HAL
open science

High Performance Computing linear algorithms for two-phase flow in porous media

Robert Eymard, Cindy Guichard, Roland Masson

To cite this version:

Robert Eymard, Cindy Guichard, Roland Masson. High Performance Computing linear algorithms for two-phase flow in porous media. FVCA7 - The International Symposium of Finite Volumes for Complex Applications VII, Jun 2014, Berlin, Germany. 10.1007/978-3-319-05591-6_55 . hal-00958131

HAL Id: hal-00958131

https://hal.science/hal-00958131

Submitted on 11 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High Performance Computing linear algorithms for two-phase flow in porous media

Robert Eymard[*], Cindy Guichard[†], Roland Masson[‡]

March 11, 2014

Abstract

We focus here on the difficult problem of linear solving, when considering implicit scheme for two-phase flow simulation in porous media. Indeed, this scheme leads to ill-conditioned linear systems, due to the different behaviors of the pressure unknown (which follows a diffusion equation) and the saturation unknown (mainly advected by the total volumic flow). This difficulty is enhanced by the parallel computing techniques, which reduce the choice of the possible preconditioners. We first present the framework of this study, and then we discuss different algorithms for linear solving. Finally, numerical results show the performances of these algorithms.

1 Introduction

We consider the flow of two immiscible compressible phases, the water phase (denoted w) and the gas phase (denoted g), in porous media; each phase is only composed of one component. In order to characterize the mathematical coupling of diffusion and advection, we consider the case where the capillary pressure effects can be neglected in front of the high level of pressure gradients imposed by the production and injection wells. The mass conservation equations are therefore the following,

$$\phi \; \partial_t(\; \rho_\alpha(P) \; S_\alpha \;) \; + \; \mathrm{div}\,(\; \rho_\alpha(P) \; \mathbf{V}_\alpha \;) \; = \; Q_\alpha, \qquad \alpha = w, g \; , \tag{1}$$

together with the generalized Darcy law

$$\mathbf{V}_\alpha \; = \; - \frac{k_{r\alpha}(S_\alpha)}{\mu_\alpha(P)} \; \Lambda \; (\; \nabla P \; - \; \rho_\alpha(P) \; \mathbf{g} \;) \, , \qquad \alpha = w, g \; . \tag{2}$$

In equations (1) and (2), the main unknowns are the pressure P and one saturation, for example S_w, since the phase saturations are linked by $S_w + S_g = 1$. Additionally, ϕ is the porosity, Λ is the absolute permeability tensor (these values only depending on the rock material), $\mathbf{g}$ is the gravity acceleration, and, for each phase $\alpha = w, g$, ρ_α represents the bulk density, $k_{r\alpha}$ is the relative permeability (nonnegative increasing function with respect to S_α), μ_α is the viscosity and Q_α is the source term that represents the contribution of the wells. The ratio $\frac{k_{r\alpha}}{\mu_\alpha}$ is called the mobility of the phase α. These equations are considered in a time-space domain $(0, t_\mathrm{f}) \times \Omega$, where Ω is a polygonal open bounded and connected subset of $\mathbb{R}^3$, and $t_\mathrm{f} > 0$ is the time duration of the simulation. Finally, these equations are considered together with homogeneous Neumann conditions at the boundary of

[*]Laboratoire d'Analyse et de Mathématiques Appliquées, CNRS, UPEM, UPEC, 5 boulevard Descartes, Champs-sur-Marne 77454 Marne-la-Vallée Cedex 2, France, `Robert.Eymard@u-pem.fr`

[†]Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, `guichard@ljll.math.upmc.fr`

[‡]University Nice Sophia Antipolis, and team COFFEE, INRIA Sophia Antipolis Méditerranée, France, `roland.masson@unice.fr`

the domain Ω, and initial conditions on the pressure and on the saturation.

The approximation of the solution to (1-2) in the industrial framework with large time and space scales, requires High Performance Computing techniques. This implies to handle the difficult problem of solving the linear systems which arise from fully coupled schemes and domain decomposition, using multi-threading algorithms: these schemes happen to be the only ones used for the approximation of (1-2) in the industrial framework. For this purpose, we consider here the use of PETSc [5] together with external preconditioners libraries like MUMPS [4] and HYPRE [2]. Note that other packages, like DUNE [1], are available. In the example of a gas storage case [10], a very good scalability has been observed using Boomer AMG [2] as a preconditioner on the full system, although AMG is usually not adapted to solve the full system but only the pressure elliptic block. But for more general situations of two-phase flow (such as the case considered in the numerical example of this paper), this strategy fails. This has led to the development of efficient Combinative-AMG preconditioners [11], combining typically an AMG preconditioner on the pressure block with an ILU preconditioner on the full system. This paper focuses on an alternative algorithm, based on the PETSc environment, for the resolution of the linear systems issued from a fully implicit scheme for the approximation of (1-2). Its main advantages are the following:

1. It makes a bridge between sequential and fully implicit schemes.
2. It leads to the sequential use of robust solvers suited for the nature of each unknown.

This paper is organized as follows. In Section 2, we present a discretization scheme and its parallel implementation. We then discuss in Section 3 the fix-point methods used for the approximation of the solution to the nonlinear systems of equations (3-4). Some numerical results, in Section 4, illustrate our method.

2 Discretization and parallel implementation

In order to study the algorithms for solving the linear systems in a parallel framework, we have extended to the two-phase flow model a recent work (see [6]) done for a linear parabolic equation. For the implementation details, we use below the same notations as [6], thus we focus on the specific points regarding the discretization of two-phase flow. Hence, the continuous model (1-2) is discretized using an Euler fully implicit method in time, and the VAG scheme (Vertex Approximate Gradient scheme introduced in [7]) in space with up-winding of the mobilities according to the sign of the Darcy fluxes. We emphasize that the VAG scheme is a symmetric scheme based on a hybrid formulation, both in terms of vertices and cells unknowns, but in the resulting linear system the cell unknowns are algebraically eliminated without any fill-in. The VAG scheme involves linear fluxes between a cell and its vertices and its implementation matches with that of a standard Multi-Points Flux Approximation. We refer to [8, 9] for details on the VAG scheme for multiphase flow in porous media in the case of a sequential implementation.

Parallel discretization. We consider a mesh of the domain Ω (the elements of the mesh are called cells in the following). As in [6, Sub-section 2.1], we denote the set of processes by $\mathcal{P}$, and we consider a partition of the mesh. For a given process $p \in \mathcal{P}$, we denote by $\mathcal{M}^p$ the set of its own cells (in practice selected by applying the Metis package [3]) and by $\overline{\mathcal{M}}^p$ the set of its overlapped cells which is defined as the set of all cells sharing a vertex with $\mathcal{M}^p$. Then we can define the overlapping decomposition of the set of vertices as follows:

$$\overline{\mathcal{V}}^p = \bigcup_{K \in \overline{\mathcal{M}}^p} \mathcal{V}_K, \quad p \in \mathcal{P},$$

where $\mathcal{V}_K$ is the set of the vertices of a given cell K. Finally, the set of the own vertices of a process $p \in \mathcal{P}$, denoted $\mathcal{V}^p$, is obtained by the application of a rule detailed in [6, Sub-section 2.1]. We then

discretize the continuous equations (1-2) on each process p, for each phase $\alpha = w, g$, by writing

$$\frac{|\mathbf{s}|}{\delta t^{(n)}} \left(\rho_\alpha(P_{\mathbf{s}}^{(n+1)}) S_{\alpha,\mathbf{s}}^{(n+1)} - \rho_\alpha(P_{\mathbf{s}}^{(n)}) S_{\alpha,\mathbf{s}}^{(n)} \right) \; - \sum_{K \in \mathcal{M}_{\mathbf{s}}} M_{\alpha,K\mathbf{s}}^{\mathrm{up},(*)} \; V_{\alpha,K\to\mathbf{s}}^{(n+1)} = 0 \qquad \forall \mathbf{s} \in \mathcal{V}^p, \tag{3a}$$

$$\frac{|K|}{\delta t^{(n)}} \left(\rho_\alpha(P_K^{(n+1)}) S_{\alpha,K}^{(n+1)} - \rho_\alpha(P_K^{(n)}) S_{\alpha,\mathbf{s}}^{(n)} \right) \; + \sum_{\mathbf{s} \in \mathcal{V}_K} M_{\alpha,K\mathbf{s}}^{\mathrm{up},(*)} \; V_{\alpha,K\to\mathbf{s}}^{(n+1)} = |K| Q_{\alpha,K}^{(n+1)} \qquad \forall K \in \overline{\mathcal{M}}^p, \tag{3b}$$

together with the Darcy fluxes (see [8, Sub-sections 3.1.2 and 3.2])

$$V_{\alpha,K\to\mathbf{s}}^{(n+1)} = \sum_{\mathbf{s}' \in \mathcal{V}_K} a_{K,\mathbf{s}}^{\mathbf{s}'} \left(\; P_K^{(n+1)} - P_{\mathbf{s}'}^{(n+1)} + \rho_\alpha(P_K^{(n)}) \mathbf{g} \cdot (x_K - x_{\mathbf{s}'}) \; \right). \tag{4}$$

In (3-4), we use the following notations: $\mathcal{M}_{\mathbf{s}}$ is the set of cells K such that $\mathbf{s} \in \mathcal{V}_K$, $|K|$ (resp. $|\mathbf{s}|$) is the porous volume associated to a cell K (resp. to a vertex $\mathbf{s}$), computed from a redistribution of the total porous volume of the space domain with respect to the mesh and the rock type properties [8, 9], $x_K \in \mathbb{R}^3$ (resp. $x_{\mathbf{s}} \in \mathbb{R}^3$) denotes the coordinates of the center of the cell K (resp. of the vertex $\mathbf{s}$). Note that the VAG scheme construction does not use the geometry of the control volumes $\mathbf{s} \in \mathcal{V}$ and $K \in \mathcal{M}$ but only their volumes. For $n \in \mathbb{N}$, $\delta t^{(n)} = t^{(n+1)} - t^{(n)}$ is the time step between times $t^{(n+1)}$ and $t^{(n)}$. For any control volume I ($I = K$ or $I = \mathbf{s}$), $P_I^{(n)}$ (resp. $S_{\alpha,I}^{(n)}$) is an approximation of P (resp. of S_α) in I at time $t^{(n)}$. $a_{K,\mathbf{s}}^{\mathbf{s}'}$ is computed with respect to the mesh and the permeability tensor Λ [8, 9]. $M_{\alpha,K\mathbf{s}}^{\mathrm{up},(*)}$ denotes the upstream mobility of the phase α and is defined by

$$M_{\alpha,K\mathbf{s}}^{\mathrm{up},(*)} = \left(\rho_\alpha(P_{K\mathbf{s}}^{(n)}) \frac{k_{r\alpha}(S_{\alpha,K\mathbf{s}}^{(*)})}{\mu_\alpha(P_{K\mathbf{s}}^{(n)})} \right),$$

where $K\mathbf{s}$ denotes the cell K if $V_{\alpha,K\to\mathbf{s}}^{(n+1)} \geqslant 0$, or the vertex $\mathbf{s}$ otherwise. The upper index $(*)$ stands for (n) (ImPES scheme) or $(n+1)$ (fully implicit scheme); this point is reviewed in Section 3. Finally, $Q_{\alpha,K}^{(n+1)}$ is the possible source term if any well is open through cell K.
As usual, no special numerical treatment is needed for taking into account the homogeneous Neumann boundary conditions (see [8, 9]). The set of equations (3-4) leads to a system of nonlinear equations at each time step. This system is solved by a fix-point algorithm based on the Newton-Raphson method (up to a possible under-relaxation in order to prevent from nonconvergence behaviors). Thus, the unknowns of the resulting discrete problem are, on each process $p \in \mathcal{P}$, the variations of $(P_I^{(n+1)})_{I \in \overline{\mathcal{V}}^p \cup \overline{\mathcal{M}}^p}$ and $(S_{w,I}^{(n+1)})_{I \in \overline{\mathcal{V}}^p \cup \overline{\mathcal{M}}^p}$ between two fix-point iterations. As in [6, Sub-section 2.4], their values are obtained through the construction of rectangular linear systems on each process $p \in \mathcal{P}$ and, as mentioned above, a consequence of equations (3-4) is that the cell unknowns can be eliminated by a Schur complement without fill-in, in order to reduce the linear system to the vertices unknowns. Thanks to our general definition of the overlap, the assembling step may be performed locally on each process without communication.

3 Fix-point methods

This section presents the fix-point method used in our parallel implementation, and its implementation thanks to open-source libraries. The variation of the vertices unknowns between two fix-point iterations is denoted as follows, omitting the time superscript $(n+1)$ and the Newton iteration index for the sake of clarity,

$$U^p \;=\; \left(\; (\triangle P_{\mathbf{s}})_{\mathbf{s} \in \overline{\mathcal{V}}^p} \; , \; (\triangle S_{w,\mathbf{s}})_{\mathbf{s} \in \overline{\mathcal{V}}^p} \; \right), \quad \forall p \in \mathcal{P}.$$

If the fix-point method were exactly the Newton method, then U^p would be the solution of a linear system over all processes p, under the form

$$\begin{pmatrix} A^p_{PP} & A^p_{PS} \\ A^p_{SP} & A^p_{SS} \end{pmatrix} U^p = \begin{pmatrix} B^p_P \\ B^p_S \end{pmatrix}, \tag{5}$$

where, for $u1, u2 = P, S$, the vectors right-hand-side B^p_{u1} belong to $\mathbb{R}^{\mathcal{V}^p}$ and the sub-matrices $A^p_{u1,u2}$ belong to $\mathbb{R}^{\mathcal{V}^p} \otimes \mathbb{R}^{\overline{\mathcal{V}^p}}$.

Then, we define the diagonal blocks, of size 2×2, by

$$D^{\mathbf{s}} = \begin{pmatrix} A^p_{PP}(\mathbf{s},\mathbf{s}) & A^p_{PS}(\mathbf{s},\mathbf{s}) \\ A^p_{SP}(\mathbf{s},\mathbf{s}) & A^p_{SS}(\mathbf{s},\mathbf{s}) \end{pmatrix}, \quad \forall \mathbf{s} \in \mathcal{V}^p, \quad \forall p \in \mathcal{P},$$

where $A^p_{u1,u2}(\mathbf{s},\mathbf{s})$ is the term associated to the equation on $\mathbf{s}$ and the unknown on $\mathbf{s}$ of the sub-matrix $A^p_{u1,u2}$. We then left-multiply the system (5) by the square matrix $[\mathrm{diag}(D^{\mathbf{s}}, \mathbf{s} \in \mathcal{V}^p)]^{-1} \in \mathbb{R}^{(\mathcal{V}^p)^2} \otimes \mathbb{R}^{(\mathcal{V}^p)^2}$. We then get the following linear system

$$\begin{pmatrix} \widehat{A}^p_{PP} & \widehat{A}^p_{PS} \\ \widehat{A}^p_{SP} & \widehat{A}^p_{SS} \end{pmatrix} U^p = \begin{pmatrix} \widehat{B}^p_P \\ \widehat{B}^p_S \end{pmatrix}, \tag{6}$$

where the new right-hand-side and sub-matrices have the same dimension as in (5) but now satisfy

$$\widehat{A}^p_{PP}(\mathbf{s},\mathbf{s}) = 1, \ \widehat{A}^p_{PS}(\mathbf{s},\mathbf{s}) = 0, \ \widehat{A}^p_{SP}(\mathbf{s},\mathbf{s}) = 0, \ \widehat{A}^p_{SS}(\mathbf{s},\mathbf{s}) = 1, \tag{7}$$

for all vertices $\mathbf{s} \in \mathcal{V}^p$, and for any process $p \in \mathcal{P}$.
Our implementation of the sequential fix-point scheme allows the choice between the three following schemes (in all cases, the solution at own and ghost cells is computed locally on each process $p \in \mathcal{P}$ by Schur complement).

1. The **ImPES scheme**, for Implicit in Pressure and Explicit in Saturation, is obtained by taking $(*) = (n)$ in (3). This means that the linear system, under the form (6), is such that $\widehat{A}^p_{PS} = 0$ and $\widehat{A}^p_{SS}(\mathbf{s},\mathbf{s}) = 1$ if $\mathbf{s} \in \mathcal{V}^p$, and $\widehat{A}^p_{SS}(\cdot,\cdot) = 0$ otherwise. Hence the resolution of (6) first implies a full resolution on all vertices of the mesh for the equation

$$\widehat{A}^p_{PP}(\triangle P_{\mathbf{s}})_{\mathbf{s} \in \overline{\mathcal{V}^p}} = \widehat{B}_p. \tag{8}$$

This sub-system is transferred line by line to PETSc which provides the solution vector on pressure variations $(\triangle P_{\mathbf{s}})_{\mathbf{s} \in \mathcal{V}^p}$ at own vertices for each process $p \in \mathcal{P}$. The pressure variations at own and ghost vertices $(\triangle P_{\mathbf{s}})_{\mathbf{s} \in \overline{\mathcal{V}^p}}$, $p \in \mathcal{P}$ is obtained by a synchronization of the vertices. Then the saturation variations at own vertices are immediately obtained from (6) and the saturation variations at ghost vertices are obtained by a second synchronization of the vertices. Note that the matrix A^p_{PP} has the main properties of a finite element matrix for a diffusion problem. Therefore, in PETSc, it is standard to use Algebraic Multi-Grid preconditioning for the resolution of these linear systems. Unfortunately, the ImPES scheme implies a limit on the time step, for standard stability reasons, which is generally not compatible with industrial requirements.

2. The **fully implicit scheme** is obtained by taking $(*) = (n+1)$ in (3) (which implies that the matrices $\widehat{A}^p_{PS}$ do no longer vanish) and then to solve the coupled linear system (6) with a unique linear solver issued from the PETSc library. In this case, PETSc provides the solution on both pressure and saturation variations at own vertices for each process $p \in \mathcal{P}$. The variations at the unknowns at the ghost vertices are then obtained by synchronization. As discussed in the introduction of this paper, this strategy implies the implementation of Combinative preconditioners [11]. The following sequential scheme proposes a related but simpler to implement approach.

3. The **sequential scheme** consists in only approximately solving the linear systems (6), thanks to the following algorithm, based on the combination of adapted preconditioners on both the pressure and saturation blocks of the full system. For any process $p \in \mathcal{P}$, we consider the Gauss-Seidel type method,

$$\forall \mathbf{s} \in \overline{\mathcal{V}^p}, \quad \triangle S_{w,\mathbf{s}}^{\{0\}} \; = \; 0, \tag{9a}$$

$$\widehat{A}_{PP}^p(\triangle P_{\mathbf{s}}^{\{k+1\}})_{\mathbf{s}\in\overline{\mathcal{V}^p}} \; + \; \widehat{A}_{PS}^p(\triangle S_{w,\mathbf{s}}^{\{k\}})_{\mathbf{s}\in\overline{\mathcal{V}^p}} \; = \; \widehat{B}_p \tag{9b}$$

$$\widehat{A}_{SP}^p(\triangle P_{\mathbf{s}}^{\{k+1\}})_{\mathbf{s}\in\overline{\mathcal{V}^p}} \; + \; \widehat{A}_{SS}(\triangle S_{w,\mathbf{s}}^{\{k+1\}})_{\mathbf{s}\in\overline{\mathcal{V}^p}} \; = \; \widehat{B}_s, \tag{9c}$$

where in (9), the upper index $k = 0, \ldots, \mathbf{M}$ is corresponding to the sequential scheme iterations. Hence, if the integer $\mathbf{M}$ is large enough, this algorithm leads to a fix point method for solving the linear systems (6) issued from Newton's method applied to the fully implicit scheme. Nevertheless, we take in practice $\mathbf{M} \leq 5$ to ensure a reasonable wall clock time (denoted by WCT in the tables).
The resolution of (9), at each scheme iteration $k+1$, requires to solve the first parallel linear system (9b) which has the same skeleton as (8) issued from the ImPES scheme. Its resolution by PETSc provides the solution vector on pressure variations $(\triangle P_{\mathbf{s}}^{\{k+1\}})_{\mathbf{s}\in\mathcal{V}^p}$ at own vertices for each process $p \in \mathcal{P}$. The pressure variations at own and ghost vertices is still obtained by a synchronization of the vertices. Then the saturation variations is obtained by solving (9c). This implies the assembling of a second parallel linear system, once again solved by PETSc. This provides the saturation variations at own vertices for each process $p \in \mathcal{P}$, and a synchronization step concludes the iteration of the method. In terms of cost of communication, this sequential method is close to that resulting from the fully implicit scheme, if the total (sum on k) number of iterations of its two successive solvers is close to the number of iterations of the unique linear solver used for the fully implicit method. To achieve this, a very efficient strategy is then to specify the residual tolerance $\varepsilon(k)$ of these two resolutions with respect to the value of k. We have implemented the relation $\varepsilon(k+1) = \varepsilon(k)^2$. This leads to a very small number of linear solver iterations for the first values of k.

4 Numerical results

We now consider a two-phase flow on a 3D "five-spot pattern", i.e. with 4 vertical injection wells (at each corner of the domain) and 1 vertical production well (at the center of the domain). The geometry and the permeability field of the test case is illustrated by Figure 1 (*right side*). The cluster used is composed of 32 processors Intel® Xeon® CPU E5-4620 with frequency 2.20GHz. Three successive meshes have been built (with resp. $101 \times 101 \times 12$, $201 \times 201 \times 12$ and $401 \times 401 \times 12$ cells), and we focus on the beginning of the simulation where the pressure and saturation variations are the highest. Referring to Section 3: firstly, the ImPES scheme is not efficient in front of the highest variations of the unknowns; secondly, we did not find any efficient preconditioner for the coupled system in the PETSc framework, involving a strong motivation for exploring the sequential algorithm which is simpler to implement than a Combinative-AMG preconditioner.
Let us first comment the results obtained with 32-processors runs. We present in Figure 1 (*left side*) the residual in function of the Newton iterations, for different values of $\mathbf{M}$ (see Section 3). This figure shows that the convergence rate is only linear for $\mathbf{M} = 1$, and is more and more quadratic as $\mathbf{M}$ increases, since the scheme becomes closer to the pure Newton method. Table 1 exhibits the wall clock times, the total number of Newton iterations, and of solver iterations for the pressure resolution and the saturation resolution. For the ill-conditioned pressure block, the preconditioner is 1 V-cycle of boomer AMG of HYPRE with Gauss Seidel relaxation, whereas for the much better conditionned saturation block, we selected the Jacobi preconditioner. The final residual reduction specified for each linear resolution is $\varepsilon(\mathbf{M}) = 10^{-5}$. For these numerical tests we imposed a reduction of Newton residual equal to 10^{-6}. This criterion has been selected in order to impose a high precision in the nonlinear resolution, hence indicating the robustness of the algorithm with respect to severe convergence requirements. We observe that an optimum, both with respect to the wall clock time and the number of Newton iterations, is obtained with $\mathbf{M} = 3$.

	6 time steps on the $101 \times 101 \times 12$ mesh					7 time steps on the $201 \times 201 \times 12$ mesh					4 time steps on the $401 \times 401 \times 12$ mesh				
WCT init (s)	8					27					110				
M	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
WCT Newton (s)	46	38	41	61	72	353	337	309	391	449	1831	1638	1631	1905	2252
♯Newton	57	42	38	38	38	85	53	44	45	44	65	40	33	31	32
♯iter/P	739	737	717	788	809	699	585	558	638	654	722	577	520	540	593
♯iter/S	143	161	168	202	239	255	250	238	275	308	239	228	210	210	245

Table 1: Results on the three meshes with 32 processors: "WCT init" denotes the wall clock time for initialization operations (in particular including mesh reading and partitioning), "WCT Newton" is the wall clock time for nonlinear iterations with total number ♯Newton, ♯iter/P (resp. /S) total number of pressure (resp. saturation) linear iterations.

♯proc.	1	2	4	8	16	32
total WCT (s)	3214	1743	1039	655	460	336
♯Newton	45	45	45	45	45	44
♯iter/P	569	560	570	562	562	558
♯iter/S	243	244	243	244	243	238

Table 2: Results for 7 time steps on the $201 \times 201 \times 12$ mesh and $\mathbf{M} = 3$ (the total Wall Clock Time includes the initialization and Newton iteration times).

The scalability results presented in Table 2 are similar to those of [10]. The parallel efficiency is reduced from 16 to 32 processors due to a too small number of unknowns per processor in the AMG preconditioner for this problem size (see [10]). They also show a very good stability of the linear algorithms with respect to the increase of the number of processors.

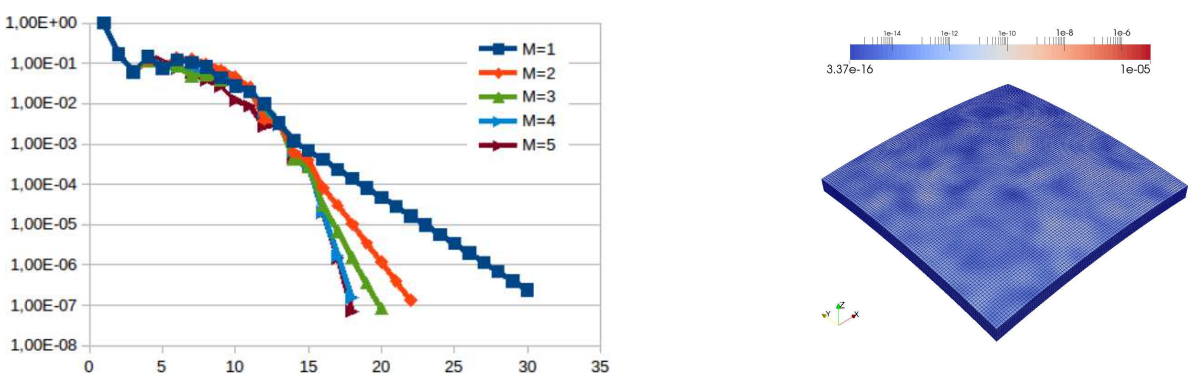

Figure 1: *Left:* residual with respect to the Newton iteration at the first time step of $401 \times 401 \times 12$ mesh. *Right:* random log normal heterogeneous permeability tensor on the same mesh.

References

[1] DUNE Distributed and Unified Numerics Environment. http://www.dune-project.org/

[2] Hypre Parallel High Performance Preconditioners. http://acts.nersc.gov/hypre/

[3] Metis Serial Graph Partitioning and Fill-reducing Matrix Ordering. http://glaros.dtc.umn.edu/gkhome/views/metis

[4] MUMPS MUltifrontal Massively Parallel sparse direct Solver. http://mumps.enseeiht.fr/

[5] PETSc Portable, Extensible Toolkit for Scientific Computation. http://www.mcs.anl.gov/petsc

[6] Dalissier, E., Guichard, C., Have, P., Masson, R., Yang, C.: ComPASS : a tool for distributed parallel finite volume discretizations on general unstructured polyhedral meshes. ESAIM: Proceedings **43**, 147–163 (2013)

[7] Eymard, R., Guichard, C., Herbin, R.: Small-stencil 3d schemes for diffusive flows in porous media. ESAIM: Mathematical Modelling and Numerical Analysis **46**, 265–290 (2012)

[8] Eymard, R., Guichard, C., Herbin, R., Masson, R.: Vertex-centred Discretization of Multiphase Compositional Darcy flows on General Meshes. Computational Geosciences **16**(4), 987–1005 (2012)

[9] Eymard, R., Guichard, C., Herbin, R., Masson, R.: Vertex centred Discretization of Two-Phase Darcy flows on General Meshes. ESAIM: Proceedings **35**, 59–78 (2012)

[10] Eymard, R., Guichard, C., Masson, R.: Simulation of two-phase Darcy flow using the VAG scheme on parallel architecture. poster at MoMaS Multiphase Seminar Days (oct. 2013), available online: http://math.unice.fr/ massonr/films/Compass.pdf

[11] Scheichl, R., Masson, R., Johannes, W.: Decoupling and block preconditioning for sedimentary basin simulations. Computational Geosciences **7**, 295–318 (2003)