Large Scale Simulations of Continuum Models using Parallel Geometric Multigrid Methods

S. Reiter, A. Vogel, G. Wittum

published in

NIC Symposium 2018
K. Binder, M. Müller, A. Trautmann (Editors)

Forschungszentrum Jülich GmbH,
John von Neumann Institute for Computing (NIC),
Schriften des Forschungszentrums Jülich, NIC Series, Vol. 49,
ISBN 978-3-95806-285-6, pp. 377.
http://hdl.handle.net/2128/17544

Large Scale Simulations of Continuum Models using Parallel Geometric Multigrid Methods

Sebastian Reiter[1], **Andreas Vogel**[1], **and Gabriel Wittum**[1,2]

[1] Goethe Center for Scientific Computing, Goethe-Universität Frankfurt,
Kettenhofweg 139, 60325 Frankfurt (M.), Germany
E-mail: {sebastian.reiter,andreas.vogel,wittum}@gcsc.uni-frankfurt.de

[2] Extreme Computing Research Center, King Abdullah University of Science and Technology,
Thuwal 23955-6900, Kingdom of Saudi Arabia
E-mail: gabriel.wittum@kaust.edu.sa

Two simulations of continuum models on the JUQUEEN supercomputer are presented: permeation of substances into the human skin as an application from computational biology and density driven subsurface flow from the field of computational geosciences. Employing geometric multigrid methods and carefully designed refinement and distribution strategies, we demonstrate the applicability and efficiency of our simulation approaches in massively parallel computations on up to 262,144 processing entities.

1 Introduction

A broad variety of physical phenomena can be modelled in terms of partial differential equations (PDE) on complex spatial domains. In order to compute the solution of the arising equations, well-established grid-based methods such as finite element[1,2] or finite volume[3] methods are commonly employed. These methods discretise the considered spatial domain using a structured or unstructured computational mesh that is used to represent the computed solution on a grid element basis. However, a large number of grid elements and thereby degrees of freedom may be required to adequately capture the specific properties of such problems and sufficiently approximate the sought solution. Especially for large and complex physical domains and for problems with high accuracy requirements such settings quickly become too large to be addressed via a single computer or small cluster. This makes them a prime subject for the application of massively parallel computing architectures.

To this end, we develop the simulation software ***UG*** 4^4 for the efficient numerical solution of PDE based problems on massively parallel computers. The aim of ***UG*** 4 is to provide software components that can be applied to a variety of problems and are at the same time highly scalable to a large number of processes. In this report, we present two studies from the diverse field of possible applications: density driven flow in a groundwater reservoir as an application from the computational geosciences and drug diffusion through the upper human skin as a problem setting in computational biology. In the following, we briefly survey the models used and the corresponding discretisations in Sec. 2 and the parallel solution strategies in Sec. 3. Computational results as well as required adjustments for HPC computing are subsequently presented for the subsurface flow in Sec. 4 and the human skin application in Sec. 5 with a focus on grid generation and scalability.

2 Continuum Models and their Discretisation

Flow and transport phenomena in a spatial domain Ω are commonly modelled assuming an underlying physical conservation law: in an isolated system, certain physical quantities (e.g., mass, concentrations, energy, ...) are preserved over time. This is mathematically expressed via the general idea that in an arbitrary volume $B \subset \Omega$ the time-derivative of the quantity, say c, is zero unless there exists some flux $\mathbf{F}(c)$ of the quantity over the domain boundaries ∂B or some external source or sink f. This reads

$$\partial_t \int_B c = -\int_{\partial B} \mathbf{F} \cdot \mathbf{n} + \int_B f, \qquad \text{for all } B \subset \Omega,$$

with the outer normal $\mathbf{n}$. Using Gauss' theorem and assuming a sufficiently smooth solution c this can be stated in divergence form as the continuity equation

$$\partial_t c = -\nabla \cdot \mathbf{F} + f, \qquad \text{for all } \mathbf{x} \in \Omega.$$

The spatial domain for the human skin model[5] is given by the stratum corneum, the uppermost layer of the skin, that serves as a barrier protecting us from environmental influences. However, the stratum corneum is still penetrable by substances. In our model, the diffusing substance c is a conserved quantity and no external sources are present, $f \equiv 0$. The difficulty of this problem is due to the fact that the model for the diffusive flux $\mathbf{F}(c) = -\mathbf{D}\nabla c$ possesses a jump of orders of magnitude in the diffusion tensor $\mathbf{D}$ between the corneocytes (skin cells) and the surrounding lipid channels (cf. Fig. 1b).

In the density driven subsurface flow application, we consider the flow of brine (solution of salt in water) in a porous medium[6]. Two conservation laws are used to provide a model for the unknown brine mass fraction ω and the pressure p: the conservation of mass for the fluid-phase as a whole and the conservation of mass of brine. The complete model reads as

$$\begin{aligned} \partial_t(\phi\rho) + \nabla \cdot (\rho\mathbf{q}) &= q, \quad \text{in } \Omega, \\ \partial_t(\phi\rho\omega) + \nabla \cdot (\rho\omega\mathbf{q} - \rho\mathbf{D}\nabla\omega) &= q_s, \quad \text{in } \Omega, \\ \mathbf{q} &= -\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g}), \end{aligned} \tag{1}$$

with the porosity ϕ, permeability $\mathbf{K}$, viscosity μ, fluid density ρ, gravity $\mathbf{g}$, diffusion-dispersion tensor $\mathbf{D}$, external sources q, q_s, and Darcy velocity $\mathbf{q}$.

In order to address these equations numerically, we employ the vertex-centred finite volume method[3]. The general idea is to partition the physical domain Ω with simple elements (tetrahedra, hexahedra, prisms, ...) such that the spatial geometry is resolved. The conservation property for the discrete solution is ensured via a set of equations on grid-based control volumes $B \subset \Omega$. These control volumes are constructed employing a dual technique such that each grid vertex is surrounded by one control volume (cf. Fig. 1a). This results in a large set of equations (one for each vertex) that must be solved.

3 Solver and Parallelisation

The mesh-based discretisation of the above models leads to a large system of equations. In order to use the available computing resources, we parallelise the discrete problem setting employing a domain decomposition approach, i.e., each processing entity is assigned

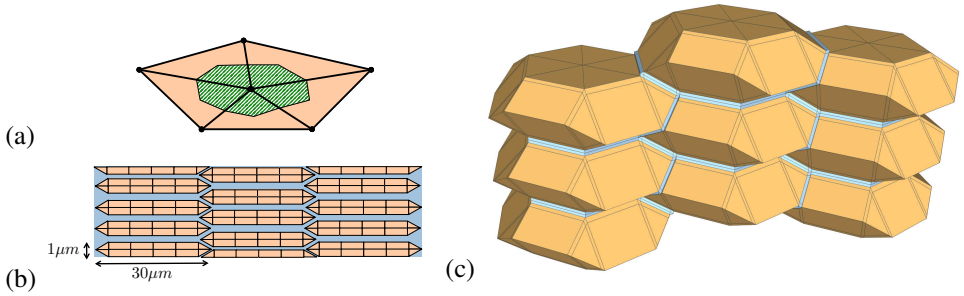

Figure 1. **(a)** Dual construction of finite volumes (green) on an unstructured grid (orange). **(b)** Simplified human skin model showing corneocytes (skin cells, orange) and surrounding lipid channels (blue). **(c)** Finite element mesh of tetrakaidecahedral cells with interconnecting lipid channels (10 times vertical exaggeration).

a part of the considered domain Ω. Once this partitioning is achieved, the setup of the discrete equations is inherently parallelised since each process can simply assemble its contribution to the equations without any communication. Given an equal distribution of the underlying grid among all processes, a perfect weak scaling can thus be expected for the assembling process. The parallelisation of the linear equation solver is more involved. Here, we employ the geometric multigrid method on a distributed grid hierarchy[7]. The multigrid method is well known for its linear complexity w.r.t. to the number of discrete unknowns[8] and therefore nicely suited for weak scaling applications.

The geometric multigrid method typically operates on grid hierarchies which are constructed by repeated refinement of a coarse grid. With each refinement a new level is created in the grid hierarchy. In the case of hexahedral meshes the number of elements grows by a factor of 8 with each refinement step and it is thus clear that meshes quickly grow too large to be handled by a single processor. At the same time coarse meshes are too small to be spread over large numbers of processes.

To circumvent this problem, we use the approach of process hierarchies[7]: The coarse grid on level 0 of the hierarchy is distributed among a small number of processes only. We then repeatedly perform refinement steps until sufficiently many elements are created in order to allow for the redistribution of the top level to a larger subset of processes. Refinement and redistribution are repeated until all available processes are occupied.

While coarse meshes accurately represent the topology of the problem domain, the geometry may often be simplified. During refinement it is thus important to position new vertices in such a way that the approximation of the original geometry is improved with each refinement. To this end, ***UG*** 4 features so called *refinement projectors* which position new vertices (inner and boundary vertices) according to predefined rules. An example for the case of the refinement of 2d layered domains is given in Fig. 2 where a coarse grid is constructed from a set of layer boundaries. In addition to the coarse grid, we also construct a projector that places newly created vertices in such a way that the original boundaries are preserved in higher levels of the hierarchy. By positioning the inner vertices using the same projector, a good element quality is preserved during refinement. This technique is used for grid generation in Sec. 4.

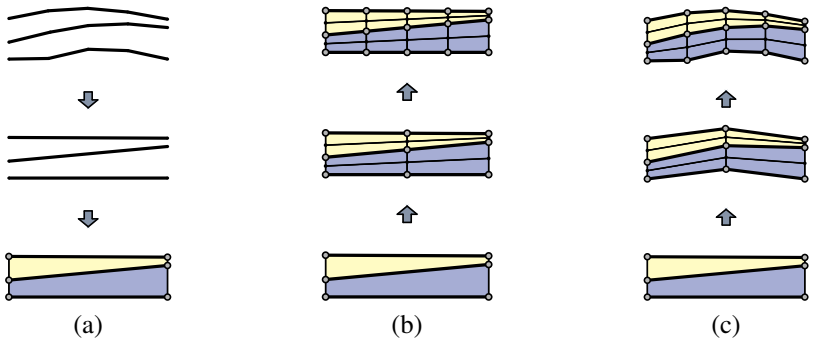

Figure 2. **(a)** Coarse grid generation from detailed input data. **(b)** Standard linear refinement with bad approximation of original domain. **(c)** Projected refinement with good approximation and good element qualities.

4 Density Driven Subsurface Flow

Large scale simulations of groundwater flow on realistic domains are a relevant research field considering necessary risk evaluation involved in the long term storage of hazardous and contaminated waste. The sheer size of the regarded domains and the required accuracy to capture important properties of the considered problems pose high demands both on suitable hardware and numerical solver efficiency. Massively parallel computers provide the computing and memory resources required to address these problems. At the same time, multigrid methods provide nearly optimal complexity and scaling behaviour[7]. However, the application to realistic problem settings is still challenging due to the size and complexity of the underlying domains. Highly detailed grids resembling those domains have to be generated and managed in a distributed way such that they span the whole set of available processes in a given simulation setup.

To this end, we developed an automated meshing algorithm which generates distributed hierarchies of semi-structured prism meshes resembling layered soil formations with large horizontal extensions[9]. Starting from a stack of 2d raster data representing the height values of interfaces between different soil layers, first a coarse prism grid is extracted which adheres to the interfaces and pinchouts between different layers. Together with the coarse grid we also generate a special projector used to place new vertices according to the underlying detailed raster data during refinement. Between refinement steps we distribute the grid using the hierarchical approach[7]. The resulting meshing approach interweaves refinement, projection, and redistribution in a highly efficient manner. The more often the mesh is refined, the better it approximates the underlying domain (cf. Fig. 3, left). The algorithm was realised with the meshing software *ProMesh*[10,11].

To explore the behaviour of the system under different boundary conditions, we varied a previous study[9] for a benchmark problem based on the WIPP model[12]: we consider density driven flow in a large aquifer which spans an area of 10000 km^2 containing 6 different soil layers. Individual layers may be as thin as $1m$ or vanish completely locally and permeabilities differ up to the order of 10^5 between the layers. We apply zero Dirichlet boundary conditions for the pressure and brine mass fraction on the top surface of the domain and a brine mass fraction at saturation level at the bottom surface. In addition

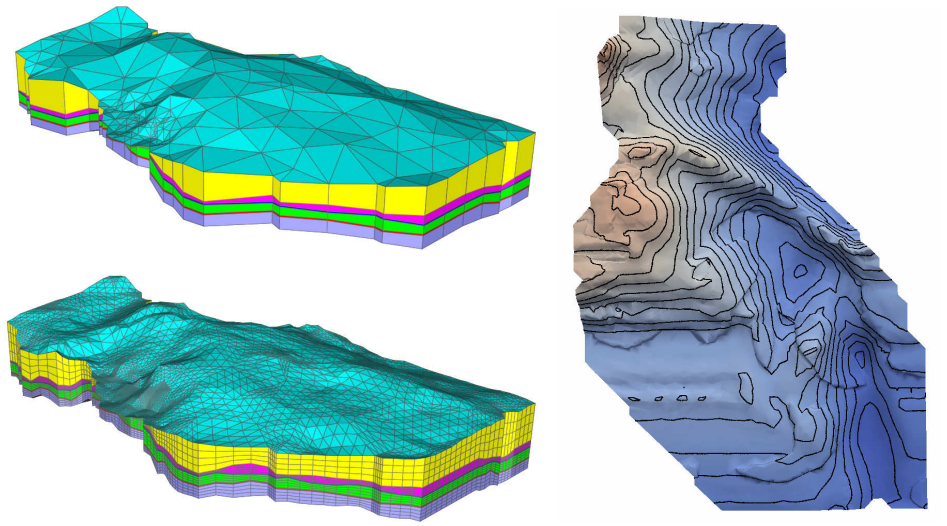

Figure 3. **Left**: Two refinement stages of the WIPP geometry. The more refinements the higher the fidelity of the mesh and the approximation of the underlying domain. All geometries are scaled by a factor of 50 along the vertical axis. **Right**: Isosurface for a concentration of 1% of brine after 6000 years. Contour lines and colouring refer to the elevation of the isosurface.

to our previous study[9] we also employ outflow boundary conditions on the sides of the domain. In the new setup, concentration may flow out of the physical domain depending on the flow field at the boundary, resulting in a realistic treatment of the concentration at the domain boundaries over time.

Following the outlined approach, we performed runs of this benchmark problem over a simulated timespan of 6000 years on 4096 cores using the maximum walltime of 24 h on JUQUEEN. We observe that the proposed meshing and parallel multigrid hierarchy creation are working very well for the considered application. The nonlinear equations were solved using a Newton method with a BiCGStab solver preconditioned by a parallel V-cycle geometric multigrid with 5 ILU pre- and post-smoothing steps and a parallel ILU preconditioned BiCGStab base solver. The parallel base solver operated on 32 processes. Higher levels were distributed onto 1024 processes and the uppermost levels to 4096 processes. An isosurface of the computed brine mass fraction after 6000 years of this benchmark problem is depicted in Fig. 3, right.

5 Drug Diffusion through the Human Skin

Another prime example for the usage of our methods is the high resolution simulation of drug permeation through the skin[5]. The key motivation behind this project is to develop a deep understanding of such transport processes and to develop computer based studies for medical substances. This report extends a previous study[13] from 32,768 to 262,144 processes.

A good approximation for the cells of the stratum corneum (the corneocytes) can be

achieved by tetrakaidecahedral elements (cf. Fig. 1c) that allow for a realistic dense packing[14], similar to what can be observed in reality. We resolve the individual tetrakaidecahedral elements by hexahedral, prism, and tetrahedral elements. Each tetrakaidecahedral element also contains a thin layer of elements which represent a part of the surrounding lipid layer. Special care has been taken to avoid obtuse angles in the geometry. To this end, additional elements have been introduced in the tetrakaidecahedral mesh[13]. Such a tetrakaidecahedral cell can then be used as a building block to construct large cell compounds by tightly stacking the cells in different space directions (cf. Fig. 1c). Meshing was performed with the software *ProMesh*[10,11].

Starting from a coarse grid we construct a distributed multigrid hierarchy through interweaved parallel refinement and redistribution. Each refinement step creates a new level in the multigrid hierarchy. To reduce anisotropies in the elements by each refinement step, we use a special anisotropic refinement scheme where we only refine those edges which are longer than a given threshold[15]. After each refinement we halve the threshold before attempting a new refinement. Elements which are not refined in a certain step are simply copied to the next level of the hierarchy. For the given geometry we perform 5 anisotropic refinements followed by an arbitrary number of isotropic refinements. The resulting mesh features mostly isotropic elements in the higher levels.

A special solver setup[15] is used: A multigrid preconditioned BiCGStab solver with ILU smoothing operates on the lower anisotropic levels. Those levels only span a small part of the available processes. This solver serves as a base solver for a multigrid preconditioned BiCGStab solver with a highly scalable Jacobi smoother on higher levels. This multigrid solver spans the whole range of available processes. The number of iterations was fixed to 10 for each run to allow for a better comparability of the timings. The averaged relative defect reduction in the different runs varied between 0.32 and 0.38, where the maximum was reached in the run with 4,096 processes.

The geometric multigrid method provided by ***UG*** 4 exhibits nearly optimal weak scalability[7]. However, due to the unstructuredness of the underlying geometry, load balancing is harder for the problem at hand. Due to these slight load imbalances the efficiency of the considered problem is also slightly deteriorating. As a measure for the load imbalance between different processes we define the distribution quality q_l of a level l of the hierarchy as

$$q_l := \frac{n_l^{\text{total}} - n_l^{\text{max}}}{n_l^{\text{max}} \cdot (P_l - 1)},$$

where $P_l > 1$ is the number of processes of the given process-hierarchy on level l, n_l^p is the number of elements in level l on process p, and $n_l^{\text{total}} := \sum_{p=1}^{P_l} n_l^p$ and $n_l^{\text{max}} := \max_{p=1,\ldots,P_l} n_l^p$. For $P_l = 1$ we set $q_l := 1$. A distribution quality equal to 1 means that all processes contain an equal share of elements, whereas $q_l = 0$ means that all elements are located on one process only ($P_l > 1$). The distribution qualities for each level are depicted in Tab. 1.

To judge whether our algorithm works efficiently on the new geometry layout, we performed a scaling study for the computation of the equilibrium state of the skin model using a 10^3 times higher diffusion coefficient in the lipid layer than in the corneocytes. The computations have been performed on the JUQUEEN supercomputer using 16 processes per node with up to 16,384 nodes working concurrently and results are shown in Tab. 2.

PE	lv-0	lv-1	lv-2	lv-3	lv-4	lv-5	lv-6	lv-7	lv-8	lv-9	lv-10
8	1	1	1	0.96	0.97	0.99	-	-	-	-	-
64	1	1	1	0.59	0.81	0.89	0.88	-	-	-	-
512	1	1	1	0.71	0.96	0.86	0.85	0.84	-	-	-
4,096	1	1	1	0.71	0.96	0.85	0.84	0.82	0.80	-	-
32,768	1	1	1	0.71	0.96	0.87	0.89	0.85	0.84	0.82	-
262,144	1	1	1	0.71	0.96	0.87	0.89	0.63	0.62	0.89	0.87

Table 1. Distribution qualities for each level of the multigrid hierarchy for the different runs.

PE	Levels	DoFs	T_{ass}	T_{init}	T_{sol}	$T_{\text{sol}}^{\text{DoF}}$
8	5	374,223	5.6	26.6	54.91	$1.17e^{-3}$
64	6	2,807,351	6.7	26.16	45.91	$1.05e^{-3}$
512	7	21,703,869	7.88	28.45	47.92	$1.13e^{-3}$
4,096	8	170,592,761	8.4	28.94	50.84	$1.22e^{-3}$
32,768	9	1,352,552,433	8.97	29.8	51.93	$1.26e^{-3}$
262,144	10	10,771,587,041	8.79	30.67	52.14	$1.27e^{-3}$

Table 2. Each line corresponds to an individual run. Recorded are the number of processes (PE), the number of levels (Levels), the number of unknowns (DoFs), the run times of assembly (T_{ass}), solver initialisation (T_{init}), and solving (T_{sol}) in seconds. $T_{\text{sol}}^{\text{DoF}}$ represents the cpu-seconds required by the solver per degree of freedom to solve the whole problem.

Since we are interested in the runtime efficiency on large process numbers, we would usually perform a weak scaling study for the problem at hand. However, due to the mixed element types contained in the grid, a weak scaling study can not be realised exactly. E.g., between 8 PE and 64 PE the number of processes grows by a factor of 8 whereas the number of DoFs only grows by a factor of 7.5. In runs with higher element counts the element growth factor is closer to 8 within each run (7.96 between runs 5 and 6). We thus also compare the average cpu-seconds (elapsed time multiplied by number of processes) spent per degree of freedom in each run,

$$T_{\text{sol}}^{\text{DoF}} := \frac{T_{\text{sol}} \cdot \text{PE}}{\text{DoF}}.$$

As shown in Tab. 2, the time spent per degree of freedom only increases slightly in the latter runs. The reason is the load imbalance between processes as shown in Tab. 1. Overall, the setup scales very well to 262,144 processes and 10^{10} degrees of freedom. The deterioration of assembling times T_{ass} is directly correlated to the load imbalance of the top level of the multigrid hierarchy in each run. On the other hand T_{init} and T_{sol} include the time spent on lower levels of the hierarchy, too. Here, the latter runs benefit from a higher degree of parallelisation in the lower levels compared to the first run. Work on the parallel partitioners would further improve the load balance and overall performance and is content of our current research. Nevertheless, the current efficiency is clearly sufficient to perform efficient large scale simulations on massively parallel systems.

Acknowledgements

This work has been funded by the DFG Priority Program 1648 *Software for Exascale Computing* (SPPEXA) in the project *Exasolvers* (WI 1037/24-2) and by the *German Ministry of Economics and Technology (BMWi)* (02E11476B). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (`http://www.gauss-centre.eu`) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUQUEEN at Jülich Supercomputing Centre (JSC).

References

1. S. C. Brenner and R. Scott, *The mathematical theory of finite element methods*, Springer, 1994.
2. D. Braess, *Finite Elemente*, Springer, 2003.
3. R. Bank and D. Rose, *Some error estimates for the box method*, SIAM Journal of Numerical Analysis, **24**, no. 4, 777–787, 1987.
4. A. Vogel, S. Reiter, M. Rupp, A. Nägel, and G. Wittum, *UG 4: A novel flexible software system for simulating PDE based models on high performance computers*, Comp. Vis. Sci., **16**, no. 4, 165–179, 2013.
5. A. Nägel, M. Heisig, and G. Wittum, *A comparison of two-and three-dimensional models for the simulation of the permeability of human stratum corneum*, European Journal of Pharmaceutics and Biopharmaceutics, **72**, no. 2, 332–338, 2009.
6. J. Bear, *Dynamics of fluids in porous media*, New York: Dover Publications, 1972.
7. S. Reiter, A. Vogel, I. Heppner, M. Rupp, and G. Wittum, *A massively parallel geometric multigrid solver on hierarchically distributed grids*, Comp. Vis. Sci., **16**, no. 4, 151–164, 2013.
8. W. Hackbusch, *Multi-grid methods and applications*, vol. 4, Springer, 1985.
9. S. Reiter, D. Logashenko, A. Vogel, and G. Wittum, *Mesh generation for thin layered domains and its application to parallel multigrid simulation of groundwater flow*, Comp. Vis. Sci., submitted.
10. `http://www.promesh3d.com`.
11. S. Reiter, *Effiziente Algorithmen und Datenstrukturen für die Realisierung von adaptiven, hierarchischen Gittern auf massiv parallelen Systemen*, PhD thesis, Universität Frankfurt am Main, 2014.
12. T. Corbet and P. Knupp, *The role of regional groundwater flow in the hydrogeology of the Culebra member of the Rustler formation at the Waste Isolation Pilot Plant (WIPP), southeastern New Mexico*, Tech. Rep., University of North Texas Libraries, 1996.
13. S. Reiter, A. Nägel, A. Vogel, and G. Wittum, *Massively parallel multigrid for the simulation of skin permeation on anisotropic tetrakaidecahedral cell geometries*, High Performance Computing in Science and Engineering 17., 2017.
14. T. D. Allen and C. S. Potten, *Significance of cell shape in tissue architecture*, Nature, **264**, no. 5586, 545–547, Dec 1976.
15. S. Reiter, A. Vogel, A. Nägel, and G. Wittum, *A massively parallel multigrid method with level dependent smoothers for problems with high anisotropies*, High Performance Computing in Science and Engineering 16., 667–675, 2016.