Computational Particle Mechanics (2021) 8:561–574
https://doi.org/10.1007/s40571-020-00353-2

A meshfree generalized finite difference method for solution mining processes

Isabel Michel[1] · Tobias Seifarth[1] · Jörg Kuhnert[1] · Pratik Suchde[1]

Received: 30 January 2020 / Revised: 23 July 2020 / Accepted: 19 August 2020 / Published online: 5 September 2020

Abstract
Experimental and field investigations for solution mining processes have improved intensely in recent years. Due to today's computing capacities, three-dimensional simulations of potential salt solution caverns can further enhance the understanding of these processes. They serve as a "virtual prototype" of a projected site and support planning in reasonable time. In this contribution, we present a meshfree generalized finite difference method (GFDM) based on a cloud of numerical points that is able to simulate solution mining processes on microscopic and macroscopic scales, which differ significantly in both the spatial and temporal scales. Focusing on anticipated industrial requirements, Lagrangian and Eulerian formulations including an Arbitrary Lagrangian–Eulerian (ALE) approach are considered.

Keywords Meshfree methods · Generalized finite difference method · Lagrangian formulation · Arbitrary Lagrangian–Eulerian formulation · Solution mining

1 Introduction

The basic motivation of this research is to provide a method that is able to simulate the long-term development of a salt cavern during a double-well solution mining process. Solution mining is used to extract underground water-soluble minerals such as salt and potash. A double-well convection process has been a preferred choice for solution mining due to its large recovery rate [2,33]. As the name suggests, this involves the use of two boreholes or wells for the extraction process: an injection well and a recovery or extraction well. For the extraction of salt, freshwater is pumped into a salt deposit through the first "injection" well. Salt present in the cavern dissolves in the water to produce a saturated brine solution. This is then extracted at the second "extraction" well. A schematic of this process is shown in Fig. 1. The main direction of dissolution is vertical, which is controlled by alternate lifting of the injection and the extraction well.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s40571-020-00353-2) contains supplementary material, which is available to authorized users.

✉ Isabel Michel
isabel.michel@itwm.fraunhofer.de

1 Fraunhofer Institute for Industrial Mathematics ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

In this work, we focus on modeling of the fluid flow involved in such a double-well solution mining procedure, including the formation of the salt–water solution. An essential aspect of this is to accurately model the long-term geometrical evolution of the salt cavern. This is needed to steer the actual process of solution mining, in terms of, for example, determining when and at what rate the injection and extraction wells are raised. However, numerically modeling this is very challenging, as it is a highly dynamic three-dimensional process involving different spatial and temporal scales. Over the timescale of several years, as salt in the cavern dissolves in the water, the cavern starts to erode, causing significant deformations in its overall shape. However, the dissolution process relies on a smaller timescale of several minutes. On the spatial scale, the former involves the modeling of the entire salt cavern, while the latter is more localized and is relevant near the cavern walls. In the present work, we model both these processes in separate simulation setups. A *macroscopic* simulation is carried out to model the evolution of the cavern over many years. This is done on actual salt cavern geometries. The computation of the diffusion rate of the salt (and related minerals) to be used in these macroscopic simulations is done in a separate simulation, in the so-called *microscopic* setup. This involves simulations over the smaller timescale of a few minutes and over representa-

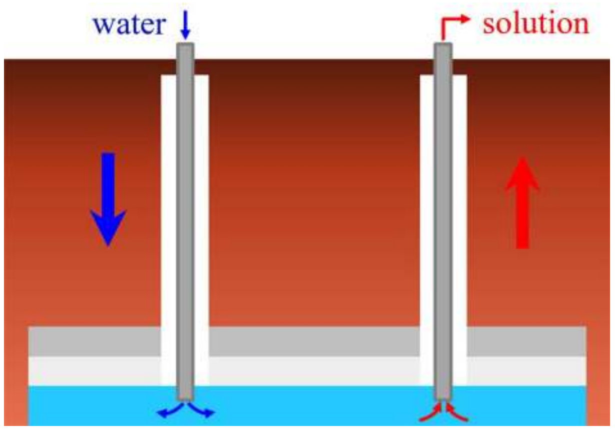

Fig. 1 Schematic of double-well solution mining (adapted from [25])

tive geometries several orders of magnitude smaller than the size of the salt cavern in the macroscopic simulations.

Over the past few decades, meshfree simulation methods have emerged as an alternative to the conventionally used mesh-based simulation procedures, especially in the context of modeling fluid flow. The advantages of meshfree methods are most notable for applications with complex domains, or those with moving geometry parts, free surfaces, phase boundaries, or large deformations. While modeling each of the latter cases with mesh-based methods, the highly dynamic nature of the simulations often requires an expensive global remeshing procedure. On the other hand, moving Lagrangian and semi-Lagrangian procedures fit in naturally with meshfree methods, making the simulation of dynamic geometries or phase boundaries a lot easier. In the application at hand, the modeling of the changing domain during the long-term evolution of the salt cavern falls into this category. We thus choose a meshfree approach.

In this paper, we use a meshfree generalized finite difference method (GFDM) [5,7,12,20] based on a cloud of numerical points. This method has already been successfully applied in various CFD and continuum mechanics applications. Prominent examples include water crossing of cars, water turbines, hydraulic valves, soil mechanics, metal cutting, and mold filling [10,14,21,23,31,32]. The methods presented here are part of the in-house developed software MESHFREE,[1] which combines the advantages of GFDM and the fast linear solvers of SAMG [22].

We start by using a Lagrangian formulation where the discretizing point cloud moves according to the flow velocity [10,13]. This results in an accurate and natural transport of physical information. The basic physical model consists of the conservation equations for mass, momentum, and energy. For solution mining processes, we extend it by the standard k-ε turbulence model and equations for the concentration of the occurring species (see Sect. 2). The GFDM-specific numerics are presented in Sect. 3 with special focus on the Lagrangian and Eulerian formulations. Microscopic simulations are presented in Sect. 4 and are used to determine the necessary effective model parameters of the macroscopic problem which follows. For macroscopic simulations, the Lagrangian formulation leads to a significant restriction of the time step size due to the explicit movement of the point cloud. To enable simulations in reasonable time, an Eulerian formulation should be preferred in this context. Here, the point cloud is fixed and convective terms represent the transport of physical information. The movement of the boundary of the salt cavern is based on the solution rate of the salt in the flowing water. To accurately handle this moving boundary, interior points close to the boundary are subject to an ALE approach (Arbitrary Lagrangian–Eulerian) according to [8]. This procedure gives rise to covering the complete life cycle of a salt cavern—several decades—by a meshfree simulation. In Sect. 5, we demonstrate the advantages of the Eulerian formulation for a simplified macroscopic example of a double-well solution mining process, followed by concluding remarks in Sect. 6.

2 Physical model

In this section, we describe the basic physical flow model and its extensions for modeling solution mining processes, in both the macroscopic and microscopic simulations. Specific models needed for the density, viscosity, and heat capacity of a solution are also discussed.

2.1 Basic equations

The basic underlying physical model is given by the conservation equations of mass, momentum, and energy in Lagrangian formulation.

$$\begin{aligned}
&\frac{d\rho}{dt} + \rho \cdot \nabla^{\mathrm{T}}\mathbf{v} = 0,\\
&\frac{d}{dt}(\rho\cdot\mathbf{v}) + (\rho\cdot\mathbf{v})\cdot\nabla^{\mathrm{T}}\mathbf{v} = \left(\nabla^{\mathrm{T}}\mathbf{S}\right)^{\mathrm{T}} - \nabla p + \rho\cdot\mathbf{g},\\
&\frac{d}{dt}(\rho\cdot E) + (\rho\cdot E)\cdot\nabla^{\mathrm{T}}\mathbf{v} = \nabla^{\mathrm{T}}(\mathbf{S}\cdot\mathbf{v}) - \nabla^{\mathrm{T}}(p\cdot\mathbf{v})\\
&\quad + \rho\cdot\mathbf{g}^{\mathrm{T}}\cdot\mathbf{v} + \nabla^{\mathrm{T}}(\lambda\cdot\nabla T),
\end{aligned} \tag{1}$$

for density ρ, velocity $\mathbf{v}\in\mathbb{R}^3$, stress tensor $\mathbf{S}\in\mathbb{R}^{3\times3}$ (deviatoric part, i.e., $\mathrm{tr}(\mathbf{S})=0$), pressure p, body forces $\mathbf{g}\in\mathbb{R}^3$, total energy $E = c_{\mathrm{v}}\cdot T + \frac{1}{2}\cdot(\mathbf{v}^{\mathrm{T}}\cdot\mathbf{v})$, heat capacity c_{v}, temperature T, and heat conductivity λ. Further, $\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v}^{\mathrm{T}}\nabla$ denotes the material derivative, and $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^{\mathrm{T}}$ denotes the nabla operator.

[1] https://www.meshfree.eu.

In general, the stress tensor is split into its viscous and solid parts by $\mathbf{S} = \mathbf{S}_{\text{visc}} + \mathbf{S}_{\text{solid}}$ [10,13]. For the present application, the stress tensor is purely viscous, $\mathbf{S}_{\text{solid}} = \mathbf{0}$. The viscous part is defined by

$$\mathbf{S}_{\text{visc}} = (\eta + \eta_{\text{turb}}) \cdot \left(\nabla \mathbf{v}^{\mathrm{T}} + (\nabla \mathbf{v}^{\mathrm{T}})^{\mathrm{T}} - \frac{2}{3} \cdot \left(\nabla^{\mathrm{T}} \mathbf{v} \right) \cdot \mathbf{I} \right), \tag{2}$$

where $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity.

To incorporate turbulent effects, the standard k-ε turbulence model [18] is considered for turbulent kinetic energy k and turbulent dissipation ε

$$\begin{aligned} \frac{dk}{dt} &= \frac{1}{\rho} \cdot \nabla^{\mathrm{T}} \left(\left(\eta + \frac{\eta_{\text{turb}}}{\sigma_k} \right) \cdot \nabla k \right) - \varepsilon + \frac{1}{\rho} \cdot (P_{\text{pr}} + P_{\text{b}}), \\ \frac{d\varepsilon}{dt} &= \frac{1}{\rho} \cdot \nabla^{\mathrm{T}} \left(\left(\eta + \frac{\eta_{\text{turb}}}{\sigma_\varepsilon} \right) \cdot \nabla \varepsilon \right) - C_{2\varepsilon} \cdot \frac{\varepsilon^2}{k} \\ &\quad + \frac{1}{\rho} \cdot C_{1\varepsilon} \cdot \frac{\varepsilon}{k} \cdot \left(P_{\text{pr}} + C_{3\varepsilon} \cdot P_{\text{b}} \right), \end{aligned} \tag{3}$$

where η is the laminar viscosity and $\eta_{\text{turb}} = \rho \cdot C_\eta \cdot \frac{k^2}{\varepsilon}$ is the turbulent viscosity. Fluctuating dilatation and source terms are omitted [18]. The turbulent production rate is defined by $P_{\text{pr}} = \eta_{\text{turb}} \cdot \|\nabla \mathbf{v}^{\mathrm{T}}\|_{\mathrm{M}}^2$ with von Mises matrix norm $\|\cdot\|_{\mathrm{M}}$. The turbulent buoyancy is given by $P_{\text{b}} = -\frac{1}{\rho} \cdot \frac{\eta_{\text{turb}}}{\mathrm{Pr}_{\text{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \nabla T)$. For this model, well-established values for the constants are used $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$, $C_{1\varepsilon} = 1.44$, $C_{2\varepsilon} = 1.92$, $C_{3\varepsilon} = -0.33$, $C_\eta = 0.09$, and turbulent Prandtl number $\mathrm{Pr}_{\text{turb}} = 0.85$. Furthermore, a logarithmic wall function is used in the vicinity of walls.

In order to simulate solution mining processes, the above basic model is extended by convection–diffusion equations to represent the different minerals or species present in the salt mixture. For the concentration c_i of species $i = 1, \dots, N$ with effective diffusion coefficient $D_{i,\text{eff}}$, we have

$$\frac{dc_i}{dt} + c_i \cdot \nabla^{\mathrm{T}} \mathbf{v} = \nabla^{\mathrm{T}} (D_{i,\text{eff}} \cdot \nabla c_i). \tag{4}$$

In the Eulerian formulation, the material derivative is replaced by its definition, i.e., $\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{v}^{\mathrm{T}} \nabla$.

2.2 Modeling density, viscosity, and heat capacity

Consider the general form of the equation of state

$$\rho = \rho(T, c_1, \dots, c_N), \tag{5}$$

where density depends on the temperature and each of the concentrations. Based on the formulation in [16,17], the density of a solution of N species in water is given by

$$\rho_{\text{sol}} = \left(\frac{w_{\mathrm{H_2O}}}{\rho_{\mathrm{H_2O}}} + \sum_{i=1}^{N} \frac{w_i}{\rho_{\text{apparent},i}} \right)^{-1}, \tag{6}$$

where $w_{\mathrm{H_2O}}$ and w_i are the mass fraction of water and species i, respectively. Additionally, $w_{\mathrm{H_2O}} + \sum_{i=1}^{N} w_i = 1$ has to be satisfied. The density of water is determined by the nonlinear relation

$$\rho_{\mathrm{H_2O}} = \frac{(((((A_1 \cdot T + A_2) \cdot T + A_3) \cdot T + A_4) \cdot T + A_5) \cdot T + A_6)}{1 + A_7 \cdot T}, \tag{7}$$

with $A_1, \dots, A_7$ and $C_0^*, \dots, C_4^*$ defined according to [17]. The apparent density of species i is given by

$$\rho_{\text{apparent},i} = \frac{(C_0^* \cdot (1 - w_{\mathrm{H_2O}}) + C_1^*) \cdot \exp\left(0.000001 \cdot (T + C_4^*)^2\right)}{(1 - w_{\mathrm{H_2O}}) + C_2^* + C_3^* \cdot T}. \tag{8}$$

The mass fractions w_i are defined by the concentrations c_i as

$$w_i = \frac{c_i}{\sum_{i=1}^{N} c_i + \rho_{\mathrm{H_2O}}}. \tag{9}$$

The viscosity of the solution, $\eta_{\text{sol}}(T, c_1, \dots, c_N)$, and its heat capacity $c_{\text{v,sol}}(T, c_1, \dots, c_N)$ are modeled in a similar manner. For the viscosity of a solution of N species in water, we use a modified version of the Arrhenius equation

$$\eta_{\text{sol}} = \left(\eta_{\mathrm{H_2O}}\right)^{w_{\mathrm{H_2O}}} \prod_{i=1}^{N} (\eta_i)^{w_i}, \tag{10}$$

where the viscosity of water depends on the temperature as

$$\eta_{\mathrm{H_2O}} = \frac{T + 246}{(0.05594 \cdot T + 5.2842)T + 137.37}. \tag{11}$$

Furthermore, the viscosity of species i is given by

$$\eta_i = \frac{\exp\left(\frac{V_1^* (1 - w_{\mathrm{H_2O}})^{V_2^*} + V_3^*}{V_4^* T + 1} \right)}{V_5^* (1 - w_{\mathrm{H_2O}})^{V_6^*} + 1} \tag{12}$$

with constants $V_1^*, \dots, V_6^*$ according to [15].

A weighted summation of the mass fractions is used to obtain the heat capacity of a solution of N species in water

$$c_{\text{v,sol}} = w_{\mathrm{H_2O}} c_{\text{v},\mathrm{H_2O}} + \sum_{i=1}^{N} w_i c_{\text{v},i}. \tag{13}$$

Table 1 Model constants for the sodium chloride solution according to [15–17]

i	A_i	C_i^*	V_i^*	B_i^*
0		-3.2411×10^{-3}		
1	2.8054×10^{-10}	0.0636	16.2217	−0.0694
2	1.0556×10^{-7}	1.0137	1.3229	−0.0782
3	4.6170×10^{-5}	0.0146	1.4849	3.8480
4	7.9870×10^{-3}	3317.3485	0.0075	−11.2762
5	16.9452		30.7802	8.7319
6	999.8385		2.0583	1.8125
7	0.0169			

Furthermore, the heat capacity of species i is modeled by

$$c_{\mathrm{v},i} = B_1^* \exp(a) + B_5^*(1 - w_{\mathrm{H_2O}})^{B_6^*}, \tag{14}$$

where $a = B_2^* T + B_3^* \exp(0.01 \cdot T) + B_4^*(1 - w_{\mathrm{H_2O}})$ and constants $B_1^*, \dots, B_6^*$ are according to [16].

We use quadratic interpolation (extrapolation) for the definition of the heat capacity of water. Assume given temperatures T_1, T_2, T_3, with $T_2 = T_1 + \Delta T$ and $T_3 = T_2 + \Delta T$ ($\Delta T > 0$). The corresponding heat capacities of water c_{v,T_1}, c_{v,T_2}, and c_{v,T_3} are also assumed given. Then, the heat capacity of water at arbitrary temperature T is determined by

$$c_{\mathrm{v,H_2O}} = c_{\mathrm{v},T_1} + (c_{\mathrm{v},T_2} - c_{\mathrm{v},T_1}) \frac{T - T_1}{T_2 - T_1} + \frac{c_{\mathrm{v},T_3} - 2c_{\mathrm{v},T_2} + c_{\mathrm{v},T_1}}{2} \frac{T - T_1}{T_2 - T_1} \left(\frac{T - T_1}{T_2 - T_1} - 1 \right). \tag{15}$$

T_1, T_2, T_3 are chosen adaptively with $\Delta T = 5\,^\circ\mathrm{C}$, depending on the value of T. The range of c_{v,T_k} values, $k = 1, 2, 3$, are taken from [16], which provides the values between $0\,^\circ\mathrm{C}$ and $95\,^\circ\mathrm{C}$.

We restrict the study in this paper to sodium chloride as the species of interest. All the model constants mentioned in this section for sodium chloride are summarized in Table 1.

3 Numerics based on GFDM

3.1 Point cloud preliminaries

In the GFDM approach, the computational domain is discretized by a cloud of numerical points. The point cloud is composed of $NP = NP(t)$ number of points, which includes points in the interior of the domain, and those at the boundary. The initial seeding of these point clouds is done by a meshfree advancing front technique, details of which can be found in [19,24]. The density of the point cloud is given by a sufficiently smooth function $h = h(\mathbf{x}, t)$, the so-called interaction radius or smoothing length. Thus, h prescribes the resolution of the point cloud. It is also used to define the neighborhood of each point. For a point $\mathbf{x}_j$ in the point cloud, all approximations are performed using only nearby points within a distance h from it. This set of nearby points is referred to as the neighborhood or support of $\mathbf{x}_j$ and is denoted by $S_j = \{\mathbf{x}_l : \|\mathbf{x}_l - \mathbf{x}_j\|_2 \le h(\mathbf{x}_j)\}$.

To ensure a sufficient quality of the point cloud, it is ensured that no two points are closer than $r_{\min}h$ apart, and that every sphere of radius $r_{\max}h$ in the domain has at least one point. Thus, the inter-point distance between each point and its nearest neighbor lies in the range $(r_{\min}h, r_{\max}h)$. We follow conventionally used values of these parameters in Lagrangian meshfree GFDM literature, and set $r_{\min} = 0.2$ and $r_{\max} = 0.4$ [4,30]. This results in about 40–50 points in each interior neighborhood, with lesser at and near the boundary.

For the Lagrangian and ALE formulations, the movement of (parts of) the point cloud with the fluid velocity can result in the minimum and maximum inter-point distance criteria being violated. This happens in the form of accumulation or scattering of points which would reduce the quality of the numerical results. To prevent this, points are added in holes containing insufficient points and are merged in regions of accumulation. This method of fixing distortion is entirely local and much cheaper than the remeshing done in mesh-based methods. Details about these procedures of adding and deleting points follow from [4,13,14,24,28].

3.2 Differential operators

GFDMs generalize classical finite differences to arbitrarily spaced point clouds, using a specialized weighted moving least squares approach. Consider a function ϕ defined on each point of the point cloud. At each point $\mathbf{x}_j$, numerical derivatives of ϕ are defined as a linear combination of function values in its neighborhood

$$\partial^* \phi(\mathbf{x}_j) \approx \tilde{\partial}_j^* \phi = \sum_{l \in S_j} c_{jl}^* \phi_l, \tag{16}$$

where $* = x, y, z, \Delta, \dots$ denotes the derivative of interest, ∂^* is the continuous differential operator, $\tilde{\partial}_j^*$ is the numerical differential operator at point $\mathbf{x}_j$, and $\phi_l = \phi(\mathbf{x}_l)$. The numerical differential operators are thus given by the coefficients c_{jl}^*, which are independent of the function being differentiated. They are computed by a norm minimization process that ensures that monomials up to a specified order are differentiated exactly.

$$\sum_{l \in S_j} c_{jl}^* m_l = \partial_j^* m, \qquad \forall m \in \mathcal{M},$$

$$\min \sum_{l \in S_j} \left(\frac{c^*_{jl}}{W_{jl}} \right)^2 , \tag{17}$$

where $\mathcal{M}$ is the set of monomials being differentiated exactly. To compute the Laplacian, the monomials are complemented by the delta function to control the central stencil value c^{Δ}_{jj}, which improves stability in the pressure Poisson equations [26]. In the present work, we consider monomials up to the order of 2. The weighting function W is defined such that neighboring points with the smallest distance to the considered point obtain the highest weight. In the present work, we use a truncated Gaussian weighting function

$$W_{jl} = \begin{cases} \exp\left(-c_W \frac{\|\mathbf{x}_j - \mathbf{x}_l\|^2}{h_j^2 + h_l^2}\right), & \text{if } \mathbf{x}_l \in S_j \\ 0, & \text{elsewhere} \end{cases} \tag{18}$$

for a constant $c_W > 0$. We note that the same differential operators as defined above can also be equivalently derived by minimizing errors in Taylor expansions [26].

Using this procedure, we compute numerical gradient operators and a numerical Laplacian. For more details on the computation of the differential operators, we refer to [4,14,26].

3.3 Time integration

3.3.1 Lagrangian formulation

A strong form discretization of the physical model (Sect. 2) is done using the numerical differential operators defined above and a chosen time integration scheme. For simplicity, the following considerations are based on a first-order time integration.

Starting with the Lagrangian formulation, equations (1) can be rewritten as

$$\begin{aligned} \frac{d\rho}{dt} &= -\rho \cdot \nabla^{\mathrm{T}}\mathbf{v}, \\ \frac{d\mathbf{v}}{dt} &= \frac{1}{\rho} \cdot \left(\nabla^{\mathrm{T}}\mathbf{S}\right)^{\mathrm{T}} - \frac{1}{\rho} \cdot \nabla p + \mathbf{g}, \\ (\rho \cdot c_{\mathrm{v}}) \cdot \frac{dT}{dt} &= \nabla^{\mathrm{T}}(\mathbf{S} \cdot \mathbf{v}) - \left(\nabla^{\mathrm{T}}\mathbf{S}\right) \cdot \mathbf{v} \\ &- p \cdot \nabla^{\mathrm{T}}\mathbf{v} + \nabla^{\mathrm{T}}(\lambda \cdot \nabla T). \end{aligned} \tag{19}$$

To improve readability, we henceforth use the shorthand $\rho = \rho_{\mathrm{sol}}$ and $c_{\mathrm{v}} = c_{\mathrm{v,sol}}$.

Together with Eqs. (2)–(4), this is the starting point of the numerical discretization. The continuous spatial derivatives are replaced by their least squares approximated counterparts described in Sect. 3.2. We consider the superscript $n+1$ to denote the next time level and n for the current one, giving the time step size $\Delta t = t^{n+1} - t^n$. Below, we explain each of the steps of the discretization in the Lagrangian formulation for the microscopic-scale simulations. Most of the steps are the same also for the macroscopic-scale simulations, and the few differences are explained in Sect. 5.

Step 1 Point cloud movement

The discretization procedure begins by moving the point cloud according to a second-order method [27] by

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \cdot \mathbf{v}^n + \frac{1}{2} \frac{\mathbf{v}^n - \mathbf{v}^{n-1}}{\Delta t_0} \cdot (\Delta t)^2 , \tag{20}$$

with previous time step size $\Delta t_0 = t^n - t^{n-1}$.

Step 2 Temperature

A semi-implicit time integration is then carried out to compute the new temperature T^{n+1} by

$$(\mathbf{I}_T + \mathbf{D}_T) \cdot T^{n+1} = \left(\rho^n \cdot c^n_{\mathrm{v}}\right) \cdot T^n + f_T, \tag{21}$$

with

$$\begin{aligned} \mathbf{I}_T &= \rho^n \cdot c^n_{\mathrm{v}} \cdot \mathbf{I}, \\ \mathbf{D}_T &= -\Delta t \cdot \tilde{\nabla}^{\mathrm{T}}\left(\lambda \cdot \tilde{\nabla}\right), \\ f_T &= \Delta t \cdot \left(\tilde{\nabla}^{\mathrm{T}}(\mathbf{S}^n \cdot \mathbf{v}^n) - \left(\tilde{\nabla}^{\mathrm{T}}\mathbf{S}^n\right) \cdot \mathbf{v}^n - p^n \cdot \tilde{\nabla}^{\mathrm{T}}\mathbf{v}^n\right), \end{aligned} \tag{22}$$

where the overhead $\sim$ indicates the discrete differential operators.

To simplify notation, the index of the points has been omitted. Equation (21) forms a sparse linear system of equations with unknowns T^{n+1} at each point of the point cloud. All sparse implicit linear systems arising in this and the coming steps are solved with a BiCGSTAB solver, without the use of a preconditioner.

Step 3 Concentrations

A similar procedure as that done for the temperature is carried out for the concentrations. We use a semi-implicit time integration for the concentration of each species c^{n+1}_i, $i = 1, \ldots, N$,

$$\left(\mathbf{I}_{c_i} + \mathbf{D}_{c_i}\right) \cdot c^{n+1}_i = c^n_i , \tag{23}$$

with

$$\begin{aligned} \mathbf{I}_{c_i} &= (\mathbf{I} + \Delta t \cdot \tilde{\nabla}^{\mathrm{T}}\mathbf{v}^n), \\ \mathbf{D}_{c_i} &= -\Delta t \cdot \tilde{\nabla}^{\mathrm{T}}(D_{i,\mathrm{eff}} \cdot \tilde{\nabla}). \end{aligned} \tag{24}$$

Step 4 ρ, η, and c_{v}

The updated density ρ^{n+1}, viscosity $\eta^{n+1}_{\mathrm{sol}}$, and heat capacity c^{n+1}_{v} are then determined according to the definitions in Sect. 2.2.

Using the updated solution viscosity, a preliminary viscosity for the momentum equation is computed as $\hat{\eta}^{n+1} = \eta_{\text{sol}}^{n+1} + \eta_{\text{turb}}^{n}$.

Step 5 Hydrostatic pressure

The pressure is split into its hydrostatic (body forces) and dynamic parts (movement of the fluid) as

$$p = p_{\text{hyd}} + p_{\text{dyn}}. \tag{25}$$

First the updated hydrostatic pressure p_{hyd}^{n+1} is computed

$$\tilde{\nabla}^{\text{T}} \left(\frac{1}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\text{hyd}}^{n+1} \right) = \tilde{\nabla}^{\text{T}} \mathbf{g}. \tag{26}$$

Using the updated hydrostatic pressure, a pressure guess $\hat{p}$ is computed which will be used while computing the new velocity

$$\hat{p} = p_{\text{hyd}}^{n+1} + p_{\text{dyn}}^{n}. \tag{27}$$

Step 6 Coupled velocity–pressure

Time integration of the first equation in (19) provides the targeted divergence of velocity $\tilde{\nabla}^{\text{T}} \mathbf{v}^{n+1}$. To solve for $\mathbf{v}^{n+1}$ and p^{n+1} in an implicit time integration scheme, we use the penalty formulation introduced in [10,13]. Using the pressure guess defined above, we obtain the following coupled velocity–pressure system for preliminary velocity $\hat{\mathbf{v}}^{n+1}$ and correction pressure p_{corr}^{n+1}:

$$\begin{aligned} &\left(\mathbf{I} - \frac{\Delta t}{\rho^{n+1}} \cdot \tilde{\psi}_{\hat{\eta}^{n+1}}^{n+1} \right) \cdot \hat{\mathbf{v}}^{n+1} + \frac{\Delta t}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\text{corr}}^{n+1} \\ &\quad = \mathbf{v}^{n} - \frac{\Delta t}{\rho^{n+1}} \cdot \tilde{\nabla} \hat{p} + \Delta t \cdot \mathbf{g}, \\ &\tilde{\nabla}^{\text{T}} \left(\frac{\Delta t_{\text{virt}}}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\text{corr}}^{n+1} \right) = \tilde{\nabla}^{\text{T}} \hat{\mathbf{v}}^{n+1} - \tilde{\nabla}^{\text{T}} \mathbf{v}^{n+1}, \end{aligned} \tag{28}$$

with

$$\begin{aligned} \left(\tilde{\psi}_{\hat{\eta}^{n+1}}^{n+1} \right)^{\text{T}} &= \tilde{\nabla}^{\text{T}} \left(\hat{\eta}^{n+1} \cdot \tilde{\nabla} \right) \left(\hat{\mathbf{v}}^{n+1} \right)^{\text{T}} \\ &\quad + (\tilde{\nabla} \hat{\eta}^{n+1})^{\text{T}} \cdot \left(\tilde{\nabla} \left(\hat{\mathbf{v}}^{n+1} \right)^{\text{T}} \right)^{\text{T}} \\ &\quad + \frac{\hat{\eta}^{n+1}}{3} \cdot \left(\tilde{\nabla} \left(\tilde{\nabla}^{\text{T}} \hat{\mathbf{v}}^{n+1} \right) \right)^{\text{T}} \\ &\quad - \frac{2}{3} \cdot \left(\tilde{\nabla}^{\text{T}} \hat{\mathbf{v}}^{n+1} \right) \cdot \left(\tilde{\nabla} \hat{\eta}^{n+1} \right)^{\text{T}}, \end{aligned} \tag{29}$$

and $\Delta t_{\text{virt}} = A_{\text{virt}} \cdot \Delta t$, $0 \leq A_{\text{virt}} \leq 1$. If $A_{\text{virt}} = 1$, the scheme corresponds to an implicit Chorin projection; see [3]. Theoretically, choosing $A_{\text{virt}} = 0$ would give the exact solution. However, the linear system is ill-conditioned and cannot be solved in most cases. For $0.001 \leq A_{\text{virt}} \leq 0.1$, conditioning of the linear system is sufficiently good. Furthermore, the resulting preliminary velocity features a divergence which is very close to the targeted one. We note that in Eqs. (28) and (29), the stress tensor $\mathbf{S}^{n+1}$ was determined according to Eq. (2).

Step 7 Update velocity and pressure

The updates of velocity and dynamic pressure are given by

$$\begin{aligned} \mathbf{v}^{n+1} &= \hat{\mathbf{v}}^{n+1} - \frac{\Delta t_{\text{virt}}}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\text{corr}}^{n+1}, \\ p_{\text{dyn}}^{n+1} &= p_{\text{dyn}}^{n} + p_{\text{corr}}^{n+1}. \end{aligned} \tag{30}$$

Step 8 Turbulence

For the k-ε turbulence model, we derive a singularity formulation from Eq. (3):

$$\begin{aligned} \frac{d}{dt} \left(\frac{k}{\varepsilon} \right) &= (C_{2\varepsilon} - 1) + C_{\eta} \cdot (1 - C_{1\varepsilon}) \cdot \|\tilde{\nabla} \mathbf{v}^{\text{T}}\|_{\text{M}}^{2} \cdot \left(\frac{k}{\varepsilon} \right)^{2} \\ &\quad + \frac{C_{\eta} \cdot (C_{1\varepsilon} \cdot C_{3\varepsilon} - 1)}{\rho \cdot \text{Pr}_{\text{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \tilde{\nabla} T) \cdot \left(\frac{k}{\varepsilon} \right)^{2} \\ &\quad + \frac{1}{\rho} \cdot \tilde{\Delta}_{\eta^*} \left(\frac{k}{\varepsilon} \right), \\ \frac{d}{dt} \left(\frac{\varepsilon}{k} \right) &= (1 - C_{2\varepsilon}) \cdot \left(\frac{\varepsilon}{k} \right)^{2} + C_{\eta} \cdot (C_{1\varepsilon} - 1) \cdot \|\tilde{\nabla} \mathbf{v}^{\text{T}}\|_{\text{M}}^{2} \\ &\quad + \frac{C_{\eta} \cdot (1 - C_{1\varepsilon} \cdot C_{3\varepsilon})}{\rho \cdot \text{Pr}_{\text{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \tilde{\nabla} T) \\ &\quad + \frac{1}{\rho} \cdot \tilde{\Delta}_{\eta^*} \left(\frac{\varepsilon}{k} \right), \end{aligned} \tag{31}$$

where

$$\begin{aligned} \tilde{\Delta}_{\eta^*} \left(\frac{k}{\varepsilon} \right) &= \frac{\varepsilon \cdot \tilde{\Delta}_{\eta_k} k - k \cdot \tilde{\Delta}_{\eta_\varepsilon} \varepsilon}{\varepsilon^2}, \\ \tilde{\Delta}_{\eta^*} \left(\frac{\varepsilon}{k} \right) &= \frac{k \cdot \tilde{\Delta}_{\eta_\varepsilon} \varepsilon - \varepsilon \cdot \tilde{\Delta}_{\eta_k} k}{k^2} \end{aligned} \tag{32}$$

with

$$\begin{aligned} \tilde{\Delta}_{\eta_k} &= \tilde{\nabla}^{\text{T}} \left(\left(\eta + \frac{\eta_{\text{turb}}}{\sigma_k} \right) \cdot \tilde{\nabla} \right), \\ \tilde{\Delta}_{\eta_\varepsilon} &= \tilde{\nabla}^{\text{T}} \left(\left(\eta + \frac{\eta_{\text{turb}}}{\sigma_\varepsilon} \right) \cdot \tilde{\nabla} \right). \end{aligned} \tag{33}$$

If $k, \varepsilon > 0$ for all $t^n \leq t \leq t^{n+1}$, numerical mean values can be determined from (31):

$$\begin{aligned} \left. \frac{k}{\varepsilon} \right|_{\text{m}} &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \frac{d}{dt} \left(\frac{k}{\varepsilon} \right) dt, \\ \left. \frac{\varepsilon}{k} \right|_{\text{m}} &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \frac{d}{dt} \left(\frac{\varepsilon}{k} \right) dt. \end{aligned} \tag{34}$$

We use the mean values to avoid singularities in the discretized k-ε turbulence model.

$$\begin{aligned}\frac{dk}{dt} &= \frac{\tilde{\Delta}_{\eta_k} k}{\rho} - \left.\frac{\varepsilon}{k}\right|_{\mathrm{m}} \cdot k + C_\eta \cdot P_{\mathrm{prb},k} \cdot \left.\frac{k}{\varepsilon}\right|_{\mathrm{m}} \cdot k, \\ \frac{d\varepsilon}{dt} &= \frac{\tilde{\Delta}_{\eta_\varepsilon} \varepsilon}{\rho} - C_{2\varepsilon} \cdot \left.\frac{\varepsilon}{k}\right|_{\mathrm{m}} \cdot \varepsilon + C_{1\varepsilon} \cdot C_\eta \cdot P_{\mathrm{prb},\varepsilon} \cdot \left.\frac{k}{\varepsilon}\right|_{\mathrm{m}} \cdot \varepsilon,\end{aligned} \tag{35}$$

where

$$\begin{aligned}P_{\mathrm{prb},k} &= \|\tilde{\nabla}\mathbf{v}^{\mathrm{T}}\|_{\mathrm{M}}^2 - \frac{1}{\rho \cdot \mathrm{Pr}_{\mathrm{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \tilde{\nabla} T), \\ P_{\mathrm{prb},\varepsilon} &= \|\tilde{\nabla}\mathbf{v}^{\mathrm{T}}\|_{\mathrm{M}}^2 - \frac{C_{3\varepsilon}}{\rho \cdot \mathrm{Pr}_{\mathrm{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \tilde{\nabla} T).\end{aligned} \tag{36}$$

A fully implicit time integration scheme for the turbulent kinetic energy k^{n+1} can now be developed as

$$\begin{aligned}&k^{n+1} - \frac{\Delta t \cdot \tilde{\Delta}_{\eta_k} k^{n+1}}{\rho} + \Delta t \cdot \left.\frac{\varepsilon}{k}\right|_{\mathrm{m}} \cdot k^{n+1} \\ &\quad - \Delta t \cdot C_\eta \cdot P_{\mathrm{prb},k}^{n+1} \cdot \left.\frac{k}{\varepsilon}\right|_{\mathrm{m}} \cdot k^{n+1} = k^n.\end{aligned} \tag{37}$$

A similar procedure is used to compute the updated turbulent dissipation

$$\begin{aligned}&\varepsilon^{n+1} - \frac{\Delta t \cdot \tilde{\Delta}_{\eta_\varepsilon} \varepsilon^{n+1}}{\rho} + \Delta t \cdot C_{2\varepsilon} \cdot \left.\frac{\varepsilon}{k}\right|_{\mathrm{m}} \cdot \varepsilon^{n+1} \\ &\quad - \Delta t \cdot C_{1\varepsilon} \cdot C_\eta \cdot P_{\mathrm{prb},\varepsilon}^{n+1} \cdot \left.\frac{k}{\varepsilon}\right|_{\mathrm{m}} \cdot \varepsilon^{n+1} = \varepsilon^n.\end{aligned} \tag{38}$$

The mean values are determined analytically. This is illustrated in detail for $\left.\frac{k}{\varepsilon}\right|_{\mathrm{m}}$. Assuming that the diffusion term $\frac{1}{\rho} \cdot \tilde{\Delta}_{\eta^*}\left(\frac{k}{\varepsilon}\right)$ is negligible as well as defining

$$\begin{aligned}x &= \frac{k}{\varepsilon}, \quad a = C_{2\varepsilon} - 1, \\ b &= C_\eta \cdot (C_{1\varepsilon} - 1) \cdot \|\tilde{\nabla}\mathbf{v}^{\mathrm{T}}\|_{\mathrm{M}}^2 \\ &\quad + \frac{C_\eta \cdot (1 - C_{1\varepsilon} \cdot C_{3\varepsilon})}{\rho \cdot \mathrm{Pr}_{\mathrm{turb}}} \cdot \frac{\partial \rho}{\partial T} \cdot (\mathbf{g} \cdot \tilde{\nabla} T),\end{aligned}$$

we can rewrite equation (31) as

$$\frac{dx}{dt} = a - b \cdot x^2. \tag{39}$$

For $x_0 = \sqrt{\frac{a}{b}}$, we obtain

$$x^{n+1} = \begin{cases} x_0 \cdot \tanh\left(\Delta t \cdot \sqrt{a \cdot b} + \operatorname{arctanh}\left(\frac{x^n}{x_0}\right)\right), & x^n < x_0 \\ x_0, & x^n = x_0 \\ x_0 \cdot \coth\left(\Delta t \cdot \sqrt{a \cdot b} + \operatorname{arccoth}\left(\frac{x^n}{x_0}\right)\right), & x^n > x_0 \end{cases}. \tag{40}$$

Finally, the updated turbulent viscosity is determined by

$$\eta_{\mathrm{turb}}^{n+1} = \rho^{n+1} \cdot C_\eta \cdot \frac{(k^{n+1})^2}{\varepsilon^{n+1}}. \tag{41}$$

3.3.2 Eulerian formulation

In case of the Eulerian formulation, [25] shows that a second-order time integration scheme should be applied to numerically solve transport terms of the form $\mathbf{v}^{\mathrm{T}}\nabla$ in the GFDM context. For this purpose, the SDIRK2 method is proposed (see [1]), which features the same stability properties as an implicit Euler time integration scheme. Furthermore, an upwind discretization by means of a MUSCL reconstruction with a Superbee limiter is used.

The majority of the steps are the same as those carried out in the Lagrangian formulation. The movement step of the Lagrangian formulation is skipped here. And the coupled velocity–pressure system is modified to the following two-step procedure:

$$\begin{aligned}&\left(\mathbf{I}_{\hat{\mathbf{v}}^{n+\alpha}} - \frac{\alpha \cdot \Delta t}{\rho^{n+\alpha}} \tilde{\psi}_{\hat{\eta}^{n+\alpha}}^{n+\alpha}\right) \cdot \hat{\mathbf{v}}^{n+\alpha} + \frac{\alpha \cdot \Delta t}{\rho^{n+\alpha}} \cdot \tilde{\nabla} p_{\mathrm{corr}}^{n+\alpha} \\ &\quad = \mathbf{v}^n - \frac{\alpha \cdot \Delta t}{\rho^{n+\alpha}} \cdot \tilde{\nabla} \hat{p} + \alpha \cdot \Delta t \cdot \mathbf{g}, \\ &\quad \tilde{\nabla}^{\mathrm{T}} \left(\frac{\Delta t_{\mathrm{virt}}}{\rho^{n+\alpha}} \cdot \tilde{\nabla} p_{\mathrm{corr}}^{n+\alpha}\right) = \tilde{\nabla}^{\mathrm{T}} \hat{\mathbf{v}}^{n+\alpha} - \tilde{\nabla}^{\mathrm{T}} \mathbf{v}^{n+\alpha},\end{aligned} \tag{42}$$

with

$$\begin{aligned}\mathbf{I}_{\hat{\mathbf{v}}^{n+\alpha}} &= \left(\mathbf{I} + \alpha \cdot \Delta t \cdot \left(\widetilde{\mathbf{v}^{\mathrm{T}}\nabla}\right) \hat{\mathbf{v}}^{n+\alpha}\right), \\ \left(\tilde{\psi}_{\hat{\eta}^{n+\alpha}}^{n+\alpha}\right)^{\mathrm{T}} &= \tilde{\nabla}^{\mathrm{T}} \left(\hat{\eta}^{n+\alpha} \cdot \tilde{\nabla}\right) \left(\hat{\mathbf{v}}^{n+\alpha}\right)^{\mathrm{T}} \\ &\quad + \left(\tilde{\nabla}\hat{\eta}^{n+\alpha}\right)^{\mathrm{T}} \cdot \left(\tilde{\nabla} \left(\hat{\mathbf{v}}^{n+\alpha}\right)^{\mathrm{T}}\right)^{\mathrm{T}} \\ &\quad + \frac{\hat{\eta}^{n+\alpha}}{3} \cdot \left(\tilde{\nabla} \left(\tilde{\nabla}^{\mathrm{T}} \hat{\mathbf{v}}^{n+\alpha}\right)\right)^{\mathrm{T}} \\ &\quad - \frac{2}{3} \cdot \left(\tilde{\nabla}^{\mathrm{T}} \hat{\mathbf{v}}^{n+\alpha}\right) \cdot \left(\tilde{\nabla}\hat{\eta}^{n+\alpha}\right)^{\mathrm{T}},\end{aligned} \tag{43}$$

and $\alpha = 1 - \frac{\sqrt{2}}{2}$. Density and viscosity for the intermediate step can for instance be determined by linear interpolation between time levels n and $n+1$.

In the second step, the preliminary velocity is determined as

$$\hat{\mathbf{v}}^{n+1} - \Delta t \cdot \alpha \cdot \mathbf{V}\left(\hat{\mathbf{v}}^{n+1}, p_{\mathrm{corr}}^{n+1}\right)$$

$$= \mathbf{v}^n + \Delta t \cdot (1-\alpha) \cdot \mathbf{V}(\hat{\mathbf{v}}^{n+\alpha}, p_{\mathrm{corr}}^{n+\alpha}),$$
$$\tilde{\nabla}^{\mathrm{T}} \left(\frac{\Delta t_{\mathrm{virt}}}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\mathrm{corr}}^{n+1} \right) = \tilde{\nabla}^{\mathrm{T}} \hat{\mathbf{v}}^{n+1} - \tilde{\nabla}^{\mathrm{T}} \mathbf{v}^{n+1} \tag{44}$$

with

$$\begin{aligned} \mathbf{V}\left(\hat{\mathbf{v}}^{n+1}, p_{\mathrm{corr}}^{n+1}\right) &= -\frac{1}{\rho^{n+1}} \cdot \left(\widetilde{\mathbf{v}^{\mathrm{T}}\nabla}\right) \hat{\mathbf{v}}^{n+1} + \frac{1}{\rho^{n+1}} \cdot \tilde{\psi}_{\hat{\eta}^{n+1}}^{n+1} \\ &\quad - \frac{1}{\rho^{n+1}} \cdot \tilde{\nabla} \hat{p}^{n+1} - \frac{1}{\rho^{n+1}} \cdot \tilde{\nabla} p_{\mathrm{corr}}^{n+1} + \mathbf{g}, \\ \mathbf{V}\left(\hat{\mathbf{v}}^{n+\alpha}, p_{\mathrm{corr}}^{n+\alpha}\right) &= \frac{\hat{\mathbf{v}}^{n+\alpha} - \mathbf{v}^n}{\alpha \cdot \Delta t}. \end{aligned} \tag{45}$$

3.3.3 Further details

For more details on the Eulerian procedure, we refer to [25], and for similar GFDM Eulerian formulations, we refer to [29].

For numerical validations of the velocity–pressure scheme used here, their implementations within a GFDM framework, and a comparison of GFDM results with other numerical methods on benchmark problems, we refer to our earlier work [4,10,13,21,24,25,30].

4 Microscopic scale

To study the smaller-scale (both spatially and temporally) dissolution of the salt species in the water, we consider representative geometries of the salt cavern in a so-called microscopic setup. In this section, we identify effective parameters of the dissolution process. Specifically, we compute the effective diffusion coefficient and the effective transition coefficient between water and surrounding species. These will be used later, in Sect. 5, in the macroscopic procedure to simulate the overall evolution of the salt cavern. The Lagrangian formulation is used here. The time integration of the underlying equations is done as presented in Sect. 3.3.1.

For the sake of brevity, we restrict the following description to sodium chloride as the species of interest. The same procedure can directly be transferred to any other species.

4.1 Setup

We consider a cylinder with diameter of 5 m and height of 10 m which is initially filled with pure water, i.e., $c_{\mathrm{NaCl}}(t = 0) = 0$. During the simulation, the temperature is fixed to $T_0 = 20\,^{\circ}\mathrm{C}$.

The roof of the cylinder acts as an inexhaustible supply of sodium chloride which is modeled by applying a Dirichlet condition with saturation concentration

$$c_{\mathrm{NaCl}}^{\mathrm{s}} = c_{\mathrm{NaCl}}^{\mathrm{s}}(T_0) = 357 \frac{\mathrm{kg}}{\mathrm{m}^3}. \tag{46}$$

For the hull of the cylinder, a homogeneous Neumann condition is applied. Aiming at a quasi-steady state, the bottom of the cylinder models an outflow boundary. In the interior, we solve

$$\frac{dc_{\mathrm{NaCl}}}{dt} + c_{\mathrm{NaCl}} \cdot \nabla^{\mathrm{T}} \mathbf{v} = \nabla^{\mathrm{T}}(D_{\mathrm{micro}} \cdot \nabla c_{\mathrm{NaCl}}), \tag{47}$$

where $D_{\mathrm{micro}} = D_{\mathrm{laminar}} + D_{\mathrm{turb}}$. The laminar diffusion coefficient for sodium chloride is given by $D_{\mathrm{laminar}} = 1.611 \cdot 10^{-9} \frac{\mathrm{m}^2}{\mathrm{s}}$ (see [6]). For the turbulent part, we have

$$D_{\mathrm{turb}} = C_\eta \cdot \frac{k^2}{\varepsilon}. \tag{48}$$

Standard boundary conditions (Dirichlet and Neumann) are prescribed for velocity, pressure, and the turbulent quantities. The simulation runs until a quasi-steady state is reached, which will be explained below.

4.2 Evaluation strategy

In order to determine the effective quantities, the cylinder is split in the axial direction (z-direction) into equal sub-cylinders SC_j, $j = 1, \ldots, J$. These are used to estimate the mass flow. The planes between the sub-cylinders are denoted by help-planes HP_j, $j = 1, \ldots, J-1$.

We note that the moving Lagrangian nature of the simulations means that point locations are always changing in each time step, except in the trivial case when $\mathbf{v} = 0$ which does not occur here. Thus, a true steady state never occurs. Rather, simulations run till a quasi-steady state is reached, which is determined by the averaged values of the mass flow in the sub-cylinders. A quasi-steady state is said to be reached when the relative change of the mass flow in each of the sub-cylinders is within a tolerance specified (here, 10^{-4}) for 5 consecutive time steps.

4.2.1 Effective diffusion coefficient

The mass flow of sodium chloride is given by

$$\frac{dm}{dt} = -D_{\mathrm{NaCl,eff}} \cdot \frac{\partial \bar{c}_{\mathrm{NaCl}}}{\partial \mathbf{n}}, \tag{49}$$

where $\bar{c}_{\mathrm{NaCl}}$ is the mean concentration. The mass flow and the mean concentration in sub-cylinder SC_j are determined by

$$\begin{aligned} \frac{dm}{dt}(SC_j) &= \frac{\int_{SC_j} c_{\mathrm{NaCl}} \cdot v_3 \, dV_{SC_j}}{\int_{SC_j} 1 \, dV_{SC_j}}, \\ \bar{c}_{\mathrm{NaCl}}(SC_j) &= \frac{\int_{SC_j} c_{\mathrm{NaCl}} \, dV_{SC_j}}{\int_{SC_j} 1 \, dV_{SC_j}}. \end{aligned} \tag{50}$$

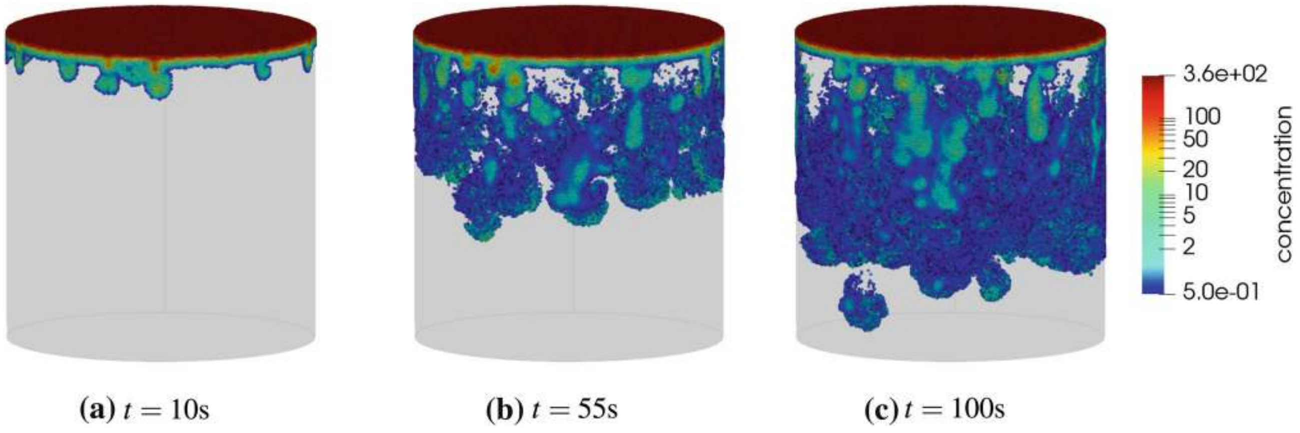

Fig. 2 Evolution of concentration in the microscopic simulation for interaction radius $h = 0.18$ m (Lagrangian formulation)

Based on the mean concentration in a sub-cylinder SC_j, we can approximate its normal derivative with respect to the help plane HP_j. This yields the effective diffusion coefficients in each sub-cylinder

$$D_{\text{NaCl,eff}}(SC_j|HP_j) = -\frac{\frac{dm}{dt}(SC_j)}{\left.\frac{\partial \bar{c}_{\text{NaCl}}}{\partial \mathbf{n}}\right|_{HP_j}}, \quad j = 1, \ldots, J-1. \tag{51}$$

Once a quasi-steady state is reached, an overall effective diffusion coefficient can be determined. To accommodate the "quasi-steady" character of the simulation, we use a time-averaged effective diffusion coefficient, over a small time interval, and over each of the sub-cylinders. This value will later be used in the macroscopic setup.

4.2.2 Effective transition coefficient

The effective transition coefficient $\gamma_{\text{NaCl,eff}}$ is derived in a manner similar to that done for the effective diffusion coefficient $D_{\text{NaCl,eff}}$.

$$\gamma_{\text{NaCl,eff}}\left(SC_j\right) = -\frac{\frac{dm}{dt}\left(SC_j\right)}{c^{\text{s}}_{\text{NaCl}} - \bar{c}_{\text{NaCl}}\left(SC_j\right)}, \quad j = 1, \ldots, J-1. \tag{52}$$

Once again, the time-averaged values of the effective transition coefficient in each of the sub-cylinders at the quasi-steady state give the overall effective transition coefficient which will be used in the macroscopic simulations in Sect. 5.

4.2.3 Effective solution rate

With the help of $\gamma_{\text{NaCl,eff}}$, we can define the solution rate of sodium chloride for given temperature T_0 by

$$R_{\text{NaCl}}(T_0) = \gamma_{\text{NaCl,eff}}\left(c^{\text{s}}_{\text{NaCl}} - c_{\text{NaCl}}\right). \tag{53}$$

4.3 Numerical results

In the simulations carried out, we choose $J = 10$ to divide the cylinder domain considered into 10 sub-cylinders of height 1 m each. We consider several levels of resolution to study the convergence of the effective parameters being determined to resolution-independent values. The coarsest resolution used is $h = 0.8$ m corresponding to 70,400 points in the domain. h is consecutively halved till $h = 0.1$ m corresponding to 20,892,871 points in the domain. Several resolutions in between are also considered to better illustrate the converged values of the effective parameters. We note that the number of points mentioned here are at the initial time of the simulation ($t = 0$). This number of points will slightly vary in time due to the addition and deletion of points explained in Sect. 3.1.

The evolution of the concentration for $h = 0.18$ m is illustrated in Fig. 2 in the time interval [0 s, 100 s]. As expected, the flow is characterized by viscous fingering.

The convergence of effective diffusion as well as transition coefficient with decreasing h is shown in Figs. 3 and 4, respectively. The values plotted are also tabulated in Table 2, along with the relation between the interaction radius h and the number of points in the domain NP. The time step size is governed by $\Delta t = CFL_{\text{Lag}} \cdot \frac{h}{|\mathbf{v}|}$, with CFL_{Lag} set to 0.2. The diffusion coefficient converges to $D_{\text{NaCl,eff}} = 0.1\,\frac{\text{m}^2}{\text{s}}$, while the transition coefficient converges to $\gamma_{\text{NaCl,eff}} = 0.000042\,\frac{\text{m}}{\text{s}}$. Using equation (53), we obtain a maximum solution rate of $R_{\text{NaCl,max}}(20\,°\text{C}) = 0.0150\,\frac{\text{kg}}{\text{m}^2\cdot\text{s}}$.

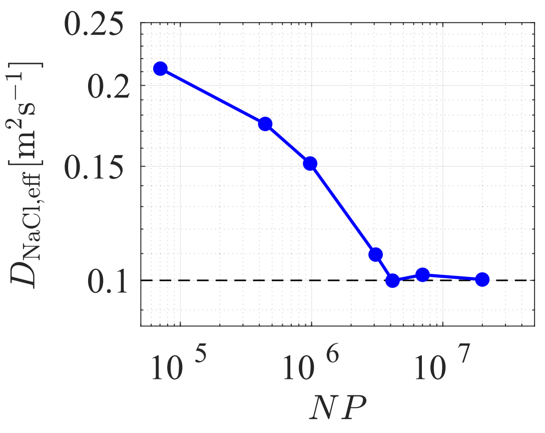

Fig. 3 Convergence of the effective diffusion coefficient in the microscopic simulations. *NP* denotes the number of the points in the initial domain

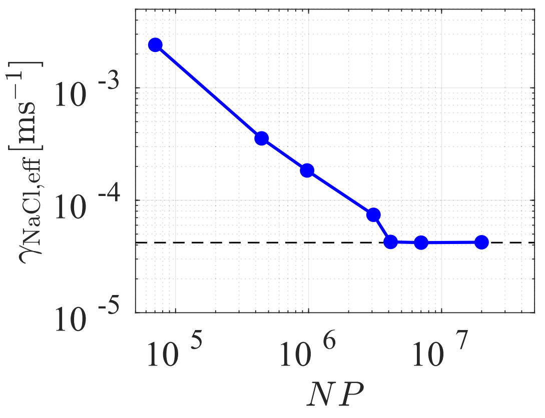

Fig. 4 Convergence of the effective transition coefficient in the microscopic simulations. *NP* denotes the number of the points in the initial domain

Table 2 Estimated effective diffusion D $[\frac{\mathrm{m}^2}{\mathrm{s}}]$, transition coefficient γ $[\frac{\mathrm{m}}{\mathrm{s}}]$, and maximum solution rate R $[\frac{\mathrm{kg}}{\mathrm{m}^2\cdot\mathrm{s}}]$, with varying interaction radius h [m] and corresponding initial number of points *NP*

h	NP	$D_{\mathrm{NaCl,eff}}$	$\gamma_{\mathrm{NaCl,eff}}$	$R_{\mathrm{NaCl,max}}$ (20 °C)
0.80	70,400	0.2123	2.41×10^{-3}	0.8604
0.40	443,318	0.1744	3.55×10^{-4}	0.1267
0.30	974,918	0.1515	1.84×10^{-4}	0.0657
0.20	3,076,350	0.1096	7.42×10^{-5}	0.0264
0.18	4,139,040	0.0999	4.26×10^{-5}	0.0152
0.15	7,014,494	0.1020	4.19×10^{-5}	0.0150
0.10	20,892,871	0.1003	4.23×10^{-5}	0.0151

Compared to the solution rate of $0.0488\frac{\mathrm{kg}}{\mathrm{m}^2\cdot\mathrm{s}}$ for $T_0 = 23\,^\circ\mathrm{C}$ determined in [11] at a crystal level, the estimated solution rate is of the correct order of magnitude.

5 Macroscopic scale

We now model the overall evolution of the salt cavern during the double-well solution mining process. Both the Lagrangian and the Eulerian formulation are evaluated for this.

The model equations and time integration procedures for the macroscopic-scale simulations are the same as those described in Sects. 2 and 3.3, respectively, with a few variations. Firstly, the dissolution of the salt into the water occurs at much smaller spatial and temporal scales than those used here. To take this into account, the dissolution process of the salt at the cavern walls is modeled using a Robin boundary condition for the concentration

$$D_{\mathrm{NaCl,eff}} \cdot \frac{\partial c_{\mathrm{NaCl}}}{\partial \mathbf{n}} = \gamma_{\mathrm{NaCl,eff}} \cdot \left(c^{\mathrm{s}}_{\mathrm{NaCl}} - c_{\mathrm{NaCl}}\right). \tag{54}$$

Here, the effective diffusion coefficient $D_{\mathrm{NaCl,eff}}$ and the effective transition coefficient $\gamma_{\mathrm{NaCl,eff}}$ are the values determined in the microscopic simulation in Sect. 4, $D_{\mathrm{NaCl,eff}} = 0.1\frac{\mathrm{m}^2}{\mathrm{s}}$ and $\gamma_{\mathrm{NaCl,eff}} = 0.000042\frac{\mathrm{m}}{\mathrm{s}}$.

A further difference in the time integration procedure comes in Steps 2 and 4 described in Sect. 3.3. Here, we fix the temperature to $T_0 = 20\,^\circ\mathrm{C}$ and, subsequently, obtain the corresponding saturation concentration $c^{\mathrm{s}}_{\mathrm{NaCl}} = 357\frac{\mathrm{kg}}{\mathrm{m}^3}$. For simplicity, the following linearized relations for density and viscosity of the solution are used (see [25])

$$\begin{aligned} \rho\left(c_{\mathrm{NaCl}}\right) &\approx \left(0.56 \cdot c_{\mathrm{NaCl}} + 1000\right)\frac{\mathrm{kg}}{\mathrm{m}^3}, \\ \eta\left(c_{\mathrm{NaCl}}\right) &\approx \left(1.96 \cdot 10^{-6} \cdot c_{\mathrm{NaCl}} + 10^{-3}\right)\frac{\mathrm{Pa}}{\mathrm{s}}. \end{aligned} \tag{55}$$

5.1 Setup

We are interested in the geometrical evolution of the double-well salt cavern. The initial geometry is given by a small cavern filled with pure water that is surrounded by sodium chloride; see Fig. 5. The dimensions of the initial cavern are approximately: width of 90 m, height of 50 m, and depth of 26 m. The sodium chloride deposit is limited to impermeable surrounding rock. The pipe on the left side acts as an inlet of freshwater with inflow velocity $|\mathbf{v}_{\mathrm{in}}| = 1\frac{\mathrm{m}}{\mathrm{s}}$, whereas the pipe on the right side acts as the outlet.

In reality, the maximum diameter of the pipes is of the order of 1 m. Hence, the resolution of the point cloud close to the inlet and the outlet has to be of the order of 0.1 m to ensure accurate results in case of the Lagrangian formulation. This would lead to an extremely small time step size compared to the desired simulation time of several years/decades due to

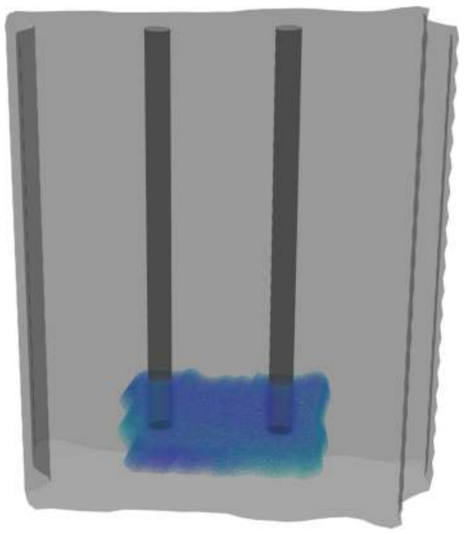

Fig. 5 Macroscopic simulation setup—initial geometry; see [25]

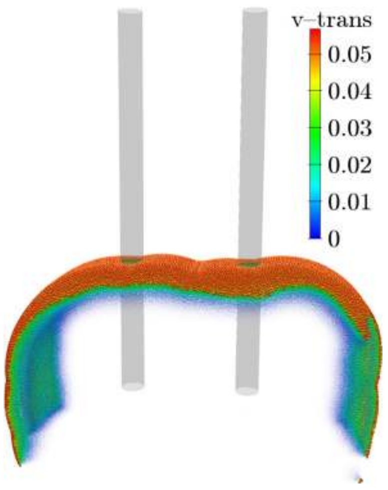

Fig. 6 Translational velocity in the macroscopic simulation, see [25] (Eulerian formulation including ALE at the moving boundary)

the CFL condition

$$\Delta t_{\text{Lag}} \leq CFL_{\text{Lag}} \cdot \frac{h_{\min}}{|\mathbf{v}|}. \tag{56}$$

Numerically, we observe that stable results are achieved for $CFL_{\text{Lag}} = 0.15$, which results in $\Delta t_{\text{Lag}} = \mathcal{O}(0.1\,\text{s})$. Performing long-term simulations over months of simulation time is not feasible with such a small time step. This results in the need for using the Eulerian formulation for the problem at hand.

In order to allow for a comparison of Lagrangian and Eulerian formulation, we consider pipes of diameter 12 m. This also decreases the required actual time being simulated. Numerically, we observe that with the significantly larger diameter used here, the evolution of the cavern only requires a few hours of physical time to be simulated, compared to the few months or years with the actual diameter. We note that despite this time reduction, this still corresponds to a timescale two orders of magnitude greater than that used in the microscopic simulations in Sect. 4.

The simulations are performed with a constant interaction radius of $h = 4\,\text{m}$. An important point to note here is that this spatial resolution considered is of the same order of magnitude as the height of the sub-cylinders in the microscopic simulations in Sect. 4.

5.2 Movement of the boundary

The movement of the boundary of the cavern can be defined by the Stefan condition

$$\rho v^{\star} = \gamma_{\text{NaCl,eff}} \left(c^{\text{s}}_{\text{NaCl}} - c_{\text{NaCl}} \right); \tag{57}$$

see [9]. This yields

$$v^{\star} = \frac{\gamma_{\text{NaCl,eff}}}{\rho} \cdot \left(c^{\text{s}}_{\text{NaCl}} - c_{\text{NaCl}} \right) \tag{58}$$

and, consequently, a movement of the boundary in normal direction $\mathbf{n}$ with velocity $\mathbf{v}_{\text{boundary}} = v^{\star} \cdot \mathbf{n}$. To speed up the computation, a time lapse procedure can be applied [25]. Due to small flow velocities inside the salt cavern, an additional speedup factor A can be introduced in the definition of $v^{\star}$ by

$$v^{\star}_{A} = \frac{A \cdot \gamma_{\text{NaCl,eff}}}{\rho} \cdot \left(c^{\text{s}}_{\text{NaCl}} - c_{\text{NaCl}} \right). \tag{59}$$

For stability reasons, we require

$$\Delta t \cdot v^{\star}_{A} \leq 0.8 \cdot h. \tag{60}$$

The maximum movement velocity of the boundary occurs in case of pure water, i.e., $c_{\text{NaCl}} = 0$, and is given by

$$v^{\star}_{A,\max} = \frac{A \cdot \gamma_{\text{NaCl,eff}} \cdot c^{\text{s}}_{\text{NaCl}}}{\rho}. \tag{61}$$

Given a maximum time step size $\Delta t_{\max}$, equation (60) leads to the constraint

$$A \leq \frac{0.8 \cdot h \cdot \rho}{\Delta t_{\max} \cdot \gamma_{\text{NaCl,eff}} \cdot c^{\text{s}}_{\text{NaCl}}}. \tag{62}$$

Due to the movement of the boundary, interior points close to this boundary have to move in the Eulerian formulation also. For this purpose, the ALE approach presented in [8] is used. Based on current and future position of an affected

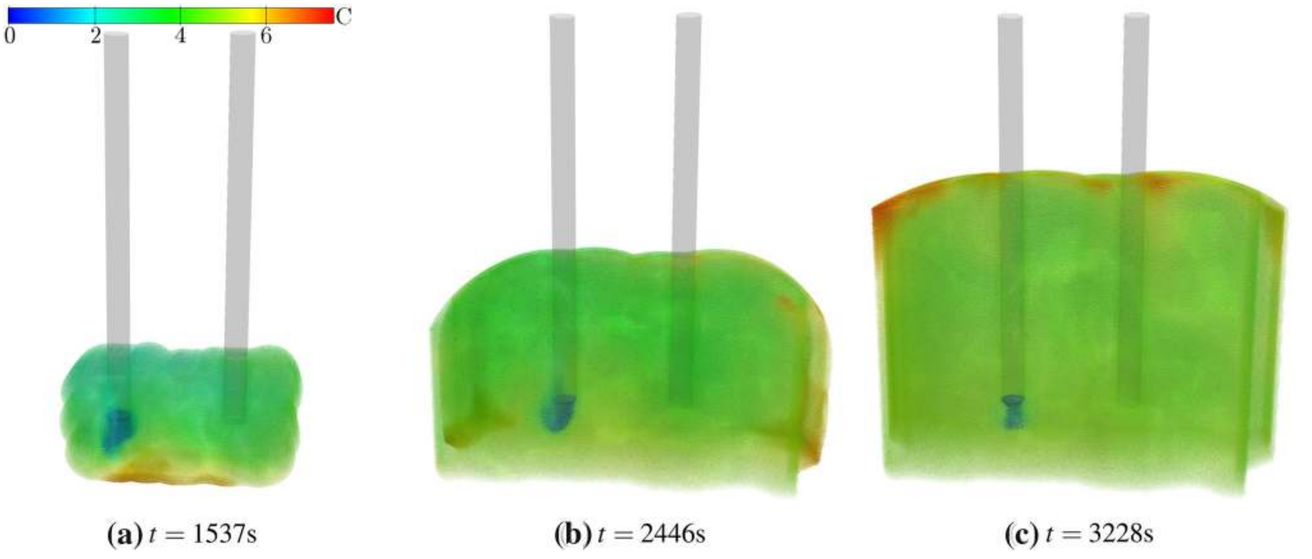

Fig. 7 Evolution of the macroscopic simulation for interaction radius $h = 4$ m—concentration, see [25] (Eulerian formulation including ALE at the moving boundary)

interior point, the translational velocity

$$\mathbf{v}_{\mathrm{trans}} = \frac{\mathbf{x}^{n+1} - \mathbf{x}^{n}}{\Delta t} \tag{63}$$

is determined; see Fig. 6. Due to the explicit movement of these points, the convection terms in the numerical model in Eulerian form in Sect. 3.3 must refer to the relative velocity $\mathbf{v} - \mathbf{v}_{\mathrm{trans}}$ instead of $\mathbf{v}$.

Furthermore, this introduces a CFL condition of the form

$$\Delta t_{\mathrm{ALE}} \leq CFL_{\mathrm{ALE}} \cdot \frac{h_{\min}}{v^{\star}}. \tag{64}$$

This depends on the boundary velocity $v^{\star} = \mathcal{O}(0.01\,\frac{\mathrm{m}}{\mathrm{s}})$ which is considerably smaller than the flow velocity (the inflow velocity is $1\,\frac{\mathrm{m}}{\mathrm{s}}$, while the maximum velocity in the domain is even bigger). $h_{\min}$ is subject to the desired resolution at the moving boundary. At the inlet and the outlet, a coarse resolution is sufficient in this case.

5.3 Numerical results

Starting from the initial domain as shown in Fig. 5, the salt cavern expands till the outer domain is filled. Physically, this outer domain can represent either a rock formation where the salt cavern ends, or prescribed limits of the region where the solution mining is to be carried out. Figure 7 illustrates the evolution of the salt cavern in the Eulerian formulation according to $\mathrm{C} = c_{\mathrm{NaCl}}$. An animation of this process is given in Online Resource. This expansion is quantified by plotting the volume as a function of time in Fig. 8. We note that once the entire outer domain is filled, the volume of the domain

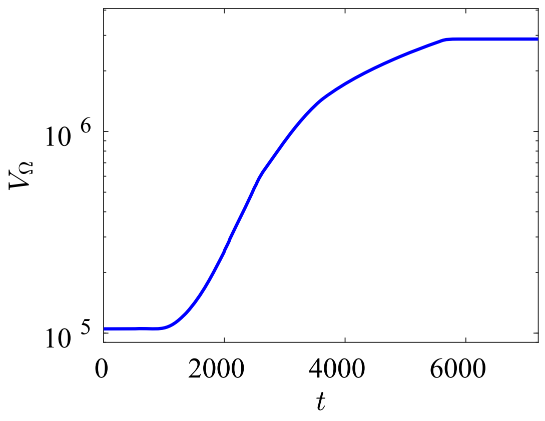

Fig. 8 Expansion in the volume of the computational domain of the macroscopic simulation as the simulation progresses

is about 27.4 times that of the initial domain. This shows the need of a meshfree method for the present application. If a mesh-based method were to be used for this simulation, the entire domain would need to be meshed initially, and an expensive and less accurate tracking of the expansion would need to be carried out.

To compare the results of the Eulerian and Lagrangian formulations, we consider simulations on the same initial point cloud with $h = 4$ m which corresponds to $NP_0 = 90{,}744$ points at the initial state. The simulations are run until $t = 7200$ s $= 2$ h. At the end time, both simulations have about 1.5 million points in the final expanded domain. To quantify the results, and to enable a comparison between the two formulations, we consider the time integration of the

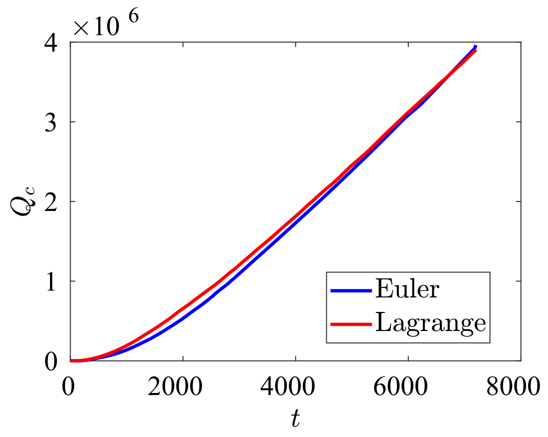

Fig. 9 Comparison of the concentration flux across the outflow pipe between the Eulerian (with moving boundaries) and Lagrangian simulations

concentration weighted flux at the outflow boundary

$$Q_c(t) = \int_0^t \left(\int_{\partial\Omega_{\text{out}}} c_{\text{NaCl}}\, \mathbf{v} \cdot \mathbf{n}\, \mathrm{d}A \right) \mathrm{d}\tau, \tag{65}$$

where $\partial\Omega_{\text{out}}$ is the outflow boundary located at the top of the extraction well. Physically, this represents a measure of the concentration of salt being extracted. The time evolution of Q_c is shown in Fig. 9. It illustrates that both formulations produce very similar results.

To emphasize the need of the ALE formulation for such a simulation, we compare the time steps required in both the ALE and Lagrangian formulations for stability. Considering the simulation time of 7200 s, we observe that the Lagrangian formulation required at least 22915 time steps to obtain stable results, which corresponds to an average time step size of $\Delta t \approx 0.31$ s. On the other hand, similar results, as shown in Fig. 9, can be obtained in the Eulerian formulation (with ALE near the boundaries) with only 936 time steps corresponding to an average time step size of $\Delta t \approx 7.69$ s, which is approximately 25 times that needed in the Lagrangian case.

6 Conclusions

In this contribution, we presented the capabilities of the generalized finite difference method (GFDM) implemented in the simulation software MESHFREE regarding solution mining processes on a macroscopic and a microscopic scale. Both Lagrangian and Eulerian approaches were considered.

On the macroscopic scale, we considered the expansion of the salt cavern as a result of erosion occurring as the salt dissolves in the water. In reality, this procedure occurs over the time span of several months or years. A simplified geometry was considered here, which enabled a comparison between the Eulerian and Lagrangian formulations. In this simplified macroscopic setup, the expansion of the salt cavern occurred over the timescale of several hours. Since the dissolution of salt in water occurs on a much smaller time level we also considered a microscopic setup over a duration of a few minutes. This was used to determine effective parameters governing the dissolution process. Using the example of sodium chloride as the species of interest, effective diffusion and transition coefficients were determined in the microscopic simulations. These values were then used in the macroscopic simulations to determine the evolution of the concentration inside the salt cavern and to specify the solution rate of the salt species at the boundary, i.e., to model the geometrical evolution of the salt cavern.

A comparison of the numerical results of the Lagrangian and Eulerian formulations (extended by an ALE approach) in the macroscopic case illustrates the advantages of the latter one due the possibility of using much larger time step sizes. Aiming at a simulation time of several years, the forecast computation time for a simulation of a double-well solution mining process based on the Lagrangian formulation would be of the order of years. In contrast to that, the flexibility of the Eulerian formulation regarding the resolution of the point cloud (local refinement only at the moving boundary) enables meshfree simulations in reasonable time—especially in terms of real applications.

Funding Open Access funding provided by Projekt DEAL.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Alexander R (1977) Diagonally implicit runge-kutta methods for stiff ode's. SIAM J Numer Anal 14(6):1006–1021
2. Chen J, Lu D, Liu W, Fan J, Jiang D, Yi L, Kang Y (2020) Stability study and optimization design of small-spacing two-well (sstw) salt caverns for natural gas storages. J Energy Storage 27:101131
3. Chorin AJ (1968) Numerical solution of the Navier–Stokes equations. Math Comput 22(104):745–762

4. Drumm C, Tiwari S, Kuhnert J, Bart H-J (2008) Finite pointset method for simulation of the liquid–liquid flow field in an extractor. Comput Chem Eng 32(12):2946–2957
5. Fan C-M, Chu C-N, Šarler B, Li T-H (2018) Numerical solutions of waves–current interactions by generalized finite difference method. Engineering Analysis with Boundary Elements
6. Flury M, Gimmi T (2002) Solute diffusion. In: Dane J, Topp G (eds) Methods of soil analysis. Part 4. Physical methods. SSSA Book Ser. 5, Madison, WI, pp 1323–1351
7. Gavete L, Urena F, Benito J, García A, Urena M, Salete E (2017) Solving second order non-linear elliptic partial differential equations using generalized finite difference method. J Comput Appl Math 318:378–387 (Computational and Mathematical Methods in Science and Engineering CMMSE-2015)
8. Hirt CW, Amsden AA, Cook J (1974) An arbitrary Lagrangian–Eulerian computing method for all flow speeds. J Comput Phys 14(3):227–253
9. Javierre-Perez E (2003) Literature study: numerical methods for solving Stefan problems. Delft University of Technology, Delft
10. Jefferies A, Kuhnert J, Aschenbrenner L, Giffhorn U (2015) Finite pointset method for the simulation of a vehicle travelling through a body of water. In: Griebel M, Schweitzer AM (eds) Meshfree methods for partial differential equations VII. Springer, Cham, pp 205–221
11. Karsten O (1954) Lösungsgeschwindigkeit von natriumchlorid, kaliumchlorid und kieserit in wasser und in wässerigen lösungen. Zeitschrift für anorganische und allgemeine Chemie 276(5–6):247–266
12. Katz A, Jameson A (2010) Meshless scheme based on alignment constraints. AIAA J 48(11):2501–2511
13. Kuhnert J (2014) Meshfree numerical scheme for time dependent problems in fluid and continuum mechanics. In: Sundar S (ed) Advances in PDE modeling and computation. Anne Books, New Delhi, pp 119–136
14. Kuhnert J, Michel I, Mack R (2019) Fluid structure interaction (fsi) in the meshfree finite pointset method (fpm): Theory and applications. In: Griebel M, Schweitzer AM (eds) Meshfree methods for partial differential equations IX, IWMMPDE2017. Springer, Berlin, pp 73–92
15. Laliberté M (2007) Model for calculating the viscosity of aqueous solutions. J Chem Eng Data 52(2):321–335
16. Laliberté M (2009) A model for calculating the heat capacity of aqueous solutions, with updated density and viscosity data. J Chem Eng Data 54(6):1725–1760
17. Laliberté M, Cooper WE (2004) Model for calculating the density of aqueous electrolyte solutions. J Chem Eng Data 49(5):1141–1151
18. Launder B, Spalding D (1974) The numerical computation of turbulent flows. Comput Methods Appl Mech Eng 3(2):269–289
19. Löhner R, Onate E (1998) An advancing front point generation technique. Commun Numer Methods Eng 14(12):1097–1108
20. Luo M, Koh CG, Bai W, Gao M (2016) A particle method for two-phase flows with compressible air pocket. Int J Numer Methods Eng 108:695–721
21. Michel I, Bathaeian SMI, Kuhnert J, Kolymbas D, Chen C-H, Polymerou I, Vrettos C, Becker A (2017) Meshfree generalized finite difference methods in soil mechanics—part ii: numerical results. Int J Geomath 8(2):191–217
22. Nick F, Plum H-J, Kuhnert J (2019) Parallel detection of subsystems in linear systems arising in the meshfree finite pointset method. In: Griebel M, Schweitzer MA (eds) Meshfree methods for partial differential equations IX. Springer, Cham, pp 93–115
23. Saucedo-Zendejo FR, Reséndiz-Flores EO, Kuhnert J (2019) Three-dimensional flow prediction in mould filling processes using a gfdm. Comput Particle Mech 6(3):411–425
24. Seibold B (2006) M-matrices in meshless finite difference methods. PhD thesis, Kaiserslautern University
25. Seifarth T (2017) Numerische Algortihmen für gitterfreie Methoden zur Lösung von Transportproblemen. PhD thesis, University of Kassel, Kassel
26. Suchde P (2018) Conservation and accuracy in meshfree generalized finite difference methods. PhD thesis, University of Kaiserslautern, Kaiserslautern, Germany
27. Suchde P, Kuhnert J (2018) Point cloud movement for fully Lagrangian meshfree methods. J Comput Appl Math 340:89–100
28. Suchde P, Kuhnert J (2019) A fully Lagrangian meshfree framework for PDEs on evolving surfaces. J Comput Phys 395:38–59
29. Suchde P, Kuhnert J (2019) A meshfree generalized finite difference method for surface PDEs. Comput Math Appl 78(8):2789–2805
30. Suchde P, Kuhnert J, Schröder S, Klar A (2017) A flux conserving meshfree method for conservation laws. Int J Numer Methods Eng 112(3):238–256
31. Uhlmann E, Barth E, Seifarth T, Höchel M, Kuhnert J, Eisenträger A (2020) Simulation of metal cutting with cutting fluid using the finite-pointset-method. In: 9th CIRP conference on high performance cutting. Submitted to Procedia CIRP
32. Uhlmann E, Gerstenberger R, Kuhnert J (2013) Cutting simulation with the meshfree finite pointset method. Procedia CIRP 8:391–396
33. Zhang G, Wang Z, Zhang K, Li Y, Wu Y, Chen Y, Zhang H (2018) Collapse mechanism of the overlying strata above a salt cavern by solution mining with double-well convection. Environ Earth Sci 77(16):588