Comput Mech (2005) 36: 409–420
DOI 10.1007/s00466-004-0654-1

ORIGINAL PAPER

C. Wieners · M. Ammann · T. Graf · W. Ehlers

Parallel Krylov methods and the application to 3-d simulations of a triphasic porous media model in soil mechanics

Received: 22 September 2004 / Accepted: 21 December 2004 / Published online: 10 August 2005

Abstract We introduce a general parallel model for solving coupled nonlinear and time-dependent problems in soil mechanics, where we employ general purpose linear solvers with specially adjusted preconditioners. In particular, we present a parallel realization of the GMRES method applied to a triphasic porous media model in soil mechanics, where we compute the deformation of unsaturated soil together with the pore-fluid flow of water and air in the soil. Therefore, we propose a pointwise preconditioner coupling all unknowns at the nodal points. In two large-scale numerical experiments we finally present an extended evaluation of our parallel model for demanding configurations of the triphasic model.

Keywords Parallel computing · Krylov methods · Theory of Porous Media · Non-associated elasto-viscoplasticity

1 Introduction

The parallel simulation of realistic finite element models in geotechnical engineering is a difficult task, since the required resolution for reliable results yields very large systems. Therefore, efficient iterative linear solvers are indispensable. In particular, detailed models in soil mechanics with several independent variables result in coupled problems which together have a linearization of saddle point structure, which is in addition non-symmetric. Thus, simple solvers such as the cg-method cannot be applied, and more advanced Krylov methods which are well suited for the applications to general systems are required.

Here, we show that the GMRES method as a general purpose iterative linear solver can be applied successfully to advanced problems in soil mechanics, provided that a suitable preconditioner is available. In our application, we are interested in the numerical simulation of a coupled, triphasic model including the soil deformation and the pore-fluid flow. In detail, these phases are a porous, elasto-viscoplastic, materially incompressible solid skeleton φ^S, a viscous, materially incompressible pore-liquid φ^L and a viscous, materially compressible pore-gas φ^G. Note that such a complex multiphasic material model is necessary for a realistic description of unsaturated soil conditions [6]. It turns out, that a coupled approach for all phases in the preconditioner leads to an efficient numerical scheme, which is well suited for parallelization, since it can be realized within the parallel linear algebra module of our general model for parallel finite elements introduced in [16].

This contribution is organized as follows: We give a short introduction into the underlying material model for the numerical simulation, and we comment on the finite element realization. Then, we describe our parallel model and the realization of parallel Krylov methods, where we study in particular the GMRES method. Finally, we apply this scheme to the numerical simulation of slope failure problems, which are initiated due to an increasing ground-water table or an extremely heavy rainfall.

2 A triphasic porous media model in soil mechanics

In this section, we summarize briefly the governing equations of the triphasic model which we use for the numerical experiments in this contribution. For a more detailed derivation of the underlying triphasic porous media model within the framework of the well-founded

W. Ehlers (✉) · T. Graf · M. Ammann
Institut für Mechanik (Bauwesen),
Universität Stuttgart, 70 550 Stuttgart,
Germany, www.mechbau.uni-stuttgart.de/ls2
E-mail: ehlers@mechbau.uni-stuttgart.de

C. Wieners
Institut für Praktische Mathematik, Universität Karlsruhe,
76 128 Karlsruhe, Germany,
www.mathematik.uni-karlsruhe.de
E-mail: wieners@math.uni-karlsruhe.de

Theory of Porous Media (TPM), we refer to Ehlers et al. [5, 6]. For a general view on the TPM, cf. the work by de Boer [3] or Lewis and Schrefler [9]. The present model is based on the concept of volume fractions $n^\alpha = n^\alpha(\mathbf{x}, t)$ which are defined as the local ratios of the constituent volumes v^α with respect to the bulk volume v, where the index $\alpha = S$ describes the solid phase, and $\alpha = F$ corresponds to the fluid phase consisting of the constituents φ^L ($\alpha = L$, liquid) and φ^G ($\alpha = G$, gas). Thus, the saturation condition yields (in the absence of any vacant space)

$$n^S + n^F = 1 \quad \text{with} \quad n^F = n^L + n^G \ , \tag{1}$$

where the overall fluid volume fraction n^F also represents the porosity of the soil. The saturation of the fluid constituents liquid and gas within the fluid phase is described by

$$s^\beta = \frac{n^\beta}{n^F}, \quad \beta \in \{L, G\} \ , \tag{2}$$

obviously satisfying the saturation constraint

$$s^L + s^G = 1. \tag{3}$$

Associated with each constituent φ^α is an effective (realistic) or material density

$$\rho^{\alpha R} = \frac{\mathrm{d}m^\alpha}{\mathrm{d}v^\alpha}$$

defining the local mass of the constituent φ^α per unit of v^α. Combined with the volume fractions $n^\alpha = \mathrm{d}v^\alpha/\mathrm{d}v$, this gives the partial or bulk density

$$\rho^\alpha = \frac{\mathrm{d}m^\alpha}{\mathrm{d}v} = n^\alpha \rho^{\alpha R} \ .$$

Thus, it is evident that the property of material incompressibility of a constituent φ^α (defined by $\rho^{\alpha R} = \text{const.}$) is not equivalent to global incompressibility of this constituent, since the partial density can still change through changes in the volume fractions n^α.

The model under study is governed by the primary variables solid displacement $\mathbf{u}_S$ and the effective pore-fluid pressures p^{LR} and p^{GR}. Proceeding from quasi-static conditions, the set of governing equations is given by the vector-valued overall momentum balance corresponding to $\mathbf{u}_S$, the scalar-valued fluid volume or mass balance equations corresponding to p^{LR} and p^{GR} and additional constitutive relations. Note that instead of this pressure-pressure formulation alternatively a pressure-saturation formulation can be used for the description of the pore-fluid flow, cf. Helmig et al. [7, 8].

The overall momentum balance yields

$$\operatorname{div}\big(\boldsymbol{\sigma}^S_E - p\mathbf{I}\big) + \rho\mathbf{g} = \mathbf{0} \ , \tag{4}$$

where $\boldsymbol{\sigma}^S_E$ is the effective solid stress, p is the effective pore-pressure exceeding the atmospheric pressure p_0, ρ is the overall density, and $\mathbf{g}$ is the overall gravity.

For the description of the solid skeleton deformation, we consider the (linearized) solid strain tensor

$$\boldsymbol{\varepsilon}_S = \frac{1}{2}(\operatorname{grad} \mathbf{u}_S + \operatorname{grad}^T \mathbf{u}_S) \tag{5}$$

(obtained from the solid displacement $\mathbf{u}_S$ through the kinematical compatibility condition), which is additively split into elastic and plastic parts

$$\boldsymbol{\varepsilon}_S = \boldsymbol{\varepsilon}_{Se} + \boldsymbol{\varepsilon}_{Sp} \ . \tag{6}$$

The effective solid stress

$$\boldsymbol{\sigma}^S_E = 2\mu^S \boldsymbol{\varepsilon}_{Se} + \lambda^S(\boldsymbol{\varepsilon}_{Se} \cdot \mathbf{I})\mathbf{I} \tag{7}$$

is given by the Hookean law, where μ^S and λ^S are the bulk Lamé constants of the porous solid skeleton. Finally, the plastic evolution of the internal variables is given by a viscoplastic flow rule of Perzyna type [12]

$$(\boldsymbol{\varepsilon}_{Sp})'_S = \frac{1}{\eta}\left\langle \frac{F(\boldsymbol{\sigma}^S_E)}{\sigma_0} \right\rangle^r \frac{\partial G(\boldsymbol{\sigma}^S_E)}{\partial \boldsymbol{\sigma}^S_E} \ . \tag{8}$$

Here, the material time derivative following the motion of φ^S is denoted by $(\cdot)'_S = \partial(\cdot)/\partial t + \operatorname{grad}(\cdot)\mathbf{x}'_S$, where $\mathbf{x}'_S$ is the velocity field of the solid motion function. Furthermore, η is the relaxation time, σ_0 the reference stress, r the viscoplastic exponent, and we use the notation $\langle x \rangle = \max\{0, x\}$. Since porous soil materials are generally known as frictional materials, the concept of non-associated plasticity has to be applied, where the yield function [4]

$$\begin{aligned} F &= \Phi^{1/2} + \beta \mathrm{I} + \epsilon \mathrm{I}^2 - \kappa = 0, \\ \Phi &= \mathrm{II}^D(1 + \gamma\vartheta)^m + \frac{1}{2}\alpha \mathrm{I}^2 + \delta^2 \mathrm{I}^4, \\ \vartheta &= \mathrm{III}^D/(\mathrm{II}^D)^{3/2}, \end{aligned} \tag{9}$$

is different from the plastic potential [10]

$$\begin{aligned} G &= \Gamma^{1/2} + \psi_2 \mathrm{I} + \epsilon \mathrm{I}^2, \\ \Gamma &= \psi_1 \mathrm{II}^D + \frac{1}{2}\alpha \mathrm{I}^2 + \delta^2 \mathrm{I}^4. \end{aligned} \tag{10}$$

Here, I, II^D and III^D are the first and deviatoric (negative) second and third principal invariants of $\boldsymbol{\sigma}^S_E$. Furthermore, (9) and (10) contain the material parameters α, β, γ, δ, ϵ, κ, m, ψ_1, ψ_2, which have to be adjusted to experimental results [11].

In the momentum balance equation (4), the effective pressure is given by

$$p = s^L p^{LR} + s^G p^{GR} \ , \tag{11}$$

and the overall density reads

$$\rho = n^S \rho^{SR} + n^L \rho^{LR} + n^G \rho^{GR} \ . \tag{12}$$

Consequently, relations for the saturation s^L_v of the pore-liquid s^L, for the volume fraction n^S of the solid and for the density ρ^{GR} of the pore-gas have to be specified. Following the ansatz of van Genuchten [15], the pore-liquid saturation

$$\begin{aligned} s^L &= s^L_{\text{eff}}(1 - s^L_{\text{res}} - s^G_{\text{res}}) + s^L_{\text{res}} \, , \\ s^L_{\text{eff}} &= [1 + (\alpha_{\text{gen}}\, p^C)^{j_{\text{gen}}}]^{-h_{\text{gen}}}, \end{aligned} \tag{13}$$

is expressed as a function of the capillary pressure

$$p^C = p^{GR} - p^{LR} \ , \tag{14}$$

where the quantities s^L_{res}, s^G_{res}, α_{gen}, j_{gen} and h_{gen} are material parameters, cf. Fig. 1. The integration of the volume balance of the solid and, thereafter, the linearization around the natural state yields the relation for the determination of the volume fraction

$$n^S = n^S_{0S}(1 - \text{ div } \mathbf{u}_S), \tag{15}$$

with respect to the initial volume fraction n^S_{0S}. Finally, Boyle's ideal gas law serves as the constitutive relation for the determination of the density

$$\rho^{GR} = \frac{p^{GR} + p_0}{\bar{R}^G \theta} \tag{16}$$

depending on the excess pressure p^{GR}, the atmospheric pressure p_0, the specific gas constant $\bar{R}^G$ of the pore-gas, and the absolute Kelvin's temperature θ, which is constant due to the assumption of an overall isothermal problem.

As was already mentioned above, the volume balance of the pore-liquid and the mass balance of the pore-gas are finally needed for the determination of the pore-fluid pressures p^{LR} and p^{GR},

$$\begin{aligned} &(n^L)'_S + n^L \text{ div}(\mathbf{u}_S)'_S + \text{ div}(n^L \mathbf{w}_L) = 0, \\ &(n^G \rho^{GR})'_S + n^G \rho^{GR} \text{ div}(\mathbf{u}_S)'_S + \text{ div}(\rho^{GR} n^G \mathbf{w}_G) = 0, \end{aligned} \tag{17}$$

depending on the Darcy-type filter velocities $n^\beta \mathbf{w}_\beta$:

$$\begin{aligned} n^L \mathbf{w}_L = &- \frac{K^S_{0S}}{\mu^{LR}} \left(s^L_{\text{eff}}\right)^{\epsilon_{\text{gen}}} \left\{ 1 - \left[1 - \left(s^L_{\text{eff}}\right)^{1/h_{\text{gen}}} \right]^{h_{\text{gen}}} \right\}^2 \\ &\times \left(\frac{1 - n^S}{1 - n^S_{0S}} \right)^\pi (\text{ grad } p^{LR} - \rho^{LR} \mathbf{g}), \\ n^G \mathbf{w}_G = &- \frac{K^S_{0S}}{\mu^{GR}} \left(1 - s^L_{\text{eff}}\right)^{\gamma_{\text{gen}}} \left[1 - \left(s^L_{\text{eff}}\right)^{1/h_{\text{gen}}} \right]^{2h_{\text{gen}}} \\ &\times \left(\frac{1 - n^S}{1 - n^S_{0S}} \right)^\pi (\text{ grad } p^{GR} - \rho^{GR} \mathbf{g}). \end{aligned} \tag{18}$$

Here, K^S_{0S} is the initial intrinsic permeability coefficient, $\mu^{\beta R}$ are the effective shear viscosities, ϵ_{gen}, γ_{gen} are additional parameters for the relative permeability functions within the van Genuchten model, and π is a material parameter governing the exponential behavior of the deformation dependence of the intrinsic permeability [6].

3 Weak formulation of the triphasic model

For the numerical realization of the model, the independent primary variables solid displacement $\mathbf{u}_S$ and effective pore-fluid pressures $p^L := p^{LR}$ and $p^G := p^{GR}$ are approximated by finite elements and determined by their values at the nodal points. They are uniquely defined by the three balance relations (4) and $(17)_{1,2}$, which we have to provide an integrated form in order to obtain a weak finite element formulation. This is complemented by the flow rule determining the internal variables $\boldsymbol{\varepsilon}_p := \boldsymbol{\varepsilon}_{Sp}$ which are represented by Gauss point values.

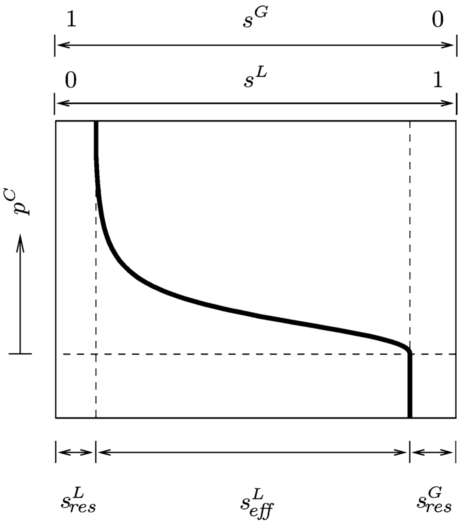

Fig. 1 Qualitative sketch of the capillary-pressure-saturation relation according to [15]

All other quantities are determined by the constitutive equations, and they can be computed directly from the primary variables:

$\boldsymbol{\sigma}^S_E(\mathbf{u}_S, \boldsymbol{\varepsilon}_p)$	solid extra stresses from (5) – (7),
$s^L(p^L, p^G)$	liquid saturation from(13) and (14),
$s^G(p^L, p^G)$	gaseous saturation from (3),
$n^S(\mathbf{u}_S)$	solid volume fraction from (15),
$n^L(\mathbf{u}_S, p^L, p^G)$	liquid vol. frac. from $(1)_1$ and (2),
$n^G(\mathbf{u}_S, p^L, p^G)$	gaseous volume fraction from (2),
$\rho^{GR}(p^G)$	gaseous density from (16),
$\tilde{\mathbf{w}}_L(\mathbf{u}_S, p^L, p^G) := n^L \mathbf{w}_L$	liq. filter vel. from $(18)_1$,
$\tilde{\mathbf{w}}_G(\mathbf{u}_S, p^L, p^G) := n^G \mathbf{w}_G$	gas. filter vel. from $(18)_2$.

For the weak formulation, we have to specify in addition boundary conditions. Therefore, let $\Gamma^S \subset \partial\Omega$, $\Gamma^L \subset \partial\Omega$ and $\Gamma^G \subset \partial\Omega$ be the boundary parts, where we prescribe Dirichlet boundary conditions $\mathbf{u}_S = \mathbf{d}^S$, $p^L = d^L$ and $p^G = d^G$, respectively. Using the above definitions and the equations (11) and (12), we obtain the weak formulation of the overall momentum balance (4)

$$\begin{aligned} &\int_\Omega \left\{ \boldsymbol{\sigma}^S_E(\mathbf{u}_S, \boldsymbol{\varepsilon}_p) - \left[s^L(p^L, p^G) p^L \right. \right. \\ &\qquad \left. \left. + s^G(p^L, p^G) p^G \right] \mathbf{I} \right\} \cdot \text{ grad } \delta\mathbf{u}_S \text{d}v \\ &- \int_\Omega \left[n^S(\mathbf{u}_S) \rho^{SR} + n^L(\mathbf{u}_S, p^L, p^G) \rho^{LR} \right. \\ &\qquad \left. + n^G(\mathbf{u}_S, p^L, p^G) \rho^{GR}(p^G) \right] \mathbf{g} \cdot \delta\mathbf{u}_S \text{ d}v \\ &= \int_{\partial\Omega} \bar{\mathbf{t}} \cdot \delta\mathbf{u}_S \text{d}a \end{aligned}$$

for all test functions $\delta\mathbf{u}_S$ satisfying $\delta\mathbf{u}_S = \mathbf{0}$ on Γ^S. The Neumann boundary condition $\bar{\mathbf{t}}$ (surface traction) is a given quantity depending on the respective initial boundary-value problem. Note that the term $\boldsymbol{\sigma}_E^S(\mathbf{u}_S, \boldsymbol{\varepsilon}_p)$ includes the solution of the evolution equation (8) for the determination of the new plastic strain $\boldsymbol{\varepsilon}_p$, cf. Wieners et al. [16].

Using the abbreviation $\dot{(\cdot)} := (\cdot)'_S$, the weak formulation of the pore-liquid volume balance $(17)_1$ reads

$$\int_\Omega \left[\dot{n}^L\left(\mathbf{u}_S,p^L,p^G\right) + n^L\left(\mathbf{u}_S,p^L,p^G\right)\mathrm{div}\,\dot{\mathbf{u}}_S\right]\delta p^L\,\mathrm{d}v$$
$$-\int_\Omega \tilde{\mathbf{w}}_L\left(\mathbf{u}_S,p^L,p^G\right)\cdot\ \mathrm{grad}\ \delta p^L\,\mathrm{d}v = \int_{\partial\Omega} \bar{v}^L\delta p^L\,\mathrm{d}a$$

for all test functions δp^L satisfying $\delta p^L = 0$ on Γ^L. Finally, we obtain the weak formulation of the pore-gas mass balance $(17)_2$

$$\int_\Omega \Big\{\left[\dot{n}^G\left(\mathbf{u}_S,p^L,p^G\right)\rho^{GR}\left(p^G\right) + n^G\left(\mathbf{u}_S,p^L,p^G\right)\dot{\rho}^{\mathrm{GR}}\left(p^G\right)\right]$$
$$+\,n^G\left(\mathbf{u}_S,p^L,p^G\right)\rho^{GR}\left(p^G\right)\ \mathrm{div}\,\dot{\mathbf{u}}_S\Big\}\delta p^G\,\mathrm{d}v$$
$$-\int_\Omega \rho^{GR}\left(p^G\right)\tilde{\mathbf{w}}_G\left(\mathbf{u}_S,p^L,p^G\right)\cdot\ \mathrm{grad}\ \delta p^G\mathrm{d}v$$
$$=\int_{\partial\Omega}\bar{q}^G\delta p^G\mathrm{d}a$$

for all test functions δp^G satisfying $\delta p^G = 0$ on Γ^G. Again, Neumann boundary conditions $\bar{v}^L$ (efflux of liquid volume) and $\bar{q}^G$ (efflux of gaseous mass) are prescribed.

For the fully discretized model, we apply an implicit Euler method in time. As this procedure was already described in detail for a biphasic model, the interested reader is referred to [16].

4 A parallel preconditioned GMRES method

We shortly summarize the parallel programming model introduced in [17], and then we explain the parallel realization of preconditioners and Krylov methods within this model. In the following, P denotes the number of processors ($P = 1$ corresponds to a sequential program). We assume that the computational domain is decomposed into tetrahedral cells $C \in \mathcal{C}$. The nodal points associated to a tetrahedral cell C are denoted by $\mathcal{N}_C$. Furthermore, we assign N_x degrees of freedom to a nodal point $x \in \mathcal{N}_C$, cf. Fig. 2.

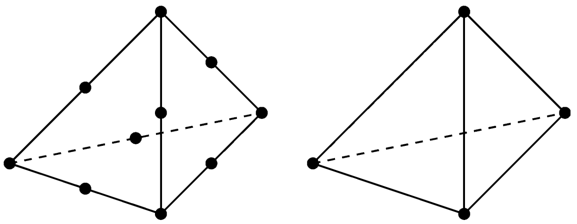

Fig. 2 In our application, we use quardratic approximation for the displacement and linear approximations for the pressure fields. Thus, $\mathcal{N}_C$ consists of all corners and edge midpoints of the tetrahedron C, and we have $N_x = 5$ for corners $x \in \mathcal{N}_C$ and $N_x = 3$ for edge midpoints $x \in \mathcal{N}_C$

Let $\mathcal{P} = \{0, \ldots, P-1\}$ be the set of processors (or, more precisely, the identification numbers of the parallel processes), and let

$$\mathcal{C} = \bigcup_{p\in\mathcal{P}} \mathcal{C}_p$$

be a disjoint parallel distribution of the cells. This defines the set of nodal points on processor p

$$\mathcal{N}_p = \bigcup_{C\in\mathcal{C}_p} \mathcal{N}_C \subset \mathbb{R}^3$$

and the overlapping partition of nodal points

$$\mathcal{N} = \bigcup_{p\in\mathcal{P}} \mathcal{N}_p \subset \mathbb{R}^3.$$

Let

$$N := \sum_{x\in\mathcal{N}} N_x$$

be the total number of unknowns, and let

$$N_p := \sum_{x\in\mathcal{N}_p} N_x$$

be the number of unknowns on processor $p \in \mathcal{P}$. Note that for $P > 1$

$$N < \sum_{p\in\mathcal{P}} N_p.$$

For the representation of the parallel distribution of $\mathcal{N}$, we introduce a partition map π: for any point $x \in \mathcal{N}_p$, the distribution defines the corresponding processor set

$$\pi_p(x) := \left\{q \in \mathcal{P} : x \in \mathcal{N}_q\right\} \subset \mathcal{P}\ .$$

Note that this is consistent in parallel, i. e.,

$$q \in \pi_p(x) \Longleftrightarrow p \in \pi_q(x)$$

for all $x \in \mathcal{N}_p \cap \mathcal{N}_q$ and $p, q \in \mathcal{P}$. Thus, $(\pi_p)_{p\in\mathcal{P}}$ combines to a partition map

$$\pi : \mathcal{N} \longrightarrow 2^{\mathcal{P}}$$

which is represented in parallel by its distributed restrictions $(\pi_p)_{p\in\mathcal{P}}$. Finally, we assign a master processor $\mu(x) = \min \pi(x)$ to every nodal point $x \in \mathcal{N}$.

We introduce a numbering on $\mathcal{N}_p$ by lexicographic ordering in $\mathbb{R}^3$:

$$\begin{aligned} x < y \quad \Longleftrightarrow \quad & x_1 < y_1 \text{ or} \\ & x_1 = y_1 \text{ and } x_2 < y_2 \text{ or} \\ & x_1 = y_1 \text{ and } x_2 = y_2 \text{ and } x_3 < y_3. \end{aligned}$$

Based on this numbering, we can assign to any vector $\underline{u}_p \in \mathbb{R}^{N_p}$ a sub-vector $\underline{u}(x) \in \mathbb{R}^{N_x}$ for $x \in \mathcal{N}_p$.

In parallel, we distinguish two representations of distributed vectors, cf. [2]:

(a) *Consistent vector representation*

For solution vectors and correction vectors, we require

$$\underline{u}_p(x) = \underline{u}_q(x), \quad x \in \mathcal{N}_p \cap \mathcal{N}_q,\ p, q \in \mathcal{P} \ .$$

This defines a global vector $\underline{u} \in \mathbb{R}^N$ by $\underline{u}\ (x) = \underline{u}_q(x)$ for any $q \in \pi(x)$.

(b) *Additive vector representation*

Load vectors and residual vectors are represented additively, i. e., the corresponding global vector $\underline{r}$ is defined by

$$\underline{r}(x) = \sum_{p\in\pi(x)} \underline{r}_p(x), \quad x \in \mathcal{N} \ .$$

This corresponds naturally to the parallel integration of load vectors. Collecting the distributed values at the master nodal points and replacing $\underline{r}_p$ by $\underline{r}_p(x) = \underline{r}(x)$ for $p = \mu(x)$ and $\underline{r}_q(x) = 0$ else, results in a unique additive representation. This allows for the parallel evaluation of the norm $\sqrt{\underline{r}^T\underline{r}} = \sqrt{\sum_{p\in\mathcal{P}} \underline{r}_p^T\underline{r}_p}$.

The stiffness matrix is also additively represented by matrices $\underline{A}_p \in \mathbb{R}^{N_p,N_p}$. Again, $\underline{A}_p(x,y) \in \mathbb{R}^{N_x,N_y}$ refers to the block matrices corresponding to the connection of two nodal points. Note that (for parallel iterative solvers) it is not necessary to assemble the full stiffness matrix entries

$$\underline{A}(x,y) = \sum_{p\in\pi(x)\cap\pi(y)} \underline{A}_p(x,y), \quad x,y \in \mathcal{N}_p.$$

In our model, we have $\underline{A}(x,y) = 0$ for $x \in \mathcal{N}_p$ and $y \notin \mathcal{N}_p$, since the finite element assembling results in $\underline{A}_p(x,y) \neq 0$ only if $x,y \in \mathcal{N}_C \subset \mathcal{N}_p$ are nodal points in one cell C on processor p.

For a given consistent vector $\underline{u}$, we define the additive result $\underline{r} = \underline{A}\,\underline{u}$ by local matrix-vector products $\underline{r}_p = \underline{A}_p\underline{u}_p$ (without communication). Then, by an interface communication on $\{x \in \mathcal{N} : |\pi(x)| > 1\}$, we collect the results $\left(\underline{r}_p(x)\right)_{p\in\mathcal{P}}$ for obtaining a unique additive representation of $\underline{r}$ on the master nodal points.

Now, we consider the linear problem

$$\underline{u} \in \mathbb{R}^N : \quad \underline{A}\,\underline{u} = \underline{f}$$

for a given right-hand side $\underline{f}$. A basic linear iteration for solving this problem requires a preconditioner $\underline{B}$ (approximating $\underline{A}^{-1}$) and runs as follows: given a start iterate $\underline{u}^0 \in \mathbb{R}^N$, we compute for $m = 0, 1, 2, \ldots$ the residual $\underline{r}^m = \underline{f} - \underline{A}\underline{u}^m$, the correction $\underline{c}^m = \underline{B}\underline{r}^m$, and the update $\underline{u}^{m+1} = \underline{u}^m + \underline{c}^m$.

A parallel preconditioner is an operator $\underline{B}$ which transforms additive residual vectors $\underline{r}$ into consistent correction vectors $\underline{c} := \underline{B}\,\underline{r}$. Here, we consider two examples:

(a) *Point-Block Jacobi preconditioner*

First, by an interface communication, we construct the consistent point-block diagonal entries

$$\underline{A}(x,x) = \sum_{p\in\pi(x)} \underline{A}_p(x,x) \in \mathbb{R}^{N_x,N_x}$$

(where point-block objects collect the degrees of freedom corresponding to a single nodal point $x \in \mathcal{N}$). We make the general assumption that $\underline{A}(x,x)$ is regular for all $x \in \mathcal{N}$. Then, independently on all $p \in \mathcal{P}$ for a given additive part $\underline{r}_p$, we define

$$\underline{c}_p^{\mathrm{add}}(x) = \underline{A}(x,x)^{-1}\underline{r}_p(x)$$

for $x \in \mathcal{N}_p$, followed by an interface communication

$$\underline{c}_p(x) = \sum_{q\in\pi(x)} \underline{c}_q^{\mathrm{add}}(x), \quad x \in \mathcal{N}_p \tag{19}$$

defining the consistent correction vector $\underline{c} = \underline{B}_{\mathrm{jac}}\underline{r}$.

(b) *Block Jacobi preconditioner with Gauss-Seidel blocks*

Here, we also construct the consistent point-block diagonal entries $\underline{A}(x,x)$, and then we perform (independently on every processor) a point-block Gauss-Seidel iteration on every block (corresponding to $\mathcal{N}_p$): for a given additive part $\underline{r}_p$, we define the local Gauss-Seidel correction by

$$\underline{c}_p^{\mathrm{add}}(x) = \underline{A}(x,x)^{-1}\left(\underline{r}_p(x) - \sum_{y<x} \underline{A}_p(x,y)\underline{c}_p^{\mathrm{add}}(y)\right)$$

for $x, y \in \mathcal{N}_p$, followed by the interface communication (19). This defines $\underline{c} = \underline{B}_{\mathrm{gs}}\underline{r}$.

For the basic linear iteration, we observe that the error satisfies the identity

$$\underline{u}^m - \underline{u} = (\mathrm{id}\ - \underline{B}\,\underline{A})^m(\underline{u}^0 - \underline{u}).$$

Thus, the iterate $\underline{u}^m$ is the affine space

$$\underline{u}^m \in \left\{\underline{u}^0 + \underline{B}\,\underline{w} : \underline{w} \in \mathcal{K}^m\right\} \ , \tag{20}$$

where

$$\mathcal{K}^m = \ \mathrm{span}\ \left\{(\underline{A}\,\underline{B})^k\underline{r}^0 : k = 0, \ldots, m-1\right\}$$

is the corresponding Krylov vector space of dimension m. There exist several Krylov methods to construct iterations satisfying (20) such that the error $\underline{u}^m - \underline{u}$ is minimal with respect to some norm, e. g., the cg-method is a Krylov method minimizing the error in the energy norm. Note that the cg-method is restricted to symmetric positive definite matrices $\underline{A}$ and $\underline{B}$.

GMRES (generalized minimal residual, cf. [1]) is an acceleration of the basic iterative solver which is defined by the following property: given the start iterate $\underline{u}^0$ and the first residual $\underline{r}^0 = \underline{f} - \underline{A}\,\underline{u}^0$, we define the iterates $\underline{u}^m \in \left\{\underline{u}^0 + \underline{B}\,\underline{w} : \underline{w} \in \mathcal{K}^m\right\}$ such that they satisfy the minimization property

$$\|\underline{B}\Big(\underline{f} - \underline{A}\,\underline{u}^m\Big)\| = \min_{\underline{w}\in\mathcal{K}^m} \|\underline{B}\Big(\underline{f} - \underline{A}\big(\underline{u}^0 + \underline{B}\,\underline{w}\big)\Big)\|$$

with respect to the Euclidian norm $\|\cdot\|$. Note that this method is designed for general regular matrices $\underline{A}$ and $\underline{B}$ without requiring symmetry or positive definiteness.

For an efficient computation of the minimal residual, the Krylov basis $\underline{r}^0$, $\underline{A}\,\underline{B}\,\underline{r}^0, \ldots, (\underline{A}\,\underline{B})^{m-1}\underline{r}^0$ is transformed by Givens rotations into $\underline{v}^0, \ldots, \underline{v}^{m-1}$ such that the inner products

$$H = \left(\underline{v}^i \cdot \underline{v}^j\right)_{i,j=0,\ldots,m-1} \in \mathbb{R}^{m,m}$$

build a matrix of Hessenberg form.

In order to restrict the required memory for storing the Krylov basis, the algorithm runs up to some fixed number $m = M$ (typically $M < 100$), and the minimization is restarted again, until the residual norm is smaller than an absolute tolerance $\|\underline{r}^k\| \leq \varepsilon$, or until a reduction $\|\underline{r}^k\| \leq \theta\|\underline{r}^0\|$ by a prescribed reduction factor θ is reached.

In detail, the parallel algorithm is explained by the source code shown in Fig. 3. The solution vector $\underline{u}$ and the correction vector $\underline{c}$ are represented consistently, and the residual vector $\underline{r}$, the Krylov basis vectors $\underline{v}^k$, and a auxiliary vector $\underline{w}$ are represented additively. The parallelization is included in the operations with the stiffness matrix $\underline{A}$ and the preconditioner $\underline{B}$, and in the inner product and the norm computations of additive vectors. All other parts in the algorithm can be performed without any further parallel communication.

In our implementation, the template class `GMRES` is derived from a base class `IterativeSolver` providing the interface for a virtual member function `Solver` and the iteration parameters `eps`, `theta` and `max_iter`. The restart parameter `M` is a template parameter. In general, this algorithm (and other iterative solver such as BiCGstab) can be included directly from suitable template libraries, provided that all vector operations (including inner product and norm evaluation) and the application of the stiffness matrix and the preconditioner can be overloaded with the parallel operations of our programming model.

5 Numerical experiments

We present two 3-dimensional boundary-value problems, both dealing with failure mechanisms of natural slopes. There, the modeled soil is assumed to behave like a clayey silt. The characterizing material parameters are given in Table 1 taken from the triphasic model of Ehlers et al. [6]. Note that in contrast to [6], the value for the intrinsic permeability K_{0S}^S was chosen to 10^{-9} m^2, which is a very large value for realistic soils. This is due to the fact that the current version of the parallel iterative solver is very sensible with respect to small values of K_{0S}^S. This motivates further development on the parallel preconditioner to achieve more robustness. Nevertheless, the obtained results from the parallel simulations show a similar behavior

```
void GMRES::Solver (Vector &u, const Operator& A,
                    const Operator& B, Vector &r)
{
  double d = d_0 = norm(r);
  double epsilon = max(eps,d*theta);
  double H[M+1][M], e[M+1], y[M], cs[M], sn[M];
  vector<Vector> v(M+1,r);
  Vector c(u), w(r);
  for (iter=0; iter < max_iter;) {
    if (d < epsilon) break;
    v[0] = (1/d) * r;
    e[0] = d;
    for (int i=1; i<M; ++i) e[i] = 0;
    int k = 0;
    for (; k<M; ++k) {
      c = B * v[k];
      w = A * c;
      v[k+1] = w;
      for (int i=0; i<=k; ++i) {
        H[i][k] = w * v[i];
        v[k+1] -= H[i][k] * v[i];
      }
      H[k+1][k] = norm(v[k+1]);
      v[k+1] *= (1/H[k+1][k]);
      for (int i=0; i<k; ++i) {
        double giv_r = H[i][k];
        double giv_h = H[i+1][k];
        H[i][k] = sn[i]*giv_r+cs[i]*giv_h;
        H[i+1][k] = cs[i]*giv_r-sn[i]*giv_h;
      }
      double giv_r = H[k][k];
      double giv_h = H[k+1][k];
      double co,si;
      d = sqrt(giv_r*giv_r+giv_h*giv_h);
      if (giv_r > 0) {
        co = giv_h / d;
        si = giv_r / d;
        H[k][k] = d;
      } else {
        co = - giv_h / d;
        si = - giv_r / d;
        H[k][k] = -d;
      }
      H[k+1][k] = 0;
      giv_r = e[k];
      e[k] = si * giv_r;
      e[k+1] = co * giv_r;
      cs[k] = co;
      sn[k] = si;
      ++iter;
      if (abs(e[k+1])<epsilon) {
        k++;
        break;
      }
    }
    for (int i=k-1; i>=0; --i) {
      double s = e[i];
      for (int j=i+1; j<k; ++j)
        s -= H[i][j] * y[j];
      y[i] = s / H[i][i];
    }
    w = 0;
    for (int i=0; i<k; ++i) w += y[i] * v[i];
    c = B * w;
    u += c;
    r -= A * c;
    d = norm(r);
  }
}
```

Fig. 3 C++ source code of the parallel GMRES method based on parallel operators $\underline{A}$, $\underline{B}$, and parallel inner products and norms of vectors

Table 1 Material parameters of the triphasic model

Lamé constants	$\mu^S = 5583$ kN/m^2, $\lambda^S = 8375$ kN/m^2
effective densities	$\rho^{SR} = 2720$ kg/m^3, $\rho^{LR} = 1000$ kg/m^3
gas phase	$\bar{R}^G = 287.17$ J/(kg K), $\theta = 283$ K, $p_0 = 10^5$ N/m^2
solid volume fraction	$n^S_{0S} = 0.54$
gravitation	$g = 9.81$ m/s^2
fluid viscosities	$\mu^{LR} = 10^{-3}$ N s/m^2, $\mu^{GR} = 1.8 \cdot 10^{-5}$ N s/m^2
intrinsic permeability	$K^S_{0S} = 10^{-9}$m^2, $\pi = 1.0$
van Genuchten model	$\alpha_{\text{gen}} = 2 \cdot 10^{-4}$, $h_{\text{gen}} = 1.5$ $j_{\text{gen}} = 2.3$, $\epsilon_{\text{gen}} = 0.5$, $\gamma_{\text{gen}} = 0.333$
residual saturations	$s^L_{\text{res}} = 0.1$, $s^G_{\text{res}} = 0.1$
yield criterion	$\alpha = 1.0740 \cdot 10^{-2}$, $\beta = 0.1196$, $\gamma = 1.555$ $\delta = 1.377 \cdot 10^{-4}$ m^2/kN, $\epsilon = 4.330 \cdot 10^{-6}$ m^2/kN $\kappa = 10.27$ kN/m^2, $m = 0.5935$
plastic potential	$\psi_1 = 1.33$, $\psi_2 = 0.107$
viscoplasticity	$\eta = 500$s, $\sigma_0 = 10.27$ kN/m^2, $r = 1$

concerning the failure mechanisms of the respective slopes in comparison with the results of [6].

In both examples, the coarse mesh is generated with the mesh generator NETGEN [13] which allows for an appropriate mesh grading near material interfaces (where we expect interesting effects and where we need a finer mesh resolution). Then, we distribute the mesh onto the processors by a simple lexicographic ordering of the cell mid-points, which is sufficient for our examples. Note that this load balancing minimizes the number of neighboring processors, but for more complex geometries it is not optimal with respect to the size of the processor interfaces.

The main emphasis in the following discussion is put on the convergence behavior of the chosen spatial discretization and the performance of the parallel code, in particular the presented GMRES solver from the previous section.

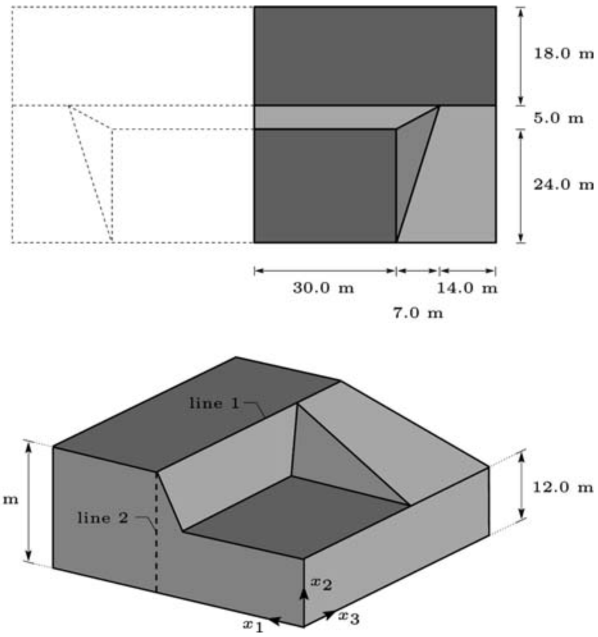

Fig. 4 Excavation problem: sketch of the whole slope (top) and perspective view on half of the excavated slope (bottom)

5.1 Excavation problem

The first example concerns a standard excavation problem carried out at a natural slope shown in Fig. 4: the top figure describes the total geometrical situation after the excavation process has been realized, and the lower figure shows a perspective image of half of the problem (due to the symmetry of the configuration, it is sufficient to consider only half of the geometry in the numerical computations).

It was shown in [6] that the slope under study results in a stable configuration under gravitational load as long as the ground-water table is so low that the pore-liquid does not leak at the slope. If this situation is changed by an increase of the ground-water table, cf. Fig. 5 (top), the failure of the slope is initiated, thus leading to a development of a shearing domain with a certain thickness, cf. Fig. 5 (bottom). Furthermore, the stream lines presented in Fig. 6 show clearly that the pore-liquid

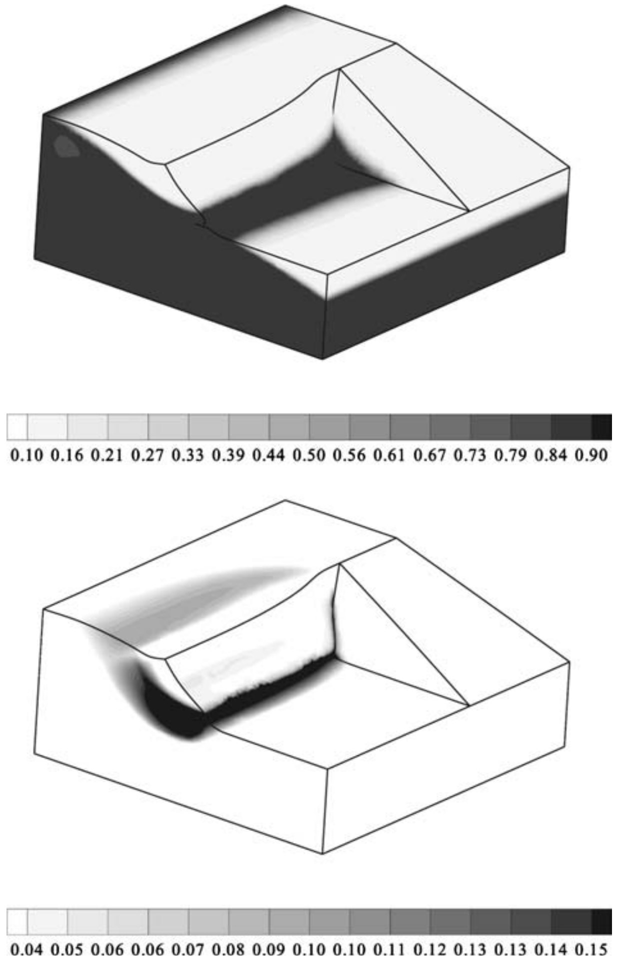

Fig. 5 Excavation problem: liquid saturation s^L [-] (top) and accumulated plastic strains $||\varepsilon_p||$ [-] (bottom) after the excavation process (deformation scaled 3 times)

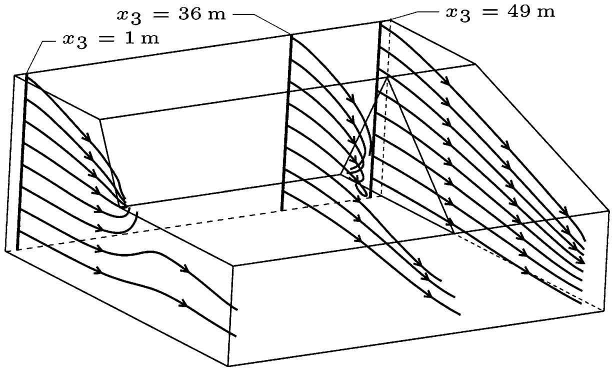

Fig. 6 Excavation problem: stream lines of the pore-liquid flow starting from 3 different lines at $x_3 = 1$ m, $x_3 = 36$ m and $x_3 = 49$ m

leaks at the surface of the slope. Note that this fact is important for a realistic modeling of the underlying problem, as the leaking of the pore-liquid prevents the development of a positive pressure (excess pressure) for p^L at the boundary surface of the slope, which would lead to unrealistic results. The leaking of the pore-fluid is realized within our calculation by a flexible setting of the Dirichlet boundary conditions d^L, i. e., if the capillary pressure p^C falls below a critical value at a certain node on the boundary surface due to an increase of the liquid saturation s^L, cf. Fig. 1, the value for p^L at this node is set to the ambient pressure. At the beginning of the calculation, the values of p^L are not specified at the boundary surface of the slope, since we want to prevent a predefined distribution of the liquid saturation $s^L = s^L(p^L, p^G)$ on this boundary surface. Such a predefined distribution would occur within our model as we already set $p^G = d^G$ (Dirichlet boundary condition) to the ambient gas pressure at the boundary surface of the slope.

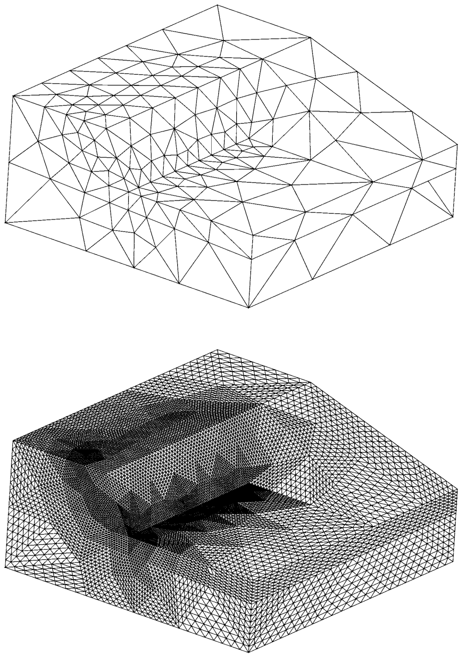

Fig. 7 Excavation problem: initial finite element mesh (top) and final mesh after three uniform refinement steps (bottom)

For a proper discussion of the convergence behavior of the underlying spatial discretization, the described boundary-value problem was calculated on four different FE meshes, where the coarsest mesh, cf. Fig. 7 (top), contains 964 tetrahedra. Starting from this mesh (refinement level 0), three uniform refinements result into the finest spatial discretization (refinement level 3) with altogether 493 568 cells, cf. Fig. 7 (bottom). Using generalized Taylor-Hood elements with quadratic ansatz functions for the displacement $\mathbf{u}_S$ and linear ansatz functions for the pressure terms p^L and p^G (cf. Fig. 2), the convergence behavior depicted in Fig. 8 is obtained. Thereby, the development of the displacements u_1 and u_2

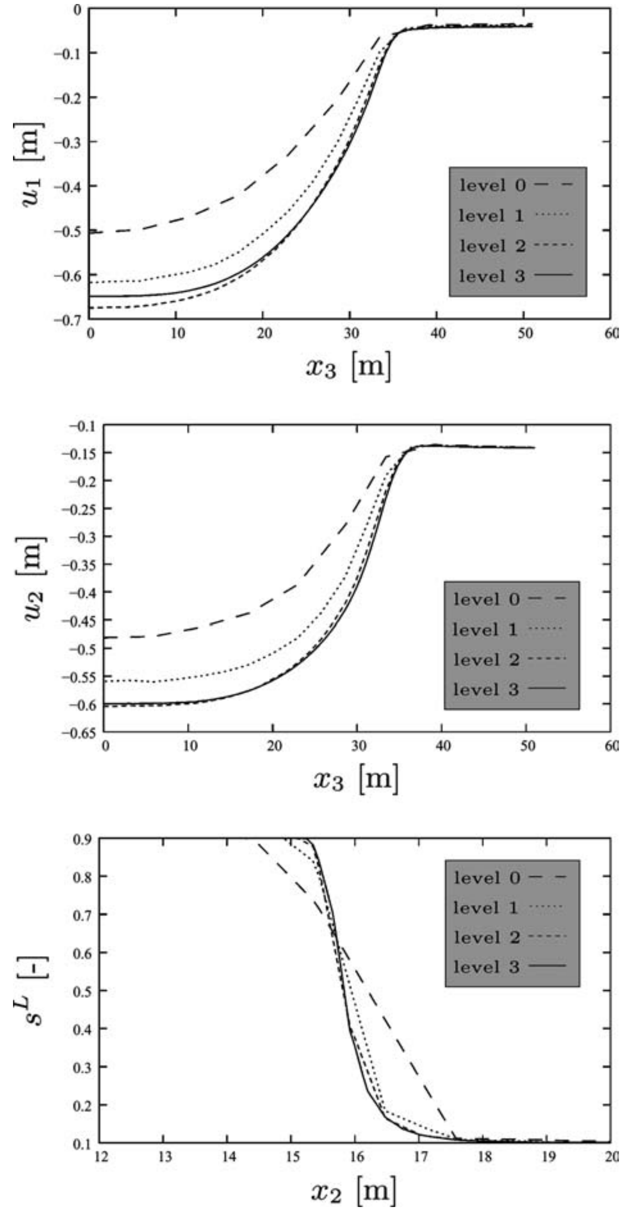

Fig. 8 Excavation problem: convergence behavior; displacement u_1 [m] and u_2 [m] plotted along line 1 and liquid saturation s^L [-] plotted along line 2

Table 2 Excavation problem: computational expense for the full simulation of 270 time steps on different refinement levels

	cells	DOFs	Gauss points	time [h]
level 0	964	5 436	14 460	00:09:50
level 1	7 712	38 185	115 680	00:48:34
level 2	61 696	285 893	925 440	06:14:47
level 3	493 568	2 211 941	7 403 520	95:42:45

along line 1 ($x_1 = 29\,\mathrm{m}$, $x_2 = 21\,\mathrm{m}$; cf. Fig. 4) and the liquid saturation s^L along line 2 ($x_1 = 29\,\mathrm{m}$, $x_3 = 0\,\mathrm{m}$; cf. Fig. 4) are shown exemplarily for the four different meshes. In these three figures, an excellent convergence behavior can be identified for the regularly refined FE meshes.

Note that the number of cells increases by the factor 8 with every uniform refinement step, i. e. each tetrahedron is subdivided into 8 tetrahedra per refinement (cf. Table 2). Furthermore, for a stable numerical integration, we have to use a quadrature formula of fifth order [14], which leads to 15 integration points per element, and, thus, to a huge number of internal variables.

In order to be able to compare the computing times of the different FE meshes, all calculations were carried out on 24 processors of a Linux cluster (Xeon 2.4 GHz). In Table 2, we observe that from level 0 to 1 and from level 1 to 2 the computing time increases by a factor smaller than 8, which indicates that we have a large parallel overhead for less than 100 000 degrees of freedom. For large problems, 24 processors can be used efficiently, and we observe that the computing time of the level 3 problem is c. approx. 15.3 times longer than those of the level 2 problem, which is nearly twice as big as the increase of the degrees of freedom (DOFs). Thus, we have a loss of efficency which corresponds well to the increase of Newton steps and GMRES steps (cf. Table 3 and Table 4), and it is therefore not caused by some overhead in the parallel communication.

Note that the scaling properties can be improved by smaller time steps on the finest mesh. For simplicity of the comparison of the results, we fix a common time series for all refinement levels, which results in a moderate increase of the required number of Newton steps, cf. Table 3. Thus, the behavior of the GMRES method within the level 3 computation is associated with too large time steps, which result in some more iterations for the solution of the linear systems, cf. Table 4. On the other hand, this does not affect the accuracy of our results, since our numerical experiments indicate that the spatial error is dominating and that a further reduction of the time steps is not changing the results substantially.

Table 3 Excavation problem: average number of Newton iterations per time increment and average computing time for one implicit Euler step on different meshes

refinement level	0	1	2	3
Newton iterations (avg.)	2.5	3.1	3.3	4.9
computing time [s] (avg.)	1.9	10.5	82.5	1142.9

Table 4 Excavation problem: performance of the GMRES solver (average number of iterations per Newton step, average computing time for the solution of one linear problem, and total number of linear problems for the full simulation of 270 time steps)

refinement level	0	1	2	3
GMRES iterations (avg.)	26.5	43.9	79.8	233.7
computing time [s] (avg.)	0.2	2.0	16.3	193.3
number of linear problems	676	810	882	1280

5.2 Slope failure

In the second example, a slope failure problem is discussed, which is caused by an extreme rainfall event. Therein, the slope under study is assumed to consist of three different soil strata, cf. Fig. 9. The soil stratum at the bottom of the model (stratum 3) should behave purely elastic, whereas the other two strata can show plastic material behavior if the elastic range of the soil is exceeded. Furthermore, the small soil stratum in the middle (stratum 2) obtains an intrinsic permeability coefficient K^S_{0S}, which is lower by the factor 1000 than the intrinsic permeability coefficient of the other two soil strata, cf. Table 1. Therefore, pore-liquid within the partially saturated soil is prevented from streaming through this "impermeable" stratum.

The slope under study is in a stable situation before the rainfall starts with an infiltration rate (negative efflux) of $\bar{v}^L = -10^{-5}\,\mathrm{m}^3/(\mathrm{m}^2\mathrm{s})$ corresponding to $36\,\mathrm{l}/(\mathrm{m}^2\mathrm{h})$. In this example, the heavy rainfall event is

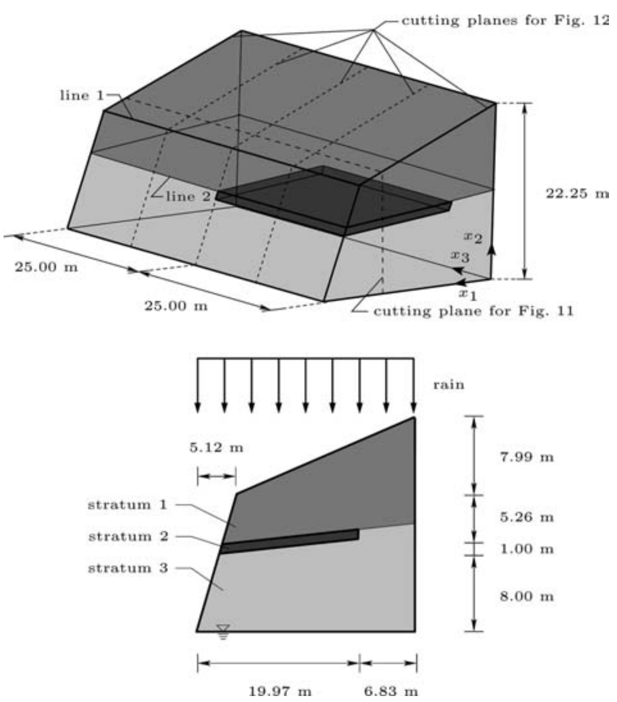

Fig. 9 Slope failure: perspective view (top) and side view (bottom) on the slope

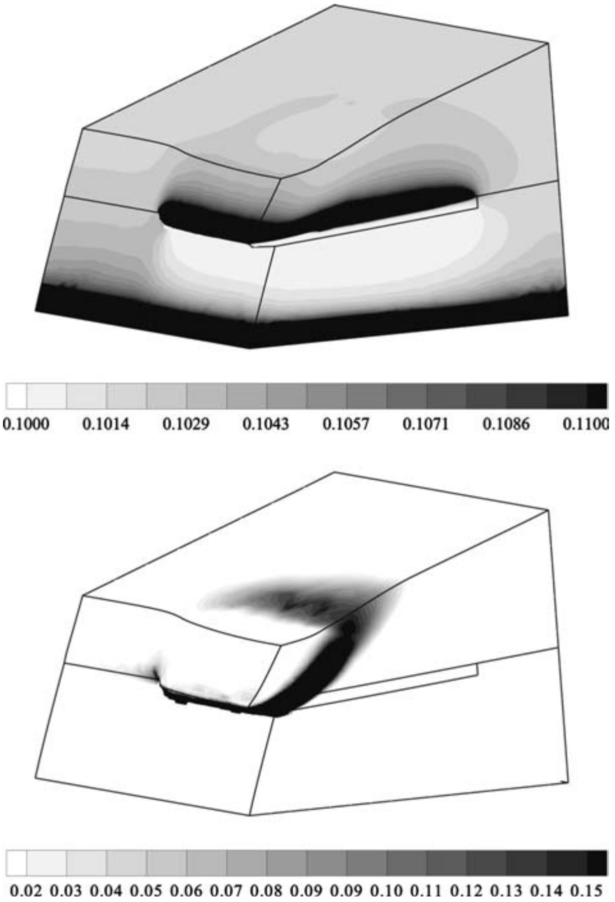

Fig. 10 Slope failure: liquid saturation s^L [-] (top) and accumulated plastic strains $||\boldsymbol{\varepsilon}_p||$ [-] (bottom) due to a heavy rainfall (scaled 3 times)

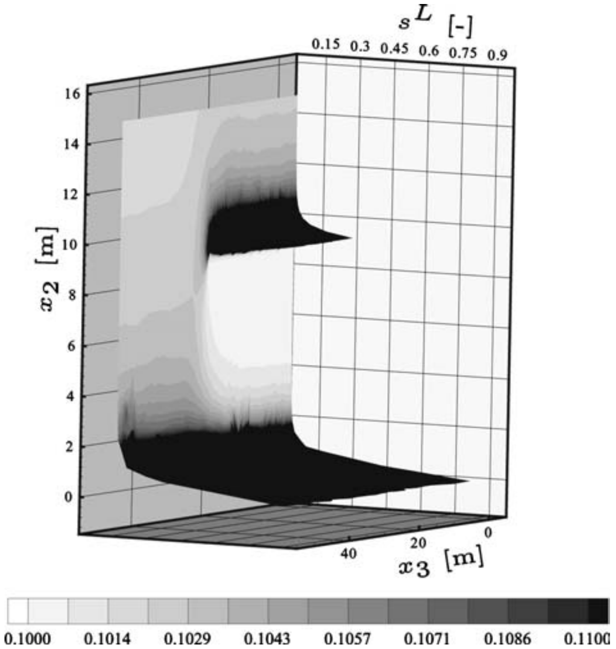

Fig. 11 Slope failure: distribution of the pore-liquid saturation s^L [-] in the cutting plane shown in Fig. 9 (top)

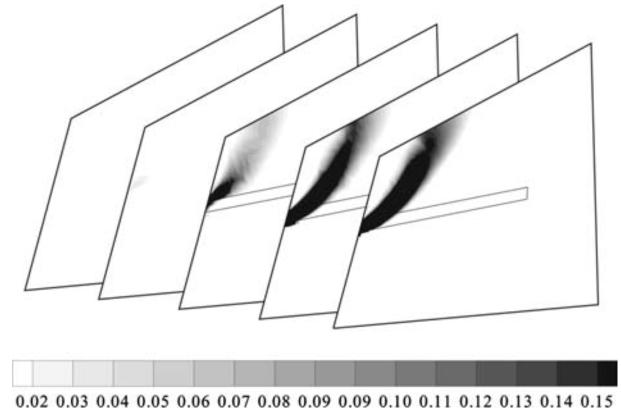

Fig. 12 Slope failure: development of the accumulated plastic strains $||\boldsymbol{\varepsilon}_p||$ [-] in the cutting planes shown in Fig. 9 (top)

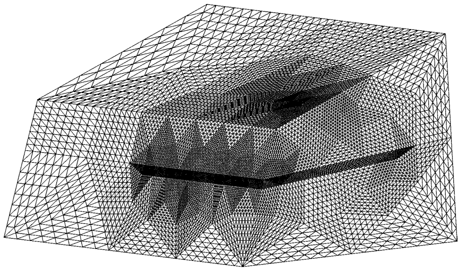

Fig. 13 Slope failure: FE mesh (refinement level 3)

assumed to last for 2.5 hours, which gives altogether 90 mm rainfall. Due to this heavy rainfall, pore-liquid is accumulated at the top of the impermeable soil stratum, cf. Fig. 10 (top) and 11. Thus, as a result of buoyancy, failure of this part of the slope is initiated, whereas the backward part of the slope remains in its initial stable state, cf. Fig. 10 (bottom) and 12.

The calculation of this problem is again carried out on four different, uniformly refined FE meshes, where the finest mesh (level 3) is shown in Fig. 13. Note that it is essential for obtaining reasonable results, that already the coarse mesh has a sufficient mesh resolution near the impermeable stratum. Like in the previous example, the convergence behavior of the spatial discretization shows very good results. In order to illustrate this statement, the distribution of the effective liquid pressure p^L is plotted along line 1 ($x_1 = 21.68$ m, $x_2 = 14.26$ m; cf. Fig. 9) and the development of the displacement u_1 and the accumulated plastic strains $||\boldsymbol{\varepsilon}_p||$ are plotted along line 2 ($x_1 = 22$ m, $x_2 = 9$ m; cf. Fig. 9).

For a reasonable comparison of the computing times of the four different calculations, all computations were again carried out on 24 processors of the same Linux cluster from the previous example. In Table 5, the number of cells and integration points, the degrees of freedom, and the total computing time for the four FE meshes are shown. Therein, it can be observed that the four FE meshes result in similar problem sizes like for the excavation problem. In contrast to the previous

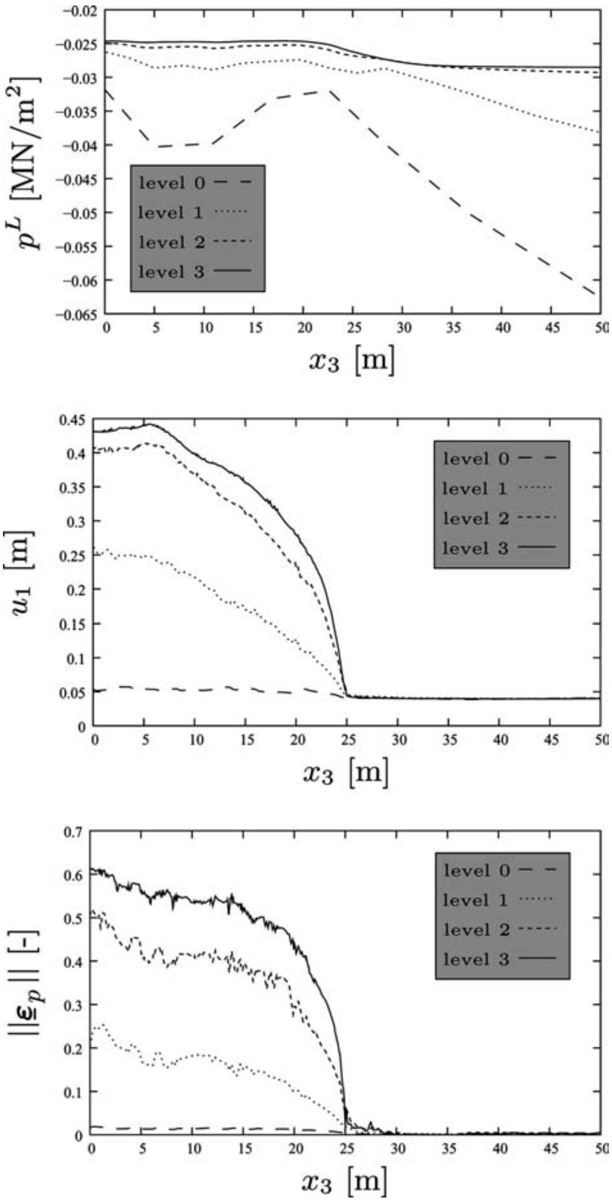

Fig. 14 Slope failure: convergence behavior; effective liquid pressure p^L [MN/m^2] plotted along line 1 and displacement u_1 [m] and accumulated plastic strains $||\varepsilon_p||$ [-] plotted along line 2

Table 5 Slope failure: computational expense for the full simulation of 800 time steps on different refinement levels

	Cells	DOFs	Gauss points	time [h]
level 0	1 000	5 339	15 000	00:16:11
level 1	8 000	38 381	120 000	03:55:02
level 2	64 000	291 589	960 000	56:50:30
level 3	512 000	2 274 501	7 680 000	633:08:52

example, the increase of computing time between the different levels is almost constant for the first three levels, i. e., there is approximately a factor of 14.5 between the computing times of the levels 0 and 1 and the levels 1 and 2. Although the total computing time of about 27.5 days for the solution on the finest mesh is rather long, the increase of computing time with respect to the calculation on the level 2 mesh is only around 11.1, which is smaller than for the coarser meshes.

Table 6 Slope failure: average number of Newton iterations per time increment and average computing time for one implicit Euler step on different meshes

refinement level	0	1	2	3
Newton iterations (avg.)	1.6	2.2	3.5	3.7
computing time [s] (avg.)	1.2	17.3	253.8	2847.6

Table 7 Slope failure: performance of the GMRES solver (average number of iterations per Newton step, average computing time for the solution of one linear problem, total number of linear problems for the full simulation of 800 time steps)

refinement level	0	1	2	3
GMRES iterations (avg.)	29.3	64.3	127.7	265.3
computing time [s] (avg.)	1.4	6.5	64.5	702.9
number of linear problems	1277	1756	2836	2987

In this example, the time step sizes were chosen in an appropriate manner for all four FE meshes such that no time step has to be repeated because of a non-converging Newton method, cf. Table 6. Furthermore, Table 7 shows a moderate increase of average computing time for the solution of the linear systems.

6 Conclusion

In the present contribution, we showed that the GMRES method together with a suitable preconditioner is a powerful tool for solving the large linear systems occurring in the parallel computation of coupled multiphasic problems. In particular, we briefly presented the underlying triphasic model for the description of partially saturated soil as the basis for the numerical experiments. The material model is treated numerically within the framework of the finite element method. Based on our parallel programming model, we discussed the usage of a parallel preconditioned GMRES method.

The efficiency of the presented parallel solver was shown in two demanding numerical simulations concerning typical geotechnical problems, where the convergence behavior of the spatial discretization as well as the performance of the parallel code was studied in detail. In both numerical experiments, the spatial discretization with generalized Taylor-Hood elements led to an excellent convergence behavior during a regular refinement of the FE meshes. The application of the presented GMRES method turned out to be well suited for the solution of the saddle point problems resulting from the linearization of the presented triphasic model. Nevertheless, further development of the iterative linear solver is necessary for the application to more general multiphasic material models, for the improvement of the overall performance, and for the scalability to a larger number of processors.

References

1. Barrett R, Berry M, Chan T, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, van der Vorst H (1994) Templates for the solution of linear systems. SIAM, Philadelphia, ftp://ftp.netlib.org/templates/templates.ps
2. Bastian P (1996) Parallele adaptive Mehrgitterverfahren. Teubner Skripten zur Numerik. Teubner, Stuttgart
3. de Boer R (2000) Theory of Porous Media. Springer-Verlag, Berlin
4. Ehlers W (1995) A single-surface yield function for geomaterials. Archive of Applied Mech 65:246–259
5. Ehlers W (2002) Foundations of multiphasic and porous materials. In Ehlers W, Bluhm J (eds) Porous Media: Theory, Experiments and Numerical Applications. Springer-Verlag, Berlin pp. 3–86
6. Ehlers W, Graf T, Ammann M (2004) Deformation and localization analysis in partially saturated soil. Computer Methods in Applied Mech Eng 193:2885–2910
7. Helmig R (1997) Multiphase Flow and Transport Processes in the Subsurface. Springer-Verlag, Berlin
8. Helmig R, Huber R (1998) Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media. Advances in Water Resources, 21:697–711
9. Lewis RW, Schrefler BA (1998) The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media, 2 nd Edition. Wiley, Chichester
10. Mahnkopf D (2000) Lokalisierung fluidgesättigter poröser Festkörper bei finiten elastoplastischen Deformationen. Dissertation, Bericht Nr. II-5 aus dem Institut für Mechanik (Bauwesen), Universität Stuttgart
11. Müllerschön H (2000) Spannungs-Verzerrungsverhalten granularer Materialien am Beispiel von Berliner Sand. Dissertation, Bericht Nr. II-6 aus dem Institut für Mechanik (Bauwesen), Universität Stuttgart
12. Perzyna P (1966) Fundamental problems in viscoplasticity. Advances in Applied Mech 9:243–377
13. Schöberl J (1997) NETGEN – an advancing front 2d/3d-mesh generator based on abstract rules. Computing and Visualization in Science 1:40–52
14. Stroud AH (1971) Approximate Calculation of Multiple Integrals. Prentice-Hall, New Jersey
15. van Genuchten MT (1980) A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Science Society of America J. 44:892–898
16. Wieners C, Ammann M, Diebels S, Ehlers W (2002) Parallel 3-d simulations for porous media models in soil mechanics. Comput Mech 29:73–87
17. Wieners C, Ammann M, Ehlers W (2004) Distributed Point Objects: A new concept for parallel finite elements applied to a geomechanical problem. Future Generation Computer Systems, in press