEN WELLBORE

This example presents all possible results for the Example 2 presented in Réveillère (2013) for the case $X_{s-} = 7\%$. It is available on http://github.com/arnaud-reveillere/SAMBA. t_sim is the simulation duration.



```
############################################################
# Parameters describing the problem
############################################################

# Bottom aquifer
h_b = 30.                    # m    height
w_b = 0.12                   #      porosity
k_b = 6.67e-13               # m2   permeability
Cr_b = 4.5e-10               # Pa-1 pore compressibility

# Top aquifer
h_t = 80.                    # m    height
w_t = 0.2                    #      porosity
k_t = 1.e-12                 # m2   permeability
Cr_t = 4.5e-10               # Pa-1 pore compressibility
MixingCoef = 0.              #      part of the lifting brine mixing in the top aquifer

# T,P initial conditions
T_lb = 65.                   # Celcius   temperature at the bottom of the leak
gradT = 0.03                 # Celcius/m geothermal gradient
P_lb = 146.e5                # Pa        pressure at the bottom of the leak

# Leak properties
r_l =.1102                   # m    leak radius
h_l = 820.                   # m    total height
h_pc = 10.                   # m    porous column height; 0 < h_pc <= h_l
w_pc = 0.15                  #      porous column porosity
k_pc = 1.22e-13              # m2   porous column permeability

# Injection parameters
d = 3025.                    # m    injection - leak distance
Q0 = 200./3600               # m3/s injection volumetric flow rate
t_inj = 5*3.1557e7           # s    injection duration
t_sim = 20*3.1557e7          # s    injection duration

# Brine salinities (Salt mass fractions, no unit)
Xs_b = 0.07                  #      bottom aquifer
Xs_t = 0.001                 #      top aquifer
Xs0_lifted = 0.001           #      bottom of the leak
gradXs_lifted = 0.           # m-1  salinity gradient
```



```
############################################################
# Semi-analytical resolution
############################################################

# Semi-analytical resolution
[t, Ql, zi, DP_lb, DP_c, DP_lt,
InjectionForce, DensityDifference, PorousColumnResistance, AquifersResistance, TubingPressureLoss,
MonotonicQl, gamma
]=LeakageSolution(h_b, w_b, k_b, Cr_b,                      # Bottom aquifer properties
                  h_t, w_t, k_t, Cr_t, MixingCoef,          # Top aquifer properties
                  T_lb, gradT, P_lb,                        # T,P initial conditions
                  r_l, h_l, h_pc, w_pc, k_pc,               # leak properties
                  Xs_b, Xs_t, Xs0_lifted, gradXs_lifted,    # Brines salinities as salt mass fraction
                  d, Q0, t_inj, t_sim,                      # Leak to injection distance, injection flow rate and duration, simulation time
                  True, False, True, True,                  # Isothermal leakage, tubing P losses not included, Compute driving P
                  "Example_1_SAMBA_User_Guide.txt")         # Export results in text file
```

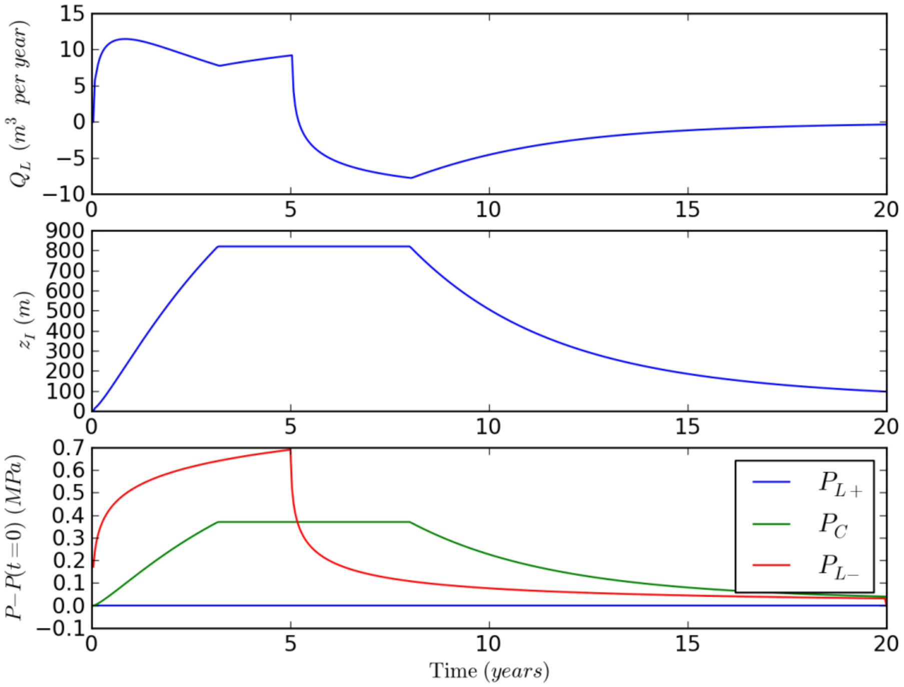


Figure 9: Flow rate, interface position and over-pressures over time for Example 2

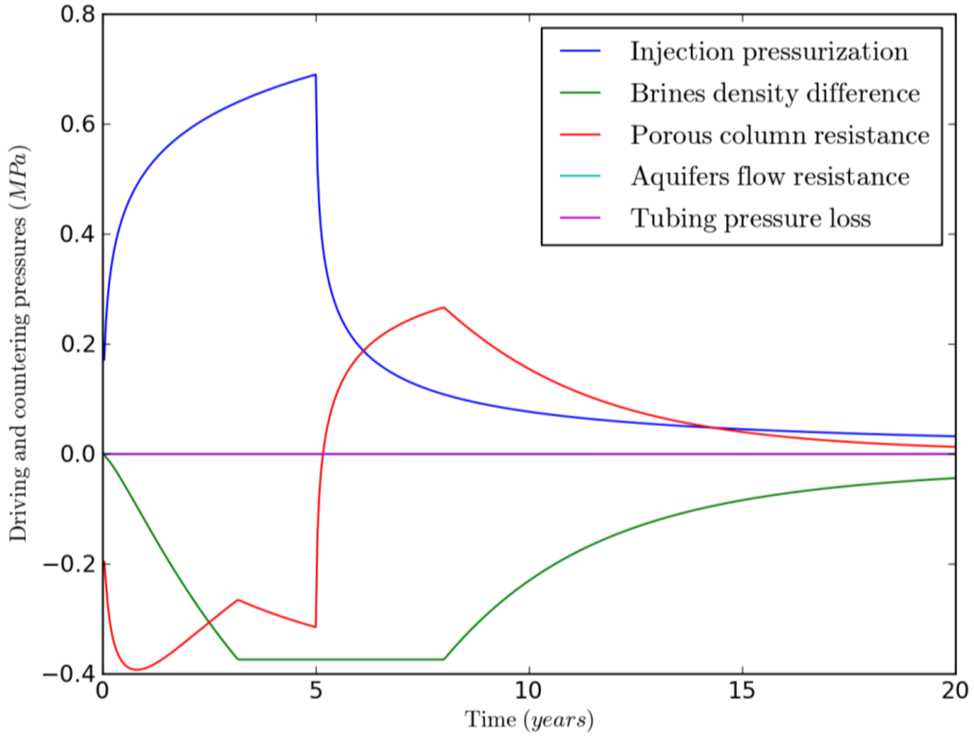


Figure 10: Driving upwards (when positive) or downwards (when negative) pressures

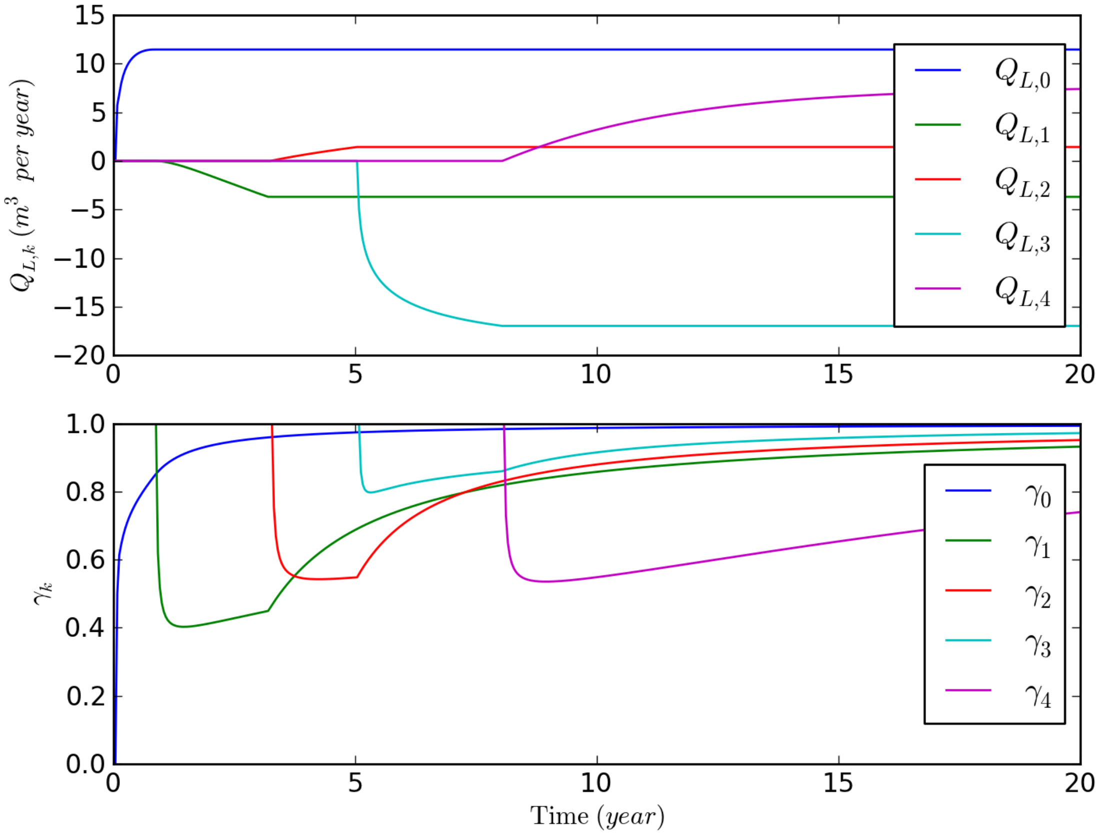
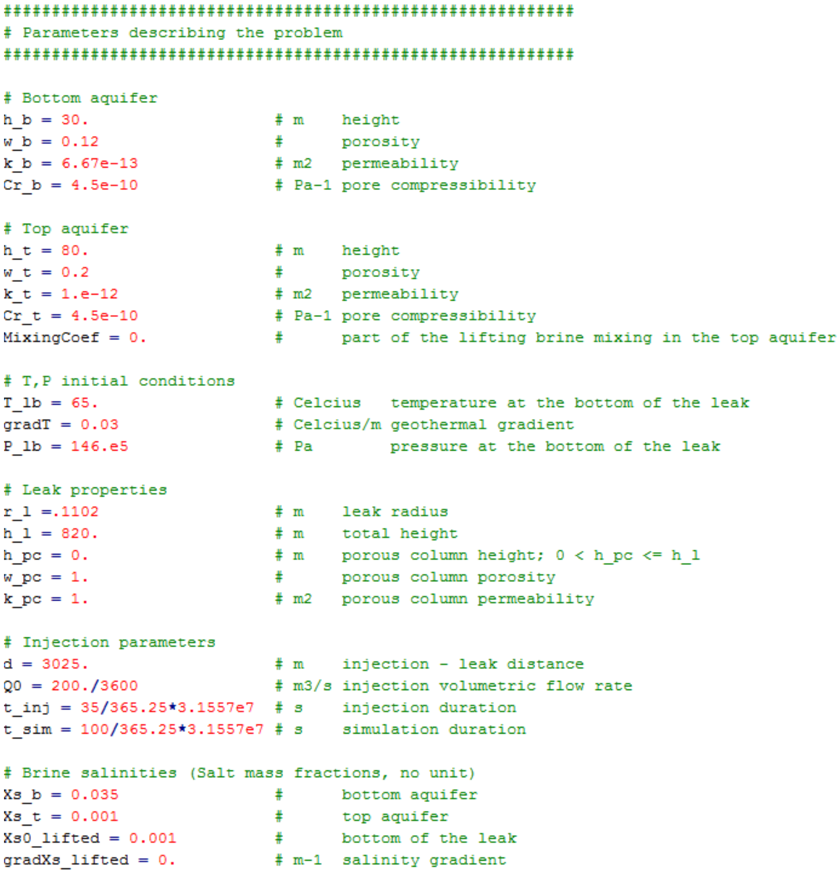


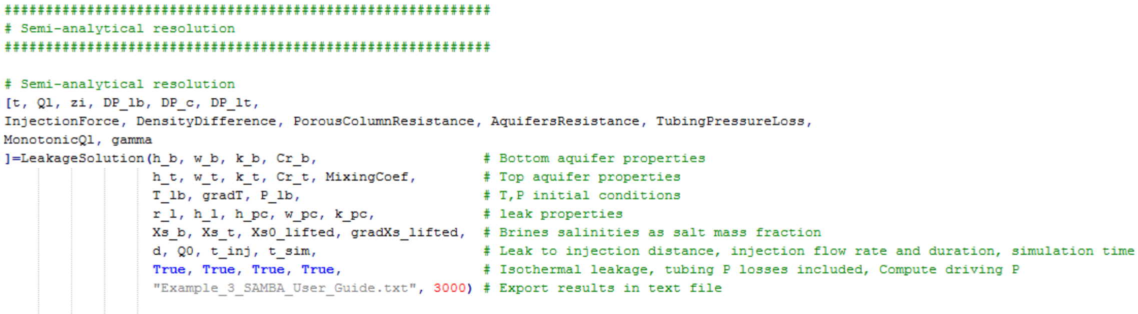
Figure 11: Construction of monotonic leakage rates $Q_{L,k}$, evolution of $Q_{L,k}$ and $\gamma_k$over time

6.3. EXAMPLE 3: LEAK MODELED AS AN OPEN WELLBORE

This example presents all possible results for the Example 3 presented in Réveillère (2013) for the case $X_{s-} = 3.5\%$. It is available on http://github.com/arnaud-reveillere/SAMBA. t_sim is the simulation duration.



```
###########################################################
# Parameters describing the problem
###########################################################

# Bottom aquifer
h_b = 30.                    # m    height
w_b = 0.12                   #      porosity
k_b = 6.67e-13               # m2   permeability
Cr_b = 4.5e-10               # Pa-1 pore compressibility

# Top aquifer
h_t = 80.                    # m    height
w_t = 0.2                    #      porosity
k_t = 1.e-12                 # m2   permeability
Cr_t = 4.5e-10               # Pa-1 pore compressibility
MixingCoef = 0.              #      part of the lifting brine mixing in the top aquifer

# T,P initial conditions
T_lb = 65.                   # Celcius   temperature at the bottom of the leak
gradT = 0.03                 # Celcius/m geothermal gradient
P_lb = 146.e5                # Pa        pressure at the bottom of the leak

# Leak properties
r_l =.1102                   # m    leak radius
h_l = 820.                   # m    total height
h_pc = 0.                    # m    porous column height; 0 < h_pc <= h_l
w_pc = 1.                    #      porous column porosity
k_pc = 1.                    # m2   porous column permeability

# Injection parameters
d = 3025.                    # m    injection - leak distance
Q0 = 200./3600               # m3/s injection volumetric flow rate
t_inj = 35/365.25*3.1557e7   # s    injection duration
t_sim = 100/365.25*3.1557e7  # s    simulation duration

# Brine salinities (Salt mass fractions, no unit)
Xs_b = 0.035                 #      bottom aquifer
Xs_t = 0.001                 #      top aquifer
Xs0_lifted = 0.001           #      bottom of the leak
gradXs_lifted = 0.           # m-1  salinity gradient
```



```
###########################################################
# Semi-analytical resolution
###########################################################

# Semi-analytical resolution
[t, Ql, zi, DP_lb, DP_c, DP_lt,
InjectionForce, DensityDifference, PorousColumnResistance, AquifersResistance, TubingPressureLoss,
MonotonicQl, gamma
]=LeakageSolution(h_b, w_b, k_b, Cr_b,                 # Bottom aquifer properties
                  h_t, w_t, k_t, Cr_t, MixingCoef,      # Top aquifer properties
                  T_lb, gradT, P_lb,                    # T,P initial conditions
                  r_l, h_l, h_pc, w_pc, k_pc,           # leak properties
                  Xs_b, Xs_t, Xs0_lifted, gradXs_lifted, # Brines salinities as salt mass fraction
                  d, Q0, t_inj, t_sim,                  # Leak to injection distance, injection flow rate and duration, simulation time
                  True, True, True, True,               # Isothermal leakage, tubing P losses included, Compute driving P
                  "Example_3_SAMBA_User_Guide.txt", 3000) # Export results in text file
```

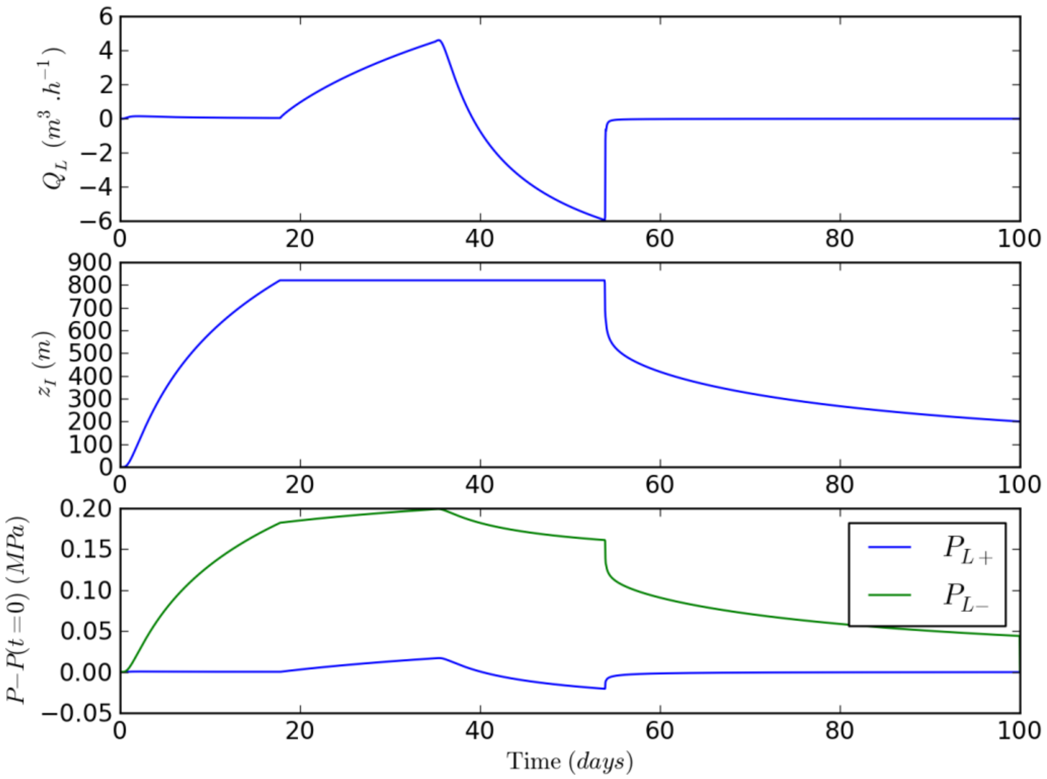


Figure 12: Flow rate, interface position and over-pressures over time for Example 3

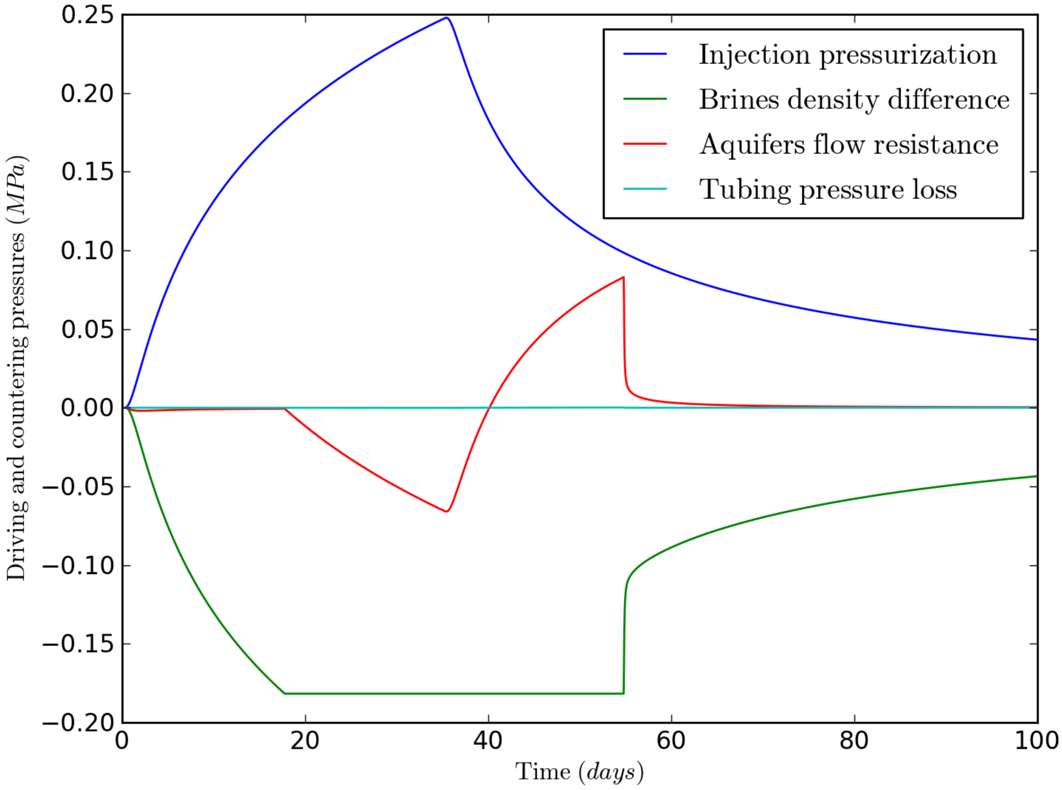


Figure 13: Driving upwards (when positive) or downwards (when negative) pressures

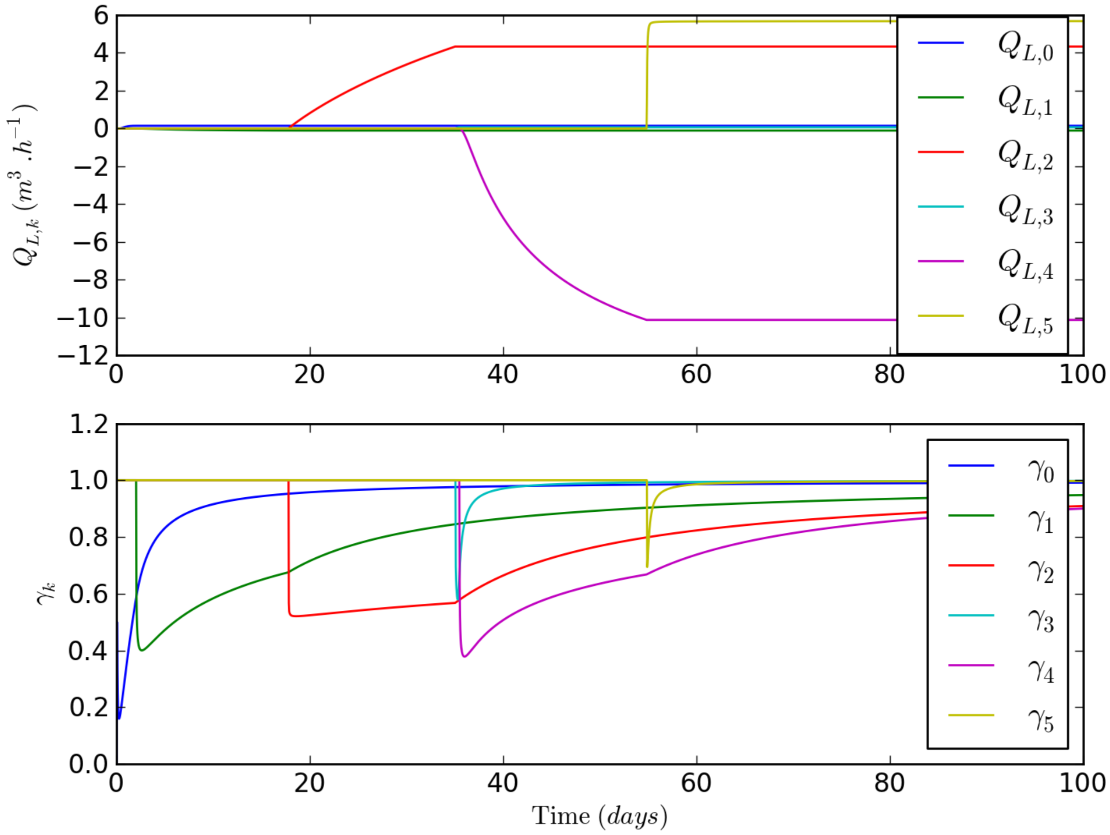


Figure 14: Construction of monotonic leakage rates $Q_{L,k}$, evolution of $Q_{L,k}$ and $\gamma_k$over time

7. Conclusion

This report describes a semi-analytical model developed for representing brine upwards migration through a porous column and/or an open wellbore following deep aquifer pressurization in the context of CO_2 geological storage. The major novelty of this model with respect to prior works lies in the accounting of density effects occurring during the progressive migration of the fluid. The report describes the model, its semi-analytical resolution and a technical manual for the code, which is distributed at http://github.com/arnaud-reveillere/SAMBA. It is meant as a user guide for this code, which has been successfully compared to TOUGH2 simulations in Réveillère (2013).

8. References

Audigane, P., Chiaberge, C., Mathurin, F., Lions, J., Picot-Colbeaux, G., 2011. A workflow for handling heterogeneous 3D models with the TOUGH2 family of codes: Applications to numerical modeling of CO2 geological storage. Computers & Geosciences 37 (2011) 610–62.

Bandilla, K. W., Kraemer, S. R., Birkholzer, J. T., 2012. Using semi-analytic solutions to approximate the area of potential impact for carbon dioxide injection. Int. J. of Greenhouse Gas Control 8 (2012) 196-204.

Birkholzer, J. T., Nicot, J. P., Oldenburg, C. M., Zhou, Q., Kraemer, S., Bandilla, K., 2011. Brine flow up a well caused by pressure perturbation from geologic carbon sequestration: Static and dynamic evaluations. Int. J. of Greenhouse Gas Control 5(2011) 850-861.

De Marsily, G, 1986. *Quantitative Hydrogeology*, Academic Press, Orlando, FL, 1986.

Nicot J.-P., Oldenburg, C., Bryant, S., Hovorka, S. (2009) Pressure perturbations from geologic carbon sequestration: Area-of-review boundaries and borehole leakage driving forces. Energy Procedia 1 (2009) 47-54

Nordbotten, J. M., Celia, M. A. and Bachu, S. (2004), Analytical solutions for leakage rates through abandoned wells, Water Resources Research 40, W04204.

Nordbotten, J. M., Celia, M. A. and Bachu, S., Dahle, K. (2005), Semianalytical solution for CO_2 leakage through an abandoned well, Environ. Sci. Technol. 2005, 39, 602-611.

Réveillère, A., 2013. Semi-analytical Solution for Brine Leakage Through Passive Abandoned Wells Taking Account of Brine Density Differences. Transport in Porous Media, November 2013, Volume 100, Issue 2, pp 337-361

Theis, C.V., 1935. The relationship between the lowering of the piezometric surface and the rate and duration of discharge of a well using ground water storage. Trans. Amer. Geophys. Union 16, 519–524

Geoscience for a sustainable Earth

brgm

Scientific and Technical Centre
Risk and Prevention Division
3, avenue Claude-Guillemin - BP 36009
45060 Orléans Cedex 2 – France – Tel.: +33 (0)2 38 64 34 34
www.brgm.fr