Comput Geosci (2014) 18:103–115
DOI 10.1007/s10596-013-9377-8

ORIGINAL PAPER

A robust method to tackle pressure boundary conditions in porous media flow: application to biogrout

W. K. van Wijngaarden · F. J. Vermolen · G. A. M. van Meurs · C. Vuik

Received: 2 May 2013 / Accepted: 27 September 2013 / Published online: 31 October 2013

Abstract Biogrout is a soil improvement method in which microorganisms are used to produce the solid calcium carbonate, which strengthens the soil by connecting soil particles. Microorganisms in the soil are supplied with some nutrients, which they convert into calcium carbonate. These nutrients and the side product of the reaction are dissolved in water. Because of these chemicals, the fluid is denser than water. Moreover, the density changes as a result of the varying composition. This changing density has a significant impact on the flow. Since the composition and hence, the density is not known beforehand, a careful choice of the (pressure) boundary conditions, especially on the outflow boundary, is needed. In this article, several methods to approximate the pressure on the outflow boundary are compared. The method that we propose also works for an unstructured mesh, which gives a large freedom in the mesh generation.

Keywords Biogrout · Pressure boundary condition · Transport in porous media · MICP

1 Introduction

Biogrout is a soil improvement method, in which microorganisms are used to produce calcium carbonate ($CaCO_3$). This solid strengthens the soil by connecting soil particles. The microorganisms are already present in the soil [10] or injected into it [17]. The microorganisms are supplied with urea ($CO(NH_2)_2$) and calcium chloride ($CaCl_2$). The microbial enzyme *urease* provides the hydrolysis of urea. The hydrolysis reaction equation is given in [17]:

$$CO(NH_2)_2 + 2H_2O \xrightarrow{\text{bacteria}} 2NH_4^+ + CO_3^{2-}. \quad (1)$$

The reaction products of this reaction are ammonium $\left(NH_4^+\right)$ and carbonate $\left(CO_3^{2-}\right)$. In the presence of calcium ions $\left(Ca^{2+}\right)$, the carbonate precipitates as calcium carbonate ($CaCO_3$):

$$Ca^{2+}(aq) + CO_3^{2-}(aq) \rightarrow CaCO_3(s). \quad (2)$$

Combining reactions (1) and (2) gives the overall reaction equation:

$$CO(NH_2)_2(aq) + Ca^{2+}(aq) + 2H_2O(l) \rightarrow 2NH_4^+(aq) + CaCO_3(s). \quad (3)$$

The urea and calcium chloride are dissolved in water, as well as the side product ammonium chloride (NH_4Cl). Because of these chemicals, the fluid is denser than pure water. In [12], the following relation between the density of the solution ρ_l(at 20 °C) and these concentrations is used:

$$\rho_l = 1,000 + 15.4996C^{\text{urea}} + 86.7338C^{Ca^{2+}} + 15.8991C^{NH_4^+}. \quad (4)$$

In this relation, C^{urea} is the concentration of urea, $C^{Ca^{2+}}$ is the concentration of calcium chloride and $C^{NH_4^+}$ is the concentration of ammonium chloride, each with M(=kmol/m^3) as a unit. This relation has been found, using [16]. From the tables of the individual species, a linear relation between the various concentrations and the density has

W. K. van Wijngaarden (✉) · F. J. Vermolen · C. Vuik
Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
e-mail: Miranda.vanWijngaarden@Deltares.nl

W. K. van Wijngaarden · G. A. M. van Meurs
Deltares, unit Geo Engineering, Stieltjesweg 2, 2628 CK Delft, The Netherlands

been found for a single species dissolved in water. However, in the Biogrout case, several species are present in the fluid. We assume that the contributions of the various species can be added in the case that more than one species are dissolved. Experimental validation showed that this relation gives a good description of reality [15]. In this article, Eq. (4) is used in the simulations.

Due to biochemical reaction (3), the solution has a varying composition and hence, also the density changes. This changing density has an effect on the water pressure. Flow boundary conditions are often given in terms of pressure, especially on the outflow boundaries, where usually a fixed head is applied to prevent desaturation of the soil. Unsaturated soil can occur when a fixed flow rate is prescribed. A fixed head gives a hydrostatic pressure boundary condition. At the injection, the composition of the solution is usually known, but at the extraction, the density is not known beforehand. Therefore, a careful choice of the boundary condition is necessary.

In this article, several methods to calculate the pressure on the extraction boundary are presented and compared. The last method, which is most robust, works for any finite-element mesh, even including unstructured grids. This gives a large freedom in the mesh generation.

2 Mathematical model

In this section, we give the model equations with the initial and boundary conditions for the model.

2.1 Model equations

In this subsection, the model equations for the Biogrout process are presented. For a derivation and a more thorough discussion, we refer to [12].

To model the concentration of the aqueous species (urea, calcium chloride and ammonium chloride), we use an advection–dispersion-reaction equation:

$$\frac{\partial(\theta C^i)}{\partial t} = \nabla \cdot (\theta \mathbf{D}^{\mathbf{i}} \cdot \nabla C^i) - \nabla \cdot (\mathbf{q} C^i) + n_i \theta r_{\text{hp}}. \tag{5}$$

In this equation, θ is the porosity, $\mathbf{D}$ is the dispersion tensor, $\mathbf{q}$ is the Darcy velocity, n_i is a constant that deals with the stoichiometry in the biochemical reaction Eq. (3) and r_{hp} is the reaction rate of the production of calcium carbonate. From the stoichiometry of reaction (3), the values of n_i for the various aqueous species are given by: $n_{\text{urea}} = -1$, $n_{Ca^{2+}} = -1$ and $n_{NH_4^+} = 2$.

The left-hand side of Eq. (5) models the accumulation. In the right-hand side, we have terms for dispersion/diffusion, for advection and for the biochemical reaction (3).

In this paper, we use the following relation for the reaction rate r_{hp} in Eq. (3):

$$r_{\text{hp}} = v_{\max} \frac{C^{\text{urea}}}{K_{m,\text{urea}} + C^{\text{urea}}}. \tag{6}$$

Here, $v_{\max}$ is the bacterial conversion rate constant for a (given) specific bacterial density and $K_{m,\text{urea}} \geq 0$ is the saturation constant. In this paper, we use a bacterial conversion rate that is constant over the whole domain, thereby neglecting any variations of temperature, bacterial density or pH over the time and space. The concentration of bacteria can also be simulated, using the model proposed in [13]. This model describes the placement of bacteria: bacteria are injected in the soil. After the injection of the bacteria, a pulse with fixation fluid is injected. This fixation fluid has less retardation than the bacterial pulse and will overtake the bacterial pulse, fixating the bacteria in the subsoil. Since the focus of this paper is on the pressure boundary condition, we use this simplified reaction rate, which implies a homogeneous distribution of bacteria.

It is assumed that calcium carbonate is not transported. Hence, there is only an accumulation term from the reaction term in the differential equation for the calcium carbonate concentration:

$$\frac{\partial C^{\text{CaCO}_3}}{\partial t} = m_{\text{CaCO}_3} \theta r_{\text{hp}}. \tag{7}$$

In this equation, C^{CaCO_3} is the concentration of calcium carbonate (in kilograms per cubic meter) and m_{CaCO_3} is the molar mass of calcium carbonate, which is used to convert moles into mass.

The solid calcium carbonate that is formed in the pores causes a decrease in porosity. The difference $(C^{\text{CaCO}_3}(t, \mathbf{x}) - C^{\text{CaCO}_3}(0, \mathbf{x}))$ gives the amount of calcium carbonate that has been formed per unit of volume in time period t. Division by the density of calcium carbonate ρ_{CaCO_3} gives the decrease in pore volume per unit of volume. That leads to the following relation between the calcium carbonate concentration and the porosity:

$$\theta(t, \mathbf{x}) = \theta(0, \mathbf{x}) - \frac{C^{\text{CaCO}_3}(t, \mathbf{x}) - C^{\text{CaCO}_3}(0, \mathbf{x})}{\rho_{\text{CaCO}_3}}. \tag{8}$$

For the flow, we use the continuity equation, that was derived in [14]. This differential equations is an adaptation of the differential equation derived in [12], since the differential equation in [12] does not conserve mass. It is based on the assumption that reaction (3) does not influence the total amount of liquid volume, which turned out to be untrue. The adapted differential equation is given by:

$$\nabla \cdot \mathbf{q} = K \theta r_{\text{hp}}. \tag{9}$$

The constant K represents the amount of volume that is produced per kilomole formed calcium carbonate by reaction

(3) and it has been defined as

$$K := \left(\frac{m_{\mathrm{CaCO_3}}}{\rho_{\mathrm{CaCO_3}}} - (1 - V_s) \right). \tag{10}$$

This constant deals with two phenomena. When reaction (3) takes place, various species disappear from the solution and therefore the liquid volume decreases. On the other hand, due to the same reaction, the solid calcium carbonate is formed, which causes a decrease in pore space. The decrease in pore space per kilomole formed calcium carbonate is $\frac{m_{\mathrm{CaCO_3}}}{\rho_{\mathrm{CaCO_3}}}$, and the decrease in liquid volume per kilomole formed calcium carbonate is $1 - V_s$. These two phenomena only partly cancel each other.

As a relation between the flow and the pressure p, Darcy's law is used [18],

$$q_x = -\frac{k_x}{\mu} \frac{\partial p}{\partial x}, \tag{11}$$

$$q_y = -\frac{k_y}{\mu} \frac{\partial p}{\partial y}, \tag{12}$$

$$q_z = -\frac{k_z}{\mu} \left(\frac{\partial p}{\partial z} + \rho_l g \right). \tag{13}$$

In Darcy's law, k_i is the intrinsic permeability in the various coordinate directions, $i \in \{x, y, z\}$, μ is the viscosity of the fluid, ρ_l is the density of the fluid and g is the gravitational constant.

The Kozeny–Carman equation is used to determine the intrinsic permeability. This equation is an empirical relation between the intrinsic permeability and the porosity, that is commonly used in ground water flow modelling (see [2]):

$$k = k_x = k_y = k_z = \frac{(d_{\mathrm{m}})^2}{180} \frac{\theta^3}{(1-\theta)^2}. \tag{14}$$

In this relation, d_{m} is the mean particle size of the soil.

Substituting Eqs. (11), (12) and (13) into Eq. (9), using relation (14), gives a partial differential equation for the pressure. This partial differential equation is solved to compute the flow pattern if the boundary conditions are given in terms of pressure, or if density differences influence the flow:

$$\nabla \cdot \mathbf{q} = \nabla \cdot \left(-\frac{k}{\mu} (\nabla p + \rho_l g \mathbf{e_z}) \right) = K \theta r_{\mathrm{hp}}. \tag{15}$$

Here, $\mathbf{e_z}$ is the unit vector in vertical direction, taken positive upwards.

2.2 Experimental set-up and initial and boundary conditions

As a model experiment, we take the 100-m^3 experiment as reported in [9]. The configuration is shown in Fig. 1. A concrete box (8 m × 5.6 m × 2.5 m) is filled with sand and fully saturated. Three injection wells (left) and three extraction wells (right) are used to flush the liquids through the sand body. The distance between injection and extraction is 5 m. The other boundaries are closed.

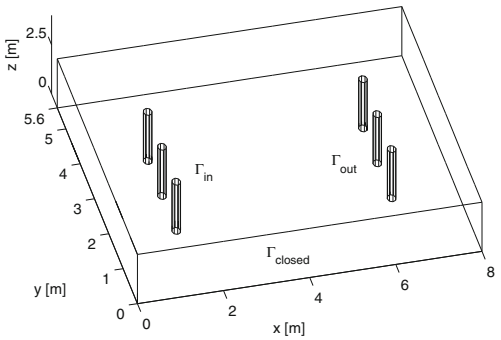

Fig. 1 Set-up of the experiment. Injection lances are denoted by Γ_{in} and the extraction lances by Γ_{out}. The other boundaries (Γ_{closed}) are closed

Initially, the concentration of calcium carbonate, urea, calcium and ammonium are equal to zero and the porosity equals some constant θ_0.

Table 1 displays the boundary conditions for the various concentrations and the flow. On the inflow boundary, we prescribe the inflow velocity and the flux. On the closed boundary, there is no flow perpendicular to the boundary and hence the flux over the boundary equals zero. On the outflow boundary, we prescribe the pressure and for the concentrations an advective flux is assumed. Due to the gravitational force, it is required that each part of the outflow boundaries is part of a *vertical* plane. In the extraction wells, at a certain depth (equal to the top of the sand body), a pump has been installed, which keeps the water level in the well at a fixed position. Note that the container is covered with a watertight foil, which is loaded with another layer of sand. This makes it possible to create an overpressure around the injection wells, which results into flow from injection to extraction. Since the resistance to flow in the extraction wells is very low, we assume a hydrostatic pressure on its boundaries, see for example [2]. That leads to the following differential equation on the surface of the extraction boundaries:

$$\frac{\partial p}{\partial z} = -\rho_l g, \tag{16}$$

with at the height of the pump (at $z = 2.5$ m)

$$p(2.5) = p_{\mathrm{atm}}, \tag{17}$$

in which p_{atm} is the atmospheric pressure. Integration of Eq. (16), combined with boundary condition (17), gives the following equation for the pressure for all points z on

the surface of the extraction boundaries, for which holds $0 \leq z \leq 2.5$ m :

$$p(x, y, z) = p_{\text{atm}} + \int_z^{2.5} \rho_l(x, y, \bar{z}) g d\bar{z}, \tag{18}$$

see also Table 1.

To compute the pressure given by Eq. (18), all the nodes of the mesh should be positioned on vertical lines. On these lines, the x- and y-coordinates are constant, while the z-coordinate is variable. Over these lines, the integral in Eq. (18) is computed, which gives the pressure at all nodes on these lines.

The requirement that all the nodes are on certain vertical lines pats a severe requirement on the mesh generation. Another possibility, which also allows unstructured meshes, is solving differential Eq. (16) with a finite element method. In that case, the partial differential equation is solved on a (2D) manifold as a boundary condition for a 3D domain. Some examples from literature in which differential equations are solved on manifolds are in [3, 5–7].

3 Numerical methods

In this section, it is explained which numerical methods are used to solve the partial differential equations in the 3D domain, Eqs. (5) to (15), as well as the equations for the pressure boundary condition, Eq. (16) and alternatively Eq. (18).

Equations (5), (11), (12), (13) and (15) are solved using the Standard Galerkin Finite Element Method. These equations are multiplied by a test function η and integrated over the domain Ω to derive the weak formulation. For the time integration of Eq. (5), an implicit scheme is used. Since the reaction rate r_{hp} is non-linear in the urea concentration, Newton's method is used to calculate the urea concentration. Since the differential equation for the calcium carbonate concentration (7) is an ordinary differential equation in each node, the finite element method is not used to solve Eq. (7). An implicit time integration method is used to solve this equation. For more details about the numerical methods to solve these equations, we refer to [11] and [12], where this has been reported in more detail.

In this article, the pressure on the outflow boundary is calculated in three different ways. The first two methods involve a calculation, based on Eq. (18). In the first method (method 1), the integral in Eq. (18) is approximated using the Lower Riemann Sum (see [1]). This is a first-order method. Let n be the number of nodes on one of the vertical lines. The nodes on this line are ordered in the following way: $0 = z_0 < z_1 < .... < z_{n-1} < z_n = 2.5$. Then, Eq. (18) is approximated by:

$$p(x, y, z_j) \approx p_{\text{atm}} + \sum_{i=j}^{n-1} g \rho_l(x, y, z_{i+1})(z_{i+1} - z_i). \tag{19}$$

This sum is calculated for each node on each vertical line.

In the second method (method 2), the second-order trapezoid rule is used to approximate (18), see [1]. Using the same notation as for the Lower Riemann Sum, we get the following approximation:

$$p(x, y, z_j) \approx p_{\text{atm}} + \sum_{i=j}^{n-1} g \frac{\rho_l(x, y, z_i) + \rho_l(x, y, z_{i+1})}{2}(z_{i+1} - z_i). \tag{20}$$

This sum is calculated for each node on each vertical line.

As a third method (method 3), the pressure on the boundary is calculated, by solving Eq. (16) subject to boundary condition (17). For this calculation, the Standard Galerkin Finite Element Method is used. To derive the weak formulation, Eq. (16) is multiplied by a test function η and integrated over the surface of the outflow boundary Γ_{out}:

$$\int_{\Gamma_{\text{out}}} \frac{\partial p}{\partial z} \eta dS = -\int_{\Gamma_{\text{out}}} \rho_l g \eta dS. \tag{21}$$

The pressure p is approximated by

$$p \approx \sum_{j=1}^{N} p_j \varphi_j, \tag{22}$$

in which φ_j is a linear basis function and where p_j denote the pressure approximations on nodes on the boundary. The z-derivatives of the basis functions are determined after a mapping to the $(x - z)$-plane. The integral over the outflow boundary is approximated by the sum of the integrals over the elements. The Newton–Cotes quadrature rules have been used to develop the element matrices and vectors. Triangular boundary elements have been used. Differential Eq. (16) with boundary condition (17) can be considered as an initial

Table 1 Boundary conditions for the various concentrations and the flow

	Γ_{in}	Γ_{out}	Γ_{closed}
C^{urea}	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = q_{\text{in}} c_{\text{in}}$	$(D\theta\nabla C) \cdot \mathbf{n} = 0$	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = 0$
$C^{\text{Ca}^{2+}}$	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = q_{\text{in}} c_{\text{in}}$	$(D\theta\nabla C) \cdot \mathbf{n} = 0$	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = 0$
$C^{\text{NH}_4^+}$	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = 0$	$(D\theta\nabla C) \cdot \mathbf{n} = 0$	$(D\theta\nabla C - \mathbf{q}C) \cdot \mathbf{n} = 0$
q	$-\mathbf{q} \cdot \mathbf{n} = q_{\text{in}}$	$p(x, y, z) = p_{\text{atm}} + \int_z^{2.5} \rho_l(x, y, \bar{z}) g d\bar{z}$	$\mathbf{q} \cdot \mathbf{n} = 0$;

Table 2 Comparison of the three pressure calculation methods with the analytical solution on a *structured* mesh for a *linear* relation between the depth and the density

Number of	Method 1			Method 2			Method 3		
elements	Error	α	Computing time (s)	Error	α	Computing time (s)	Error	α	Computing time (s)
40	613	1.00	0.0005	0		0.0005	302	1.93	0.0006
160	307	1.00	0.0008	0		0.0008	79	1.89	0.0013
640	153	1.00	0.0021	0		0.0018	21	1.89	0.0036
2,560	77	1.00	0.0035	0		0.0037	5.8	1.95	0.0193
10,240	38	1.00	0.0113	0		0.0115	1.5	1.97	0.1335
40,960	19		0.0471	0		0.0478	0.38		1.7352

The α-factor is determined from the (unrounded) errors of two subsequent meshes, using Eq. (40). The computing time (average of 10 simulations) for the various methods is also given

value problem. Using the Standard Galerkin Finite Element Method and choosing $\eta = \varphi_i$ for $i \in \{1, \cdots, N\}$ as a test function will lead to stability problems. Inspired by the Streamline-Upwind/Petrov-Galerkin (SUPG) method, see [8], we choose as a test function: $\eta = \varphi_i - \frac{h\xi}{2}\frac{\partial \varphi_i}{\partial z}$, for $i \in \{1, \cdots, N\}$. In this equation, h is some representative distance in the element and ξ is some constant. In case of application on a surface, we choose for h:

$$h := \sqrt{\frac{2A}{n_{\text{el}}}}, \tag{23}$$

in which A is the total surface of the domain and n_{el} is the number of elements on the surface. As a value for ξ, we take $\xi = 10^{-5}$. As differential Eq. (16) is similar to a stationary advection equation, the SUPG method will introduce some artificial diffusion, which stabilizes the system. If the SUPG method is not used, then the discretization matrix might have an eigenvalue that is equal to zero.

In this paragraph, we do some analysis to investigate the influence of the value of ξ on the differences in eigenvalues of the system with SUPG stabilization and the system without SUPG stabilization, applied on a surface. Let H be the matrix that is used to solve the system without SUPG stabilization and $\hat{H}(h\xi)$ the matrix for the system with SUPG. Both matrices are $n \times n$ matrices. The entries of matrix $\hat{H}(h\xi)$ are given by $\hat{H}_{ij}(h\xi) = H_{ij} + h\xi \Delta H_{ij}$ and hence, $\hat{H}(\varepsilon) = H + \varepsilon \Delta H$, with $\varepsilon = h\xi$. The $\varepsilon \Delta H$ part in the matrix accounts for the SUPG stabilization. Further, let $(\lambda, \mathbf{v})$ be an eigenpair of H ($H\mathbf{v} = \lambda \mathbf{v}$) and let $(\lambda(\varepsilon), \mathbf{v}(\varepsilon))$ be an eigenpair of $\hat{H}(\varepsilon)$ ($\hat{H}(\varepsilon)\mathbf{v}(\varepsilon) = \lambda(\varepsilon)\mathbf{v}(\varepsilon)$) and suppose that $\mathbf{w}^{\mathbf{H}}$ is a left eigenvector of H ($\mathbf{w}^{\mathbf{H}} H = \lambda \mathbf{w}^{\mathbf{H}}$). Then

$$(H + \varepsilon \Delta H)\mathbf{v}(\varepsilon) = \lambda(\varepsilon)\mathbf{v}(\varepsilon). \tag{24}$$

To estimate $\lambda(\varepsilon)$, Eq. (24) is differentiated with respect to ε, which gives

$$\Delta H \mathbf{v}(\varepsilon) + (H + \varepsilon \Delta H)\frac{d\mathbf{v}(\varepsilon)}{d\varepsilon} = \frac{d\lambda(\varepsilon)}{d\varepsilon}\mathbf{v}(\varepsilon) + \lambda(\varepsilon)\frac{d\mathbf{v}(\varepsilon)}{d\varepsilon}. \tag{25}$$

Table 3 Comparison of the three pressure calculation methods with the analytical solution on a *structured* mesh for a *non-linear* relation between the depth and the density

Number of	Method 1		Method 2		Method 3	
elements	Error	α	Error	α	Error	α
40	4,880	0.879	485	1.963	1798	1.63
160	2,654	0.942	125	1.982	580	1.78
640	1,382	0.971	32	1.991	169	1.87
2,560	705	0.986	7.9	1.996	46	1.94
10,240	356	0.993	2.0	1.998	12	1.96
40,960	179		0.50		3.1	

The α-factor is determined from the (unrounded) errors of two subsequent meshes, using Eq. (40)

Table 4 Comparison of the three pressure calculation methods with the analytical solution on an *unstructured* mesh for a *linear* relation between the depth and the density

	Number of	Method 1		Method 2		Method 3	
	elements	Error	α	Error	α	Error	α
Mesh a	38	1,628	−0.05	872	−0.53	296	1.94
Mesh *b*	152	1,690	−0.08	1,259	−0.30	77	1.39
Mesh *c*	608	1,780	−0.05	1,547	−0.15	29	2.66
Mesh *d*	2,432	1,843	−0.03	1,721	−0.08	4.6	1.81
Mesh *e*	9,728	1,879	−0.01	1,816	−0.04	1.3	1.73
Mesh *f*	38,912	1,898		1,866		0.40	
Mesh *f*2	40,860	1,929		1,850		0.39	

The α-factor is determined from the (unrounded) errors of two subsequent meshes, using Eq. (40)

Set $\varepsilon = 0$, then we get ($\mathbf{v}(0) = \mathbf{v}$, $\lambda(0) = \lambda$)

$$\Delta H\mathbf{v} + H\frac{d\mathbf{v}(0)}{d\varepsilon} = \frac{d\lambda(0)}{d\varepsilon}\mathbf{v} + \lambda\frac{d\mathbf{v}(0)}{d\varepsilon}. \tag{26}$$

Left-multiplication by $\mathbf{w^H}$ gives

$$\mathbf{w^H}\Delta H\mathbf{v} + \mathbf{w^H}H\frac{d\mathbf{v}(0)}{d\varepsilon} = \frac{d\lambda(0)}{d\varepsilon}\mathbf{w^H}\mathbf{v} + \lambda\mathbf{w^H}\frac{d\mathbf{v}(0)}{d\varepsilon}. \tag{27}$$

Hence, since $\mathbf{w^H}H = \lambda\mathbf{w^H}$:

$$\mathbf{w^H}\Delta H\mathbf{v} = \frac{d\lambda(0)}{d\varepsilon}\mathbf{w^H}\mathbf{v}. \tag{28}$$

Since $\mathbf{w^H}\mathbf{v} \neq 0$, we get

$$\left|\frac{d\lambda(0)}{d\varepsilon}\right| = \frac{\left|\mathbf{w^H}\Delta H\mathbf{v}\right|}{\left|\mathbf{w^H}\mathbf{v}\right|} \leq \frac{\left\|\mathbf{w^H}\right\|_2 \|\Delta H\|_2 \|\mathbf{v}\|_2}{\left|\mathbf{w^H}\mathbf{v}\right|} = \frac{\|\Delta H\|_2}{\left|\mathbf{w^H}\mathbf{v}\right|}. \tag{29}$$

The last step is motivated by choosing $\left\|\mathbf{w^H}\right\|_2 = \|\mathbf{v}\|_2 = 1$. Furthermore,

$$\begin{aligned}\|\Delta H\|_2 \leq \|\Delta H\|_F &:= \left[\sum_i \sum_j |\Delta H_{ij}|^2\right]^{1/2} \\ &\leq \left[\max_{i,j} cn|\Delta H_{ij}|^2\right]^{1/2} = \frac{C}{h}\max_{i,j}|\Delta H_{ij}|,\end{aligned} \tag{30}$$

where cn is motivated by the sparsity of the matrix ΔH. Moreover, it has been used that $n = L^2/h^2$ with L the length of the domain and C has been defined as $C := L\sqrt{c}$. Hence, we have

$$\left|\frac{d\lambda(0)}{d\varepsilon}\right| \leq \frac{C\max_{i,j}|\Delta H_{ij}|}{h|\mathbf{w^H}\mathbf{v}|}. \tag{31}$$

This can be rewritten as

$$\lim_{\varepsilon\to 0}\frac{|\lambda(\varepsilon) - \lambda(0)|}{|\varepsilon|} \leq \frac{C\max_{i,j}|\Delta H_{ij}|}{h|\mathbf{w^H}\mathbf{v}|}. \tag{32}$$

Table 5 Comparison of the three pressure calculation methods with the analytical solution on an *unstructured* mesh for a *non-linear* relation between the depth and the density

	Number of	Method 1		Method 2		Method 3	
	elements	Error	α	Error	α	Error	α
Mesh *a*	38	9,118	−0.05	7,147	−0.28	2,188	1.23
Mesh *b*	152	9,413	−0.08	8,704	−0.16	934	1.86
Mesh *c*	608	9,974	−0.06	9,737	−0.08	258	2.01
Mesh *d*	2,432	10,394	−0.03	10,313	−0.04	64	1.91
Mesh *e*	9,728	10,645	−0.02	11,614	−0.02	17	2.09
Mesh *f*	38,912	10,781		10,768		4.4	
Mesh *f*2	40,860	11,037		10,884		2.3	

The α-factor is determined from the (unrounded) errors of two subsequent meshes, using Eq. (40)

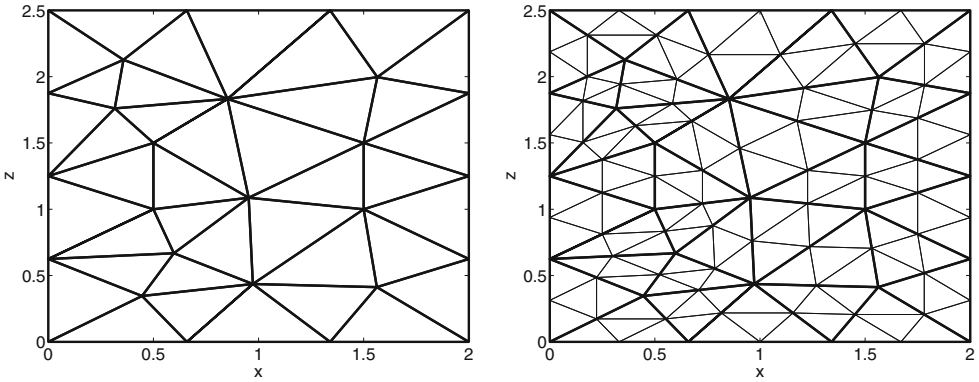

Fig. 2 Two examples of unstructured meshes. *Left* mesh *a*, the number of elements is 38. *Right* mesh *b*, this mesh is formed from mesh *a* by dividing each elements into four smaller, equisized elements

Herewith, we get for $\varepsilon \to 0$

$$|\lambda(\varepsilon) - \lambda| \leq \frac{C \max_{i,j} |\Delta H_{ij}|}{h|\mathbf{w}^{\mathbf{H}}\mathbf{v}|}\varepsilon. \tag{33}$$

Substitution of $\varepsilon = h\xi$ gives

$$|\lambda(h\xi) - \lambda| \leq \frac{C \max_{i,j} |\Delta H_{ij}|}{h|\mathbf{w}^{\mathbf{H}}\mathbf{v}|}h\xi = \frac{C \max_{i,j} |\Delta H_{ij}|}{|\mathbf{w}^{\mathbf{H}}\mathbf{v}|}\xi. \tag{34}$$

Hence, we proved:

Proposition 1 *Let* $H\mathbf{v} = \lambda\mathbf{v}$, $H \in \mathbb{R}^{n\times n}$, *and let* $\mathbf{w}^{\mathbf{H}}H = \lambda\mathbf{w}^{\mathbf{H}}$, *let* $\hat{H}_{ij}(h\xi) = H_{ij} + h\xi\Delta H_{ij}$ *where the number of non-zero entries is* $c \cdot n$, *let* C *be defined as* $C := L\sqrt{c}$, *then*

$$|\lambda(h\xi) - \lambda| \leq \frac{C \max_{i,j} |\Delta H_{ij}|}{|\mathbf{w}^{\mathbf{H}}\mathbf{v}|}\xi = \mathcal{O}(\xi), \tag{35}$$

where

$$\hat{H}(h\xi)\mathbf{v}(h\xi) = \lambda(h\xi)\mathbf{v}(h\xi).$$

The above analysis heavily relies on page 323 in [4]. If $\mathbf{w}^{\mathbf{H}}\Delta H\mathbf{v} \neq 0$, then from this proposition and Eq. (29), we can conclude that, by applying SUPG stabilization in case of an unstable system, the zero eigenvalue is mapped onto a non-zero one, where its magnitude is bounded by a value of order ξ.

At each time step, the model equations are solved in the following order. First, the porosity, permeability, fluid density and flow boundary conditions are updated. Then, the flow is calculated. Finally, the boundary conditions for the concentrations are updated and the concentrations of urea, ammonium and calcium carbonate are calculated.

4 Results

Before the results of the 100 m^3 experiment are presented, we start with a simple 2D configuration to compare the three methods to calculate the pressure on the outflow boundary, that were proposed in the last section. This is done in Subsection 4.1. Subsequently, in Subsection 4.2, we take the configuration from Fig. 1 with some known density function and calculate the pressure on the outflow boundary. Finally, in Subsection 4.3, the model results, which incorporate the numerical solution of the complete set of partial differential equations, of the 100-m^3 experiment are shown.

Table 6 Results of the variation of ξ. Errors for an application on a *structured* mesh and a *linear* relation between the depth and the density

Number of	$\xi = 10^{-1}$		$\xi = 10^{-2}$		$\xi = 10^{-5}$		$\xi = 10^{-10}$		$\xi = 10^{-15}$		$\xi = 10^{-20}$	
elements	Error	α	Error	α	Error	α	Error	α	Error	α	Error	α
8	555	1.82	323	1.47	287	1.52	287	1.52	287	0.05	287	−9.25
32	157	1.76	117	1.92	100	2.01	100	2.01	278	0.05	174,874	−1.62
128	46	1.47	30	1.83	25	2.02	25	2.02	269	2.5	536,618	0.45
512	17	1.14	8.7	1.90	6.2	1.90	6.1	2.02	47	−0.40	392,351	0.93
2,048	7.6	1.00	2.3	1.87	1.7	2.02	1.5	2.01	61	−0.21	206,335	0.57
8,192	3.8		0.64		0.41		0.38		71		138,858	

The α-factor is determined from the (unrounded) errors of two subsequent meshes, using Eq. (40)

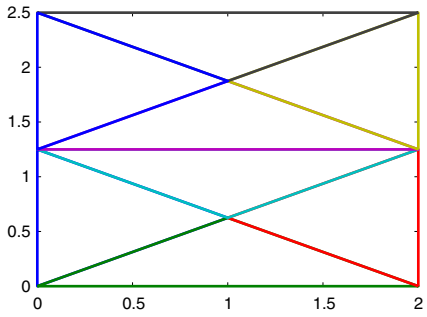

Fig. 3 The mesh that is used to examine the effect of the value of ξ

4.1 Comparison of the three methods to calculate the pressure boundary condition

As a domain, we take a rectangle with a width of 2 m and a height of 2.5 m. In this domain, we compare the three pressure calculation methods. This is done for two different relations for the density. The comparison is done for six structured meshes, with an increasing number of elements and for some unstructured meshes, with approximately the same number of elements as for the structured meshes.

As a density, the two following (arbitrarily chosen) relations between density and depth are used:

$$\rho_1 = 1,000 + 200z, \tag{36}$$

$$\rho_2 = 1,000 + 200z^3. \tag{37}$$

These relations of course do not hold at the same time. Substituting both relations into Eq. (18), gives the following two analytical solutions for the pressure:

$$\begin{aligned} p_1(x,y,z) &= p_{\text{atm}} + \int_z^{2.5} \rho_1(x,y,\bar{z}) g d\bar{z} \\ &= p_{\text{atm}} + g(1,000(2.5-z) + 100(2.5^2 - z^2)), \end{aligned} \tag{38}$$

$$\begin{aligned} p_2(x,y,z) &= p_{\text{atm}} + \int_z^{2.5} \rho_2(x,y,\bar{z}) g d\bar{z} \\ &= p_{\text{atm}} + g(1,000(2.5-z) + 50(2.5^4 - z^4)). \end{aligned} \tag{39}$$

The first method, which uses the Lower Riemann Sum as an integration technique, has a first-order error, but is exact for a constant density function. The second method, based on the Trapezium method, has a second-order error and is exact for constant and linear density functions. The third method also has a first-order error, due to the SUPG method. But since the value of ξ has been chosen very small, a better convergence is possible. When we describe the error to be of the order $O(h^\alpha)$, with h some measure for the mesh size, we expect that for a regular mesh in the limit, $\alpha = 1$ for the first and third method and $\alpha = 2$ for the second method. We calculate the value of α from the following equation:

$$\frac{e_1}{e_2} = \frac{h_1^\alpha}{h_2^\alpha} = r^\alpha, \tag{40}$$

in which e_i is the error for mesh i, $i \in \{1,2\}$, h_i is the mesh size of mesh i and r is the ratio between h_1 and h_2 ($r = h_1/h_2$). As a measure for h_i, we use Eq. (23).

In Tables 2 to 5, we display the results of the comparison of the three pressure calculation methods with the analytical solution on both a structured mesh and an unstructured mesh for an increasing number of elements and for the two relations between the density and the depth. In Table 2, the

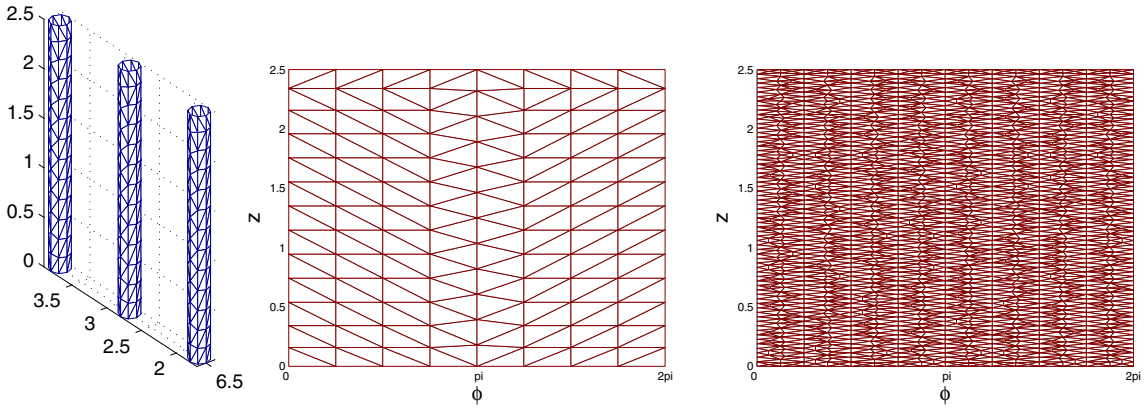

Fig. 4 Mesh on the extraction lances. The *left plot* shows the mesh on the extraction lances for mesh 1. The number of elements on the extraction lances is 618. The *middle plot* shows the same mesh on one of the extraction lances, but now as a function of the angle. Mesh 2 is formed from mesh 1 by subdividing the outflow boundary elements into four (equisized) elements. Mesh 3 is formed in the same way from mesh 2. In the *right plot*, mesh 3b is shown. This is an irregular mesh that has approximately the same number of outflow boundary elements as mesh 3. The number of elements on the extraction lances is 618 for mesh 1, 2,472 for mesh 2, 9,888 for mesh 3 and 9,600 for mesh 3b

Table 7 Comparison of the numerical solution with the analytical solution for the pressure on the outflow boundary for three meshes and two relations for the density

Mesh	Number of elements on the outflow boundaries	Density	Error	Density	Error
Mesh 1	618	ρ_1	73	ρ_2	675
Mesh 2	2,472	ρ_1	17	ρ_2	124
Mesh 3	9,888	ρ_1	4.5	ρ_2	40
Mesh 3b	9,600	ρ_1	5.9	ρ_2	41

three methods are also compared regarding computing time. In the left plot of Fig. 2, mesh a, an unstructured mesh with 38 elements, is shown. This mesh is made with the mesh generator in the COMSOL Multiphysics software. Mesh b is formed from mesh a by dividing all the elements of this mesh into four equisized elements. Mesh b is shown in the right plot of Fig. 2. In the same way, the other unstructured meshes, mesh c up to mesh f, are formed by dividing all the elements of the coarser mesh into four smaller, equisized elements. Mesh $f2$ is not formed from a coarser mesh, but directly generated with the COMSOL Multiphysics software. It has approximately the same number of elements as mesh f and is unstructured. Mesh c up to mesh $f2$ are not shown in this article.

The displayed error in Tables 2 to 5 is the mean of the (absolute value of the) error that is made in each node:

$$E = \frac{1}{n}\sum_{j=1}^{n} |p_j - p_j^E|. \tag{41}$$

In this equation for the error, n is the number of nodes, p_j is the numerical solution for the pressure in point j and p_j^E is the exact solution for the pressure in point j. The α-factor is determined from the (unrounded) errors of two subsequent meshes.

From Tables 2 to 3, we conclude that the convergence of methods 1 and 2 is as expected. For method 1, α converges to 1 and for method 2, α converges to 2. For method 3, we expected a first-order convergence but the convergence is even one order better than expected since α converges to 2. However, although the method behaves like a second-order method for a small value of ξ, the method is first order. The error looks like $E(h) = \xi h + Kh^2$. If ξ is very small, then $\xi h > Kh^2$ only if h is very small. This gives $\xi > Kh \Longleftrightarrow h < \xi/K$. This implies that if $h = \mathcal{O}(\xi)$, then the results will actually look first order. However, as long as $h \gg \xi/K$, which is used in general in practical purposes, then $E(h)$ behaves like $\mathcal{O}(h^2)$, which is observed in our experiments.

For a linear relation between density and depth, method 2 is exact and hence, the error is equal to zero. Method 1 results in the largest error. The mean error using method 3 falls within the range of the errors from methods 1 and 2.

The computing time for method 1 and method 2 is almost similar. The computing time for method 2 is slightly larger since it has an extra addition compared to method 1, see Eqs. (19) and (20). The largest part of the computing time is used for finding the various vertical lines on which the nodes lie and sorting the nodes on this lines. The computing time for method 3 is comparable with methods 1 and 2 for the coarse meshes but increases more rapidly for an increasing number of elements.

In an unstructured mesh, the nodes usually do not lie on vertical lines. Hence, methods 1 and 2 do not work properly. As can be seen from Tables 4 and 5, there is no convergence using these methods and the error even increases somewhat for an increasing number of elements. Method 3, however, also works on an unstructured mesh. The error decreases with a decreasing element size and also here we see a second-order convergence, which is one order better than expected, although the convergence is not as regular as on the structured meshes.

In Table 6, we show some results of the effect of the variation of ξ on the error of method 3. We take a mesh of 8 elements as shown in Fig. 3 and refine this mesh. As can be seen from Table 6, method 3 is first order for a high value of ξ. For values of ξ around $\xi = 10^{-2}$, the method behaves

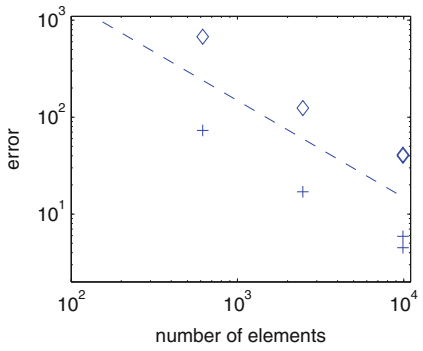

Fig. 5 The error of the density calculation on the outflow boundary plotted against the number of elements. The errors for density ρ_1 are marked with a *plus sign* and the errors for density ρ_2 are marked with a *diamond sign*. Also the trend line for an $O(h^2)$-convergence is given

Table 8 The values that have been assigned to the various model parameters

$\theta_0 = 0.41$,	$m_{CaCO_3} = 100.1$ kg/kmol,	$\mu = 1.15 \times 10^{-3}$ Pa s,
$\mathbf{D} = 0.05$[m] diag$([q_x^2\ q_y^2\ q_z^2]^T/\|\|\mathbf{q}\|\|)$,	$\rho_{CaCO_3} = 2,710$ kg/m^3,	$p_{atm} = 10^5$ Pa,
$v_{max} = 4.3681 \times 10^{-6}$ kmol/m^3/s,	$K = 0.00728$ m^3/kmol,	$g = 9.81$ m/s^2,
$K_{m,urea} = 0.01$ kmol/m^3,	$d_m = 124 \times 10^{-6}$ m.	$c_{in} = 1.00$ kmol/m^3
$Q_{in} = 5$m^3/(8×3,600 s)	$A_{in} = 4.59$ m^2	$q_{in} = 3.78 \times 10^{-5}$ m/s

second order. In this paper, we chose $\xi = 10^{-5}$. Its value can be even smaller, but for the several meshes we tried, problems arise around $\xi = 10^{-15}$. Hence, $\xi = 10^{-5}$ is a 'safe' value, which gives a second-order behaviour, for practical values of h.

4.2 Calculation of the pressure on the outflow boundary of the 100-m^3 experiment

Before simulating the 100-m^3 experiment, we first approximate the pressure on the outflow boundaries for a known density and compare this numerical solution with the analytical solution. We use the same relations between fluid density and depth as in the previous subsection: ρ_1 (36) and ρ_2 (37). We approximate the pressure, using four different meshes: mesh 1, mesh 2, mesh 3 and mesh 3b. Mesh 1 is an irregular mesh. Mesh 2 is formed from mesh 1 by dividing each outflow boundary element into four smaller, equisized elements. Mesh 3 is formed in the same way from mesh 2. Mesh 3b is another irregular mesh, with approximately the same number of outflow boundary elements as mesh 3. The three extraction lances are modelled through a prism with a regular octagon as a base. Partial differential Eq. (16) is solved on the total surface of the extraction boundaries in one matrix-vector solve, using method 3. In Fig. 4, mesh 1 (left and middle figure) and mesh 3b (right) are shown. The results of the comparison of the numerical solution with the analytical solution is shown in Table 7. As before, the error is computed from Eq. (41).

In Table 7, we see that the error decreases for an increasing number of elements, as was expected. The convergence is good, as can be seen from Fig. 5. In this figure, the error is plotted against the number of elements. The errors for density ρ_1 are marked with a $+$-sign and the errors for density ρ_2 are marked with a diamond sign. Also the trend line for an $O(h^2)$-convergence is given. It is clear that method 3 gives a quadratic convergence approximately.

In Table 7, we also compare mesh 3 and mesh 3*b*. Mesh 3*b* is a "real" irregular mesh, while mesh 3 is formed from another mesh by splitting the elements. Both meshes have approximately the same number of elements and it turns out that the error is also comparable, as was desired. The size of the boundary elements in mesh 1 is comparable with the mesh size of mesh *c*, the 2D mesh with 608 elements. In the same way, mesh 2 and mesh 3 are comparable with mesh *d* (2,432 elements) and mesh *e* (9,728 elements). For that reason, one expects that the errors are of the same order of magnitude. This turns out to be the case, although the errors in this subsection are somewhat larger. The reason might be that the geometry is more complex and that the elements are located on three different lances instead of on one rectangular 2D domain.

For mesh 1, we could also have used method 1 or method 2 to calculate the pressure on the boundaries since all the nodes lie on certain vertical lines. In mesh 2 and 3, the nodes also lie on some vertical lines. However, some of these lines do not have a node on the top edge on which the pressure is given. But after calculation of the pressure at the intersection

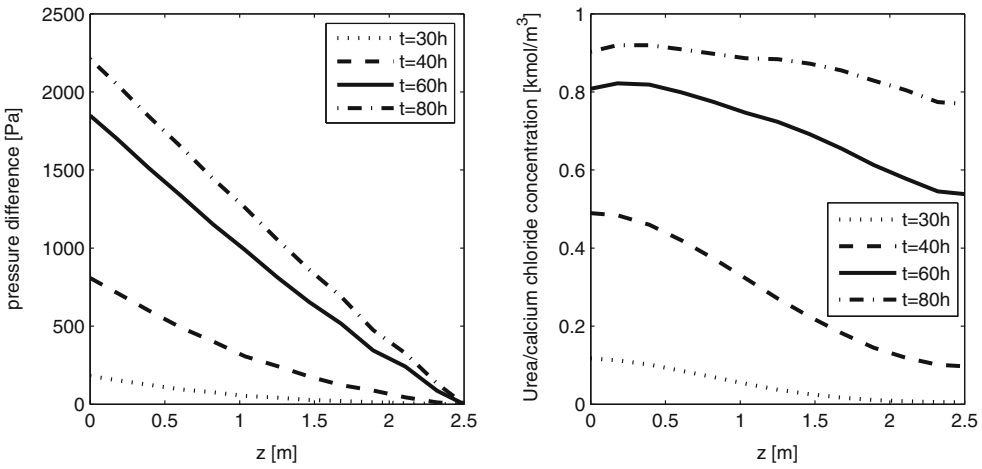

Fig. 6 *Left* The difference between the pressure at the middle extraction well and the hydrostatic pressure of water. *Right* The concentration of urea and calcium chloride at the middle extraction well

of these lines and the top edges (which is very easy in case of a constant pressure at the top), methods 1 and 2 can be used. With mesh 3b, it is not possible to use method 1 or 2 to calculate the pressure on the outflow boundary and this mesh clearly needs the application of method 3, where a small SUPG stabilization has been applied.

4.3 Application: a 100-m^3 experiment

For the 100-m^3 experiment, we take the configuration (Fig. 1), the boundary conditions (Table 1) and the initial conditions from Subsection 2.2. In this paper, the focus is on the calculation of the pressure on the boundary and not on the validation of the model. Hence, we only model the first part of the experiment. We lower the flow rate, such that the density effect becomes more visible and we extend the injection time to 80 h to allow the urea/calcium chloride pulse to reach the extraction wells. As a time step, we use $\Delta t = 0.5h$. The number of elements is approximately 23,000. The values that have been chosen for the various model parameters are shown in Table 8. Since the differential equation, the initial condition and boundary conditions are the same for urea and calcium chloride, the urea concentration is equal to the calcium chloride concentration.

In Fig. 6, we consider the pressure on the outflow boundary of the middle extraction well as a function of time. We focus on the (vertical) edge of the prism closest to the injection. The pressure on this edge is compared to the hydrostatic pressure of water with a density of $\rho_{\text{water}} = 1,000$ kg/m^3 and the difference between these two pressures is shown. This is done at several times.

During the first hours, the extraction wells are surrounded by fresh water since it takes some time before the urea/calcium chloride solution reaches the outlet. After some time, some urea and calcium chloride will reach the extraction. Due to density flow, this will happen first at the bottom of the extraction, which is illustrated by the curves for time $t = 30$ h and $t = 40$ h in the right plot of Fig. 6. As a result, only in the lower part of the extraction well the pressure (difference) increases, which results into the quadratic-like shape of the pressure difference curves in the left plot of Fig. 6 at $t = 30$ h and $t = 40$ h. Subsequently, the urea/calcium chloride concentration increases and also in higher regions the urea and calcium chloride reach the outflow, as shows the right plot of Fig. 6. Consequently, also the pressure further increases. Eventually, the urea/calcium chloride concentration is more or less constant over the depth. Due to this, finally the pressure distribution becomes linear again, while during the first time only an increase in the deeper parts was observed.

The pressure increase at the outflow at a depth of 2.5 m is approximately 2.2×10^3 Pa. We do some simple calculations to get a feeling about the magnitude of this pressure difference. The hydrostatic pressure of water at this depth is 1.25×10^5 Pa. This means that the pressure increase is 1.8 % of the hydrostatic water pressure. This is not so very much, but flow is caused by pressure gradients rather than by the pressure itself. Given the inflow flow rate Q_{in} from Table 8, the cross section of 5.6×2.5 m^2, the permeability k and viscosity μ from this simulation and a distance of 5 m between injection and extraction, it can be calculated that a pressure difference of 4.7×10^3Pa between injection and extraction is needed to generate this flow rate. (Since injection and extraction lances are used, rather than injection and extraction sheets, the difference between the highest and lowest pressure in a horizontal cross section will be somewhat larger.) Compared to the pressure difference of 4.7×10^3Pa, the pressure increase of 2.2×10^3Pa due to the higher density is 47 %. Hence, although the absolute pressure does not change so much, the pressure increase due to higher densities is significant compared to the other pressure differences, which henceforth gives a significant diversion of the flow.

Figure 7 shows a contour plot of the calcium carbonate concentration in the top and bottom of the domain at time $t = 80$ h. The highest concentrations are around the injection wells (the three blue circles left). Since there is urea and calcium chloride present during the whole simulated time,

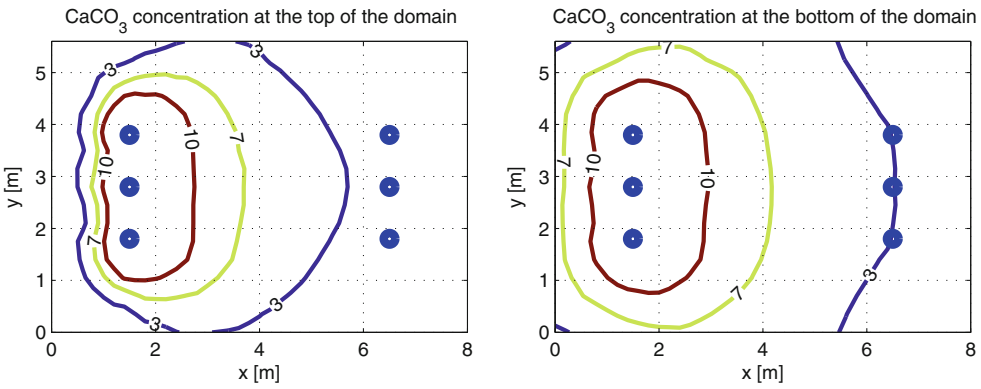

Fig. 7 The calcium carbonate content in the top (*left*) and bottom (*right*) part of the domain at time $t = 80$ h

the concentration is higher than around the extraction wells (the three blue circles right), which are only after some time reached by urea and calcium chloride. Due to density flow, the urea/calcium chloride concentration is also higher in the lower regions of the domain than in the higher ones. As a result, on the bottom, more calcium carbonate has been formed than in the top of the domain.

5 Discussion and conclusions

This article shows three different methods to calculate the pressure on the boundary. In the first method, the lower Riemann sum is used to calculate the pressure and in the second method, the trapezoid rule is used. Both methods require that all the nodes lie on certain vertical lines. In the third method, the SUPG Finite Element method is used to calculate the pressure from the solution of a boundary value problem. The large advantage is that this also works for unstructured meshes. Contrary to the first two methods, it is not required that the nodes lie on certain vertical lines.

In Subsection 4.1, the three methods are compared on a (2D) rectangle for two different relations between density and height on a structured and an unstructured mesh. Method 1 is a first-order method, while method 2 is a second-order method. That agrees with the results. Although method 3 is a first-order method, we approximate second-order convergence for a small value of ξ. We approximate first-order convergence for a larger value of ξ. The value of ξ can be chosen smaller than the value that is used in this paper, which is $\xi = 10^{-5}$. A very small value, however, deletes the Streamline Upwind part of the SUPG method and brings the method back to the Standard Galerkin method (with its instabilities). On a structured mesh, the first method gives the largest error, while the second method gives the smallest error. For unstructured meshes, only method 3 can be used to get satisfactory results.

In Subsection 4.2, the pressure is calculated on the outflow boundaries of the configuration in Fig. 1. This is done for the same relations between density and height as in Subsection 4.1. Method 3 is used to calculate the density on the outflow boundaries and this numerical solution is compared with the analytical solution. This is done on four different meshes. The results show that method 3 is a proper method to calculate the pressure on the boundary.

In Subsection 4.3, the first part of the 100-m^3 experiment as reported in [9] is simulated. Method 3 is used as a method to calculate the pressure on the outflow boundary at every time step. This results in a stable simulation and the results for the pressure are in agreement with the expectations.

The computing time for the various methods is comparable for coarse meshes, but for an increasing number of elements, methods 1 and 2 are faster than method 3. Since a 2D calculation is performed in method 3 to calculate the pressure on the boundary, the computing time will be small compared to the computing time for the 3D calculations on the full domain. For example, the computing time for the pressure on the outflow boundaries in the simulation of Subsection 4.3 is more than 800 time as small as for the 3D calculations in one time step on the full domain. Hence, the contribution to the overall CPU time is not significant, and thereby the overall CPU time is about the same for all the three methods.

We can conclude that method 3 provides a good and robust scheme to calculate the pressure on the outflow boundary. It can be applied on a manifold that represents the boundary of some domain. It works for both unstructured and structured meshes. We note that the SUPG stabilization is needed to make the method robust. For the differences in eigenvalues of the system with SUPG stabilization and the system without SUPG stabilization, we proved that $|\lambda(h\xi) - \lambda| = \mathcal{O}(\xi)$. In our observations, by applying the SUPG stabilization method, the zero eigenvalue is mapped onto a non-zero one, where its value scales with ξ. The value of ξ can be chosen very small, which for a reasonable mesh size gives the same convergence as the system without SUPG, but the value should be larger than the round off error of the computer with respect to the 8 bytes storage of floating numbers in Matlab.

Differential Eq. (16), which is solved to find the pressure on the boundary, only contains a z-derivative. The pressure in a certain point only depends on the density of the fluid straight above it. In this article, Eq. (16) is solved on the boundary (elements) only. This approach only works if the faces of the boundary are vertical, such that the fluid that determines the pressure in an arbitrary point at the boundary is on the boundary itself. If this is not the case, considering the (2D) boundary elements only is not sufficient. Instead, one should consider the 3D region above the boundary, since the density in this region determines the pressure on the boundary. Subsequently, one should solve differential Eq. (16) on this 3D subdomain to approximate the pressure on the boundary.

Acknowledgments This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by Ministry of Economic Affairs, Agriculture and Innovation.

Appendix 1: List of symbols

C^i = concentration of specie i, for $i \in \{\text{urea}, Ca^{2+}, NH_4^+\}$, [in kilomole per cubic meter];

$C^{\mathrm{CaCO_3}}$ = concentration of calcium carbonate, [in kilograms per cubic meter];
D_i = dispersion tensor for specie i, for $i \in \{\text{urea}, Ca^{2+}, NH_4^+\}$, [in square meter per second];
d_{m} = mean particle size of the sand, [in meters];
g = gravitational constant, [in meters per square second];
K = constant in the differential equation for the flow that represents the amount of volume that is produced per kilomole formed calcium carbonate, [cubic meters per kilomole];
k_i = intrinsic permeability in the various coordinate directions, $i \in \{x, y, z\}$, [in square meter];
$K_{m,\mathrm{urea}}$ = saturation constant for urea, [in kilomole per cubic meter];
$m_{\mathrm{CaCO_3}}$ = molar mass of calcium carbonate, [in kilograms per kilomole];
p = pressure, [in pascal]; p_{atm} = atmospheric pressure;
$\mathbf{q}$ = Darcy velocity, [in meters per second];
r_{hp} = reaction rate of the hydrolysis and precipitation processes, [in kilomole per cubic meter per second];
t = time, [in seconds];
$v_{\max}$ = bacterial conversion rate constant, [in kilomole per cubic meter per second];
x, y, z = Cartesian coordinates, [in meters];
θ = porosity, [in cubic meter per cubic meter];
μ = viscosity of the fluid, [pascal second];
$\rho_{\mathrm{CaCO_3}}$ = density of calcium carbonate, [in kilogram per cubic meter];
ρ_l = density of the fluid, [in kilograms per cubic meter];
$1 - V_s$ = decrease in liquid volume per kilomole formed calcium carbonate, [cubic meters per kilomole].

References

1. Adams, R.A.: Calculus: a complete course. Addison Wesley Longman, Toronto (2003)
2. Bear, J.: Dynamics of Fluids in Porous Media, pp. 59–64; 119–194. Dover Publications, New York (1972)
3. Budd, C.J., Iserles, A.: Geometric integration: numerical solution of differential equations on manifolds. Philos. Trans. R. Soc. Lond. Ser. Math. Phys. Eng. Sci. **357**, 945–956 (1999)
4. Golub, G.H., Van Loan, Ch.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
5. Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proc. Natl. Acad. Sci. **95**, 8431–8435 (1998)
6. Madzvamuse, A., Thomas, R.D.K., Maini, P.K., Wathen, A.J.: A numerical approach to the study of spatial pattern formation in the ligaments of arcoid bivalves. Bull. Math. Biol. **64**, 501–530 (2002)
7. Sekimura, T., Madzvamuse, A., Wathen, A.J., Maini, P.K.: A model for colour pattern formation in the butterfly wing of *Papilio dardanus*. Proc. R. Soc. Lond. B Biol. Sci. **267**(1446), 851–859 (2000)
8. Van Kan, J., Segal, A., Vermolen, F.: Numerical Methods in Scientific Computing, pp. 122–126. VSSD (2005)
9. Van Paassen, L.A., Harkes, M.P., van Zwieten, G.A., van der Zon, W.H., van der Star, W.R.L., van Loosdrecht, M.C.M.: Scale up of Biogrout: a biological ground reinforcement method. In: Hamza, M. et al. (eds.) Proceedings of the 17th International Conference on Soil Mechanics and Geotechnical Engineering, pp. 2328–2333 (2009)
10. Van Paassen, L.A., Daza, C.M., Staal, M., Sorokin, D.Y., Van der Zon, W.H., Van Loosdrecht, M.C.M.: Potential soil reinforcement by biological denitrification. Ecol. Eng. **36**(2), 168–175 (2010)
11. Van Wijngaarden, W.K., Vermolen, F.J., Van Meurs, G.A.M., Vuik, C.: Modelling the new soil improvement method Biogrout: extension to 3D. In: Kreiss, G. et al. (eds.) Numerical Mathematics and Advanced Applications, pp. 893-900 (2010)
12. Van Wijngaarden, W.K., Vermolen, F.J., Van Meurs, G.A.M., Vuik, C.: Modelling Biogrout: a new ground improvement method based on microbial-induced carbonate precipitation. Transp. Porous Media **87**, 397–420 (2011)
13. Van Wijngaarden, W.K., Vermolen, F.J., Van Meurs, G.A.M., Vuik, C.: A mathematical model and analytical solution for the fixation of bacteria in Biogrout. Transp. Porous Media **92**, 847–866 (2012)
14. Van Wijngaarden, W.K., Vermolen, F.J., Van Meurs, G.A.M., Vuik, C.: Various flow equations to model the new soil improvement method Biogrout.: In: Numerical Mathematics and Advanced Applications 2011, pp. 633–641. Springer, Berlin Heidelberg (2013)
15. Van Wijngaarden, W.K.: Modelling biogrout—Extension to Two Dimensions and a First Glance in Three Dimensions. Delft University of Technology (2008)
16. Weast, R.C.: Handbook of Chemistry and Physics, 60th edn. CRC, Boca Raton (1980)
17. Whiffin, V.S., van Paasen, L.A., Harkes, M.P.: Microbial carbonate precipitation as a soil improvement technique. Geomicrobiol J **24**(5), 417–423 (2007)
18. Zheng, C., Bennett, G.D.: Applied Contaminant Transport Modeling, pp. 3–79. Van Nostrand Reinhold, New York (1995)