

Final response to the open discussion of W. He et al., A parallelization scheme to simulate reactive transport in the subsurface environment with OGS#iPhreeqc, Geosci. Model Dev. Discuss., 8, 2369-2402, 2015

Dear Editor, dear Referees,

We would like to thank you for your detailed comments and constructive suggestions. Based on these reviews, we overworked the whole manuscript and included the following new contents:

- a new string-based coupling to reduce the computational overhead of the interface
- a more complex example (uranium leaching) to test the parallel performance of our approach
- the according new (updated) results, figures (1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13) and tables (1, 4, 5)
- a new section “code availability” at the end of the revised manuscript
- supplementary material: PHREEQC script for the Engesgaard benchmark
- supplementary material: model description and simulation results of the new example

We reply to these comments individually below and will also address them where possible in the revised version (marked-up version, line numbers mentioned below refer to this version). Reviewer comments are reproduced in **bold**, with our explanations in *italic*.

Wenkui He

On behalf of all authors

Response to anonymous Referee #1

	This paper describes a coupling between OGS software and iPHREEQC software, for reactive transport numerical modeling. OGS deals with flow, transport and heat transfer, whereas PHREEQC deals with geochemistry. The coupling is of type SNIA. Does it mean that the time discretization scheme is explicit?
1	<i>Within OGS we can apply both implicit and explicit time discretization schemes for flow, transport and heat transfer. The time discretization for the coupling scheme itself is explicit. The SNIA approach is applied for operator splitting, so no iterations are made between mass transport and geochemical reaction. Consequently, adequate small time step sizes are needed in order to reduce the operator splitting errors. It is worth mentioning that the SNIA approach is sufficient here, as there are no strong feedbacks of reactions on flow and mass transport in the given simulation examples. For applications involving strong feedbacks, we can implement a sequential iterative approach accordingly. We added this information into our revised version (page 6: 14; page 8:4). Additional information: We can also apply a linearized algebraic flux corrected transport (FCT) algorithm (Kuzmin, 2009) which has been implemented to reduce the effects of spurious numerical oscillations for hyperbolic terms (Kosakowski and Watanabe, 2013).</i>

2	<p>Is there any condition on the time step such as a CFL condition?</p> <p><i>CFL condition is not embedded in the code itself. However, we have always taken it into account for the spatial and temporal discretization of reactive transport problems, in order to reduce the operator splitting errors. Additionally, if we solve the advection-dispersion-equation (ADE) explicitly, then we need to obey the CFL condition to ensure numerical stability. We added this information in the revised version (page 8: 6)</i></p>
3	<p>The flow and transport models can be nonlinear. How is the nonlinearity handled in OGS?</p> <p><i>We can use Picard and Newton-Raphson schemes in OGS for nonlinear problems such as Richards flow, density-dependent flow or multiphase flow. This point was added into the revised version (page 6:11).</i></p>
4	<p>Flow and transport and heat transfer can be coupled. How is the coupling handled in OGS?</p> <p><i>Within OGS we handle such couplings by using either a monolithic or staggered approach (Wang et al. 2011). With the monolithic approach, all unknowns from different coupled partial differential equations (PDEs) are solved in one global equation system. With the staggered approach, each PDE in the coupling is solved individually with update of the solutions of the other coupled PDEs in an iteration loop. The staggered approach was applied for the test examples shown in the manuscript. This information was added into the revised version (page 6: 15).</i></p>
5	<p>The coupling between OGS and PHREEQC is done with files. This is probably very costly. What is the computational overhead? Would it be possible to develop a more efficient interface between the two software?</p> <p><i>This is a very important issue raised by the reviewer. Indeed, the data exchange through files is relatively time consuming, and especially one of the crucial factors for simulating large problems. Additionally, for clusters, in which the file reading and writing is realized through the general parallel file system (GPFS), this process can become more time consuming, especially when the GPFS is highly loaded. In the meantime, we have developed and implemented a character string-based data exchange for the coupling between OGS and IPhreeqc, and analyzed the parallel performance within different cluster environments. In the revised manuscript we described methods (Sect. 2.3), results (including the analysis of computational overhead) (Sect. 2.4) and their performance improvement in distributed memory systems (Sect. 4.2).</i></p>
6	<p>Parallelism is defined with Domain Decomposition in OGS. What does DDC mean? As many cores as domains are used in OGS. How is decomposition done? Does it use METIS for example? How are communications between subdomains defined?</p> <p><i>Thank you for these questions to give the reader better insights into parallelization techniques. Domain decomposition (abbreviated by DDC) means the splitting of an initial-boundary value problem (IBVP) into smaller IBVPs on sub-domains. In a more figurative sense, the finite element mesh is decomposed into smaller mesh domains. We already added a short and explaining definition of DDC to the revised manuscript to make it clearer to the readers (page 11:11). Yes, we use METIS as a preprocessing tool for DDC in order to balance the node quantities and minimize the border nodes among subdomains efficiently. Then we extract and convert the information about element indices (global) and the internal border nodes and store them in a DDC file. Based on these subdomains and their mesh topology, the global equation system can be partitioned into local equation systems. For coupled processes different local matrices and vectors are assembled for each process and subdomain in individual CPU cores. Then we solve the local equation systems. The local solutions are obtained by a number of product calculations</i></p>

	<p><i>of system matrix and vectors.</i></p> <p><i>Finally, communication is required among sub-domains for updating the iteration steps if the components of local matrices and vectors are associated with border nodes among different sub-domains. Furthermore, communication is also required, when we collect the norm of production from different sub-domains. More detailed information of decomposition procedures can be found in previous works by Kalbacher et al. (2008) and Wang et al. (2009). These points were added to the revised version (Sect. 3.1).</i></p>
7	<p>Geochemistry is trivially parallel, since computations in spatial elements are independent. New cores can be added for these computations, but these cores remain idle when OGS computations are done. Is this efficient?</p> <p><i>In the current study, the proposed scheme provides the possibility for a flexible allocation of computational resources for calculation of geochemical and non-geochemical processes (e.g. flow, transport). Based on the specific features of the reactive transport problem and the computational resources that are available on the hardware platform, an allocation of the compute cores can be chosen to obtain the best parallel performance.</i></p> <p><i>It is true that the adding of cores solely for geochemical reaction will lead to the degradation of parallel efficiency, if we can still get further speedup for the non-geochemical processes by using these cores. For these problems, we can use the same number of compute cores for geochemical and non-geochemical processes. All cores will take part in both OGS and IPhreeqc computation. In this case, there are no "idle" cores.</i></p> <p><i>However, adding of compute cores solely for geochemical reactions can be overall efficient for certain problems, in which the degradation of parallel performance for non-geochemical processes occurs. In this case, we can adjust the number of cores for flow and mass transport at optimum while increase that for geochemical reactions. To demonstrate this point, we added a new test example in the revised manuscript (Sect. 4.3), in which the best parallel performance were achieved by adding new cores solely for chemistry.</i></p>
8	<p>Communications seem to be done with files. Is this efficient?</p> <p><i>Please see the reply to question 5.</i></p>
9	<p>Also, it seems that global communications occur quite often. Is this scalable?</p> <p><i>In the current coupling scheme, the global communication occurs only when the concentration values of the local buffers are transferred into a global concentration vector, after the calculation of geochemical reactions is done (see page 2379, line 5 - 6). The MPI routine MPI_Allreduce (...) is applied for this purpose ("MPI_Allgather" in page 2379, line 6 is a typo and hence was corrected in the revised manuscript). We think this operation should provide good scalability.</i></p>
10	<p>Numerical examples are simple. In fact, 2D and 3D examples remain 1D, since the results do not depend on the (y, z) coordinates. Could real 2D or 3D examples be presented?</p> <p><i>It is true that the 2D and 3D example presented in the manuscript are geometrically simple. We selected these examples, since the present manuscript is focusing on the development and analyzing the novel parallelization scheme. However, OGS has already been applied to simulate complex coupled processes (Kolditz et al., 2012). In the current scheme, geochemical reactions are solved locally on each finite element node, which means that it is independent of the complexity of the model geometry.</i></p> <p><i>Nevertheless, we presented a more complex reactive transport example in the revised version (Sect. 4.3).</i></p>
11	<p>In the examples, geochemistry represents most of the CPU time, so that parallelism is globally efficient. What happens if time of OGS computations becomes higher?</p>

	<p><i>Our new approach (creating two MPI groups) is advantageous for chemically dominated problems. Adding more cores solely for chemistry will not bring more advantages if a problem is dominated by flow/transport. In this case, using the conventional parallelization scheme (which is also possible in the current scheme through solely creating the first MPI group) will provide the best performance, as demonstrated in Sect 4.2. We already mentioned this point in 2383: 24. Nevertheless, we emphasized this to the reader in the revised manuscript (page 21:21).</i></p>
--	---

Response to Glenn Hammond

1	<p>This manuscript investigates the coupling of OpenGeoSys with IPhreeqc through a file-based data transfer with the use of MPI processor groups to assign additional processes to speed up the geochemical calculation. I applaud all efforts that further the application of high performance computing to subsurface simulation. The manuscript is somewhat novel.</p> <p><i>Thank you very much for these encouraging words and short summary of our manuscript.</i></p>
2	<p>However, the degree to which performance is improved by the addition of geochemical processes (i.e. the scalability) is questionable and the file based data transfer is clearly not scalable on large supercomputers. Please see my comments below regarding deficiencies in the current draft of this manuscript.</p> <p><i>Indeed, file based approaches scale poorly and it has been only used for the first proof of concept. In the meantime, we can present the results for the more efficient string-based coupling and we already included them into the revised manuscript. The according figures and descriptions have been updated as well. We address these changes more detailed below (see comment reply 6, 12).</i></p>
3	<p>The abstract states, “The open source scientific software packages OpenGeoSys and IPhreeqc have been coupled, to combine their individual strengths and features to simulate thermohydro-mechanical-chemical coupled processes in porous”. Please elaborate on what makes IPhreeqc better than the existing OGS chemistry capability. This may be obvious to the authors, but not the reader.</p> <p><i>The main advantage of this combination is that we can make use of the wide range of geochemical capabilities and customizable database of PHREEQC which helps us to setup a variety of coupled THMC^{react} simulations faster on HPC systems.</i></p> <p><i>The parallelization scheme makes this approach especially efficient if we face more complex geochemical systems. We included these points and rewrote the abstract to better explain these advantages to the readers as follows:</i></p> <p><i>“The open source scientific software packages OpenGeoSys and IPhreeqc have been coupled to setup and simulate thermo-hydro-mechanical-chemical coupled processes with simultaneous consideration of aqueous geochemical reactions faster and more straightforwardly on high performance computers. In combination with the elaborated and extendable chemical data base of IPhreeqc, it will be possible to setup a wide range of multi-physics problems with almost any chemical reaction that is known to influence water quality in porous and fractured media. A flexible parallelization scheme using MPI (Message Passing Interface) grouping techniques has been implemented, which allows an optimized allocation of computer resources for the node-wise calculation of chemical reactions on the one hand, and the underlying processes such as for</i></p>

	<i>groundwater flow or solute transport on the other hand. This technical paper presents the implementation, verification, and parallelization scheme of the coupling interface, and discusses its performance and precision."</i>
4	Line 2373:15 states that this manuscript evaluates a new parallelization scheme to provide "detailed information" for modelers and developers. My observation is that this manuscript often lacks detail. The manuscript could be greatly improved by providing a more detailed description of the implementation along with how the developers dealt with various issues that arose during implementation (so that others attempting to implement the same approach can leverage the authors' knowledge on this topic). This would make the manuscript much more impactful.
	<i>We are providing more details regarding the technical implementations as well as description of benchmarks etc. in the revised manuscript version, in order to facilitate the interested readers to implement this method and reproduce the results we presented (see page 7- 13). We are also addressing following issues which came up during the implementation of the current scheme.e.g.:</i> <ul style="list-style-type: none"> • <i>How to integrate updated IPhreeqc releases (see revision, page: 7:25)?</i> • <i>How to make it possible to run a block of code only for the ranks of a relevant MPI group? (see revision, page 12:27; Figure 6)</i>
5	2372:13 "An elaborated code concept and development can help to reduce the time needed for solution procedures and data communication." Do you mean "a well designed and efficient parallel implementation can help to reduce: : :"?
	<i>Yes, that is clearer. Thanks! We changed the sentence in the revision as "a well-designed code concept and efficient parallel implementation can help ..." (see revision, page 4: 15)</i>
6	2372:14 "Consequently in terms of coupled reactive transport modeling, process simulation and interaction should be closely tied to enable shared data structures and reduce data exchange procedures." Do you mean a well-designed interface should be developed that maximizes sharing of data structures in order to avoid duplication of data and/or computationally expensive mapping of data. This approach is expensive in at least two ways: (1) duplication of data structures and thus more memory use and (2) transfer of data between the two data structures (perhaps not that much of an issue if not file-based).
	<i>Yes, that's what we mean and it is indeed much less of an issue for a string based coupling interface, but still it is. A file-based data transfer is extremely time-consuming of course, especially in a clusters infrastructure in which file reading and writing is realized through the general parallel file system (GPFS). After we found our parallelization concept very promising, we started immediately to implement a string-based coupling between OGS and IPhreeqc in order to avoid this bottleneck. The gain in terms of performance is discussed more detailed below (Response 12, in-memory-coupling). The manuscript has been overworked accordingly (see Sect. 2.3, 3.2, 4.1, 4.2; Figure 1, 5, 6, 8, 9, 10).</i>
7	2372:17 – I believe that the "I" in IPhreeqc stands for "interface". This should be clearly stated as in many code names "I" stands for inverse. I would immediately assume that IPhreeqc is used for calibration and sensitivity analysis.
	<i>Yes, the "I" in IPhreeqc stands for "interface". We clearly stated this in the revised version (page</i>

	4:21). Thank you for the careful reading.
	2373:2 – “If a DDC approach, e.g. for flow and transport, is applied for the attached reactions system as well, then choosing the most suitable number of compute cores will lead always to a certain trade-off.” Any choice of compute cores will lead to a “certain trade-off”. Is the point that using the same number of compute nodes for reaction as was used for flow and transport does not always lead to the most optimal performance? This could be clarified.
8	<p><i>Yes, this is the point we would like to make and it is also one important motivation of our performance tests. We clarified this in the revised manuscript, and to be clearer to the readers, we added the following explanations (page 5:6):</i></p> <p><i>“In the operator splitting approach, the chemical reaction system is solved on each finite element node individually, so that no node-wise communication is necessary. However, flow and mass transport is bound to DDC, so that additional communication is needed to exchange the results along shared subdomain boundaries. Therefore a speedup for flow/transport is not given anymore, if communication and serial fractions are more time-consuming than the parallel fractions. As a consequence, an efficient number of compute cores assigned to the transport problem may already reach a maximum limit; while a further speedup for chemical system can still be present with the adding of more compute cores.”</i></p>
9	<p>2373:9 - “Global processes will be paralleled based on DDC method” . paralleled -> parallelized?</p> <p><i>Thank you for improving the language. Yes, it should be “parallelized” (revision, page 5:22)</i></p>
10	<p>2374:3 – A tradeoff exists between implementing biogeochemistry natively in a code versus coupling to a third-party library. On the one hand, native implementation could be much faster (with respect to run times) and provide more flexibility (e.g. it is generally easier to customize one’s own code). On the other hand, duplication of effort makes leveraging a third-party library more appealing. A comparison of OGS using native biogeochemistry vs. Phreeqc biogeochemistry on the same exact problem would better inform the reader regarding the computation overhead of coupling to a third-party library. I understand the Phreeqc brings a more extensive suite of biogeochemistry to the table for the scientist, but if a scientist is employing reactions that are already natively available in OGS, is it worth the effort to use the coupling to Phreeqc as the execution is likely more complicated and computationally expensive (to some degree)? A comparison between native OGS chemistry and Phreeqc would greatly improve the impact of this manuscript.</p> <p><i>In response to this comment, we simulated the van Breukelen benchmark by using the KinReact module of OGS (“native chemistry”), and compared its runtime with OGS#Phreeqc and standalone PHREEQC. The KinReact module is faster than the coupling (1.4s vs 32.7s) of course but does not have the wide range of geochemical capabilities as PHREEQC does (e.g. surface complexation, mineral nucleation etc.).</i></p> <p><i>See revised manuscript (page 11:11) and table 5.</i></p>
11	<p>2375:10 – “The source code of PHREEQC however is not changed: : :”. Do you mean that the IPhreeqc interface does not change while the Phreeqc source code can be refactored/updated. A well-designed interface allows one to modify code with minimal impact to the interface.</p> <p><i>Yes, we do not need to change the interface. For clarification we rephrased this part in the revised manuscript (page 7:25):</i></p> <p><i>“The interface itself is version independent and can stay unchanged after updates. For example,</i></p>

	<p><i>the integration of a new IPhreeqc release into the combined code can be realized simply by updating the IPhreeqc source code. Updates which will include/exclude IPhreeqc files only need a reconfigured list in the CMake file. This allows users to benefit continuously from code developments of both sides. ”</i></p>
12	<p>2375:16 – “In the first development step, : : ”. Okay, the file-based approach is a first step and the “string-based data exchange” is the next step. A more optimal approach would be in-memory coupling where IPhreeqc is called directly from OGS with no startup, initialization, etc. Just the raw data being passed in to a previously initialized IPhreeqc instance and a time step calculated with results passed back. Do you intend to take this implementation to that level of sophistication?</p> <p><i>We are grateful to the reviewer raising this issue. Yes, we were and are still taking the “in-memory” coupling into consideration, which means to access each other’s code internal data structures directly. Technically, this is possible. Tightly coupled, the interface would almost disappear. Based on our performance analysis on a Linux cluster with distributed memory and a network file system we found so far e.g. following proportional distributions for the presented 3D example (Figure 10):</i></p> <ul style="list-style-type: none"> - <i>The time spent for the interface takes e.g. around 30% (80 cores, 20 DDCs) of the total time with file-based approach. This portion increases significantly with increasing number of cores (Figure 10 c,d)).</i> - <i>The time for data exchange has been reduced by using the string-based approach, e.g. to around 10% for 80 cores and 20 DDCs (in Fig 10 c,d). Moreover, the overall time occupied by the interface increases much less with increasing number of cores.</i> <p><i>Therefore, for the presented examples there is only little evidence that the speedup and efficiency are tremendously improved by an “in-memory” coupling. Nevertheless, to have such a highly efficient interface would be important in order to make this approach scalable for large number of compute cores (see comment reply 29). This will be part of the planed studies on a massive parallel environment.</i></p> <p><i>The only concern for an “in-memory” coupling is that the OGS and IPhreeqc source code would be strong-coupled and this would result in some strong code dependencies as well. We think this is feasible and sustainably maintainable if both open-source communities can develop a common idea or even a standard for the shared data structures.</i></p> <p><i>We stated these points in the outlook part of the revised manuscript (page 22:10).</i></p>
13	<p>2375:19 - “: : will be passed to IPhreeqc to initialize the geochemical system” . A dear description of all the tasks that IPhreeqc must perform for each chemistry calculation would greatly improve the manuscript.</p> <p>Biogeochemical codes have many setup steps including potentially memory allocation, solver setup, the reading of the reaction network and databases containing parameters (log Ks, rate constants, stoichiometries, a basis [secondary species, minerals, surface complexes, etc.all defined with respect to a set of primary species], speciation to an initial condition, etc. The steps that IPhreeqc takes to perform this setup for each instance of IPhreeqc should be described in a few sentences to inform the reader of the overhead involved with setting up IPhreeqc over and over for each geochemistry calculation.</p> <p>Or are these steps performed once during initialization of OGS#IPhreeqc and the geochemistry calls are solely updates to the stored system? If the geochemical system is stored instead of being re-initialized for each geochemical calculation, what must be stored by OGS vs. IPhreeqc?</p>

	<p>This level of detail will better inform the reader of the overhead generated from coupling OGS with IPhreeqc and greatly improve the manuscript.</p> <p><i>As mentioned from the reviewer at first: during each chemistry calculation, IPhreeqc has to complete the following tasks:</i></p> <ul style="list-style-type: none"> <i>i) create an instance;</i> <i>ii) load a thermodynamic database (e.g. reaction networks, species, phases, log ks);</i> <i>iii) read and run the input (process updated data such as lists of elements, phases, and species; perform different calculations such as aqueous speciation, ion exchange, etc.);</i> <i>iv) retrieve results from calculations;</i> <i>v) release instance from memory.</i> <p><i>We described that more clearly but briefly in the revised manuscript (see page 9:3), since very detailed information can be found in Charlton et al. (2011) or Parkhurst and Appelo (2013). Furthermore we would like to state, that the involved overheads (steps other than iii and iv) are small compared to the total simulation time, especially for large problems. Actually, we already included them into the IPhreeqc time for all the test examples we presented. These overheads account for 3.8% of the total simulation time of the Engesgaard benchmark; 2.3% of that of the van Breukelen benchmark; 0.4% of the 3D parallelization example (with 20 cores and 20 DDCs). This information was added to the revised manuscript (page: 9:11, 10:17, 11:13). Nevertheless, it is attractive to minimize these overheads by performing steps i and ii only once during the initialization and step v only once at the end of the entire simulation. We stated this in the outlook part of the revised manuscript (page: 22:14).</i></p>
14	<p>2375:24 – “ : :an input file for IPhreeqc will be prepared.” Again, does each call from OGS to IPhreeqc entail a complete IPhreeqc run (initialization, execution, finalization) or does IPhreeqc initialize and sit idle waiting for the input file to appear and execute a single time step, but without terminating at the end. I understand that the IPhreeqc cores sit idle waiting for the next IPhreeqc step, but the question is whether the entire IPhreeqc code is re-initialized over and over for each geochemistry step. If this is stated elsewhere in the manuscript, please point me to the page:line.</p> <p><i>Please, see comment response 13.</i></p> <p><i>To avoid misunderstanding, we rewrote the whole paragraph (revision, page 8: 20, 9:11).</i></p>
15	<p>2376:11 – “The latter two benchmarks will be shortly introduced here” . If either of these benchmarks have been altered in any way (i.e. if they differ from the original cited papers), a full description of the benchmark should be included in this manuscript to ensure reproducibility of results, or to allow for others to compare the performance of their simulator to OGS#IPhreeqc. If the cited references provide adequate detail the exact problem executed by OGS#IPhreeqc, this is not necessary.</p> <p><i>We provided a more detailed description and supplementary (PHREEQC input file) for the first benchmark (see page 10; uploaded supplementary material: Part 1). Concerning the second benchmark, the adequate details are already given in van Breukelen et al. (2005).</i></p>
16	<p>2376:20 – A comparison of the results between OGS#IPhreeqc and OGS-Chemapp is provided. Please discuss a comparison of the computational performance. Can you provide a detailed discussion of the breakdown of computation (e.g. % time spent in transport vs. chemistry, overall run time)? Even if this is a serial run, the breakdown helps the reader better understand</p>

	<p>the cost of coupling the codes (i.e. overhead resulting from OGS#IPhreeqc integration).</p> <p><i>We can provide a computational performance comparison between OGS#IPhreeqc and OGS-ChemApp for serial simulations. Providing numbers for HPC could be a bit demanding, as ChemApp requires installation and licenses on the cluster, and this seems to be somehow problematic on the machine here in Leipzig. Furthermore, we compared this benchmark with standalone PHREEQC simulation (batch version). We changed the time stepping of the according models simulated by OGS#IPhreeqc and OGS-ChemApp, in order to apply the same temporal and spatial discretization for all the three codes while fulfilling the Courant condition required by PHREEQC. The updated results are given and discussed in the revised manuscript (see page 10:11, table 4, figure 2).</i></p>
17	<p>2377:9 – Again, if the conceptual model executed in the second scenario differs from van Breukenlen et al. (2005), please explain for reproducibility purposes. It may be better to include a full description of the flow, transport and biogeochemistry conceptual models. Databases can be cited (or provided) to minimize the data reporting requirement.</p> <p><i>We are using the original benchmark which is fully described in van Breukenlen et al. (2005), and there are no modifications regarding the model setups.</i></p>
18	<p>2377:10 – How does the OGS#IPhreeqc computational performance compare to PHREEQC?</p> <p><i>For the first benchmark, the runtimes of OGS#IPhreeqc and PHREEQC are 7.861 s and 5.74 s. For the second benchmark, the runtimes are 32.671 s and 14.196 s, respectively. We gave an overview of the computation time for OGS#IPhreeqc in the revised manuscript in table 4, 5 and at page 10:15 and 11:11.</i></p>
19	<p>2377:18 – “All cores take part in solving the geochemical reaction system, while a subset of cores is used to solve the DDC related processes.” Is it safe to assume that all reported speedups and efficiencies factor in the idle time resulting from geochemistry cores sitting idle while flow and transport are calculated?</p> <p><i>The reported performance factors result from measuring:</i></p> <ul style="list-style-type: none"> • <i>the required time to solve flow and transport (i.e. DDC cores running while geochemistry cores sitting idle)</i> • <i>the required time to solve geochemistry (i.e. making use of all cores and no cores are sitting idle at all)</i> • <i>the required time for the interface (uses all cores as well)</i> <p><i>Does this answer the question?</i></p>
20	<p>2378:4 – This paragraph leads me to believe that each geochemistry process initializes IPhreeqc and waits for parameters to be passed to iPhreeqc at geochemical every time step. In other words, IPhreeqc is not restarted or reinitialized at every time step (no memory allocation, basis swapping, database reading, etc.). Is this true?</p> <p><i>IPhreeqc is restarted and reinitialized at each time step. Please see comment reply 13. To be clearer here, we added “(including all the IPhreeqc tasks described in Sect. 2.3)” after the sentence “which will invoke the calls to IPhreeqc for compute cores in MPI_Group2” (revision, page 13:14).</i></p>
21	<p>2378:18 – But does Ballarini et al. (2014) use OGS#IPhreeqc? If not, the parallelization schemes for reactive transport may be similar but the algorithms differ. Please clarify.</p> <p><i>No, Ballarini et al. (2014) used a loose coupling in which OGS calls simply the PHREEQC executable (page 2375: 6). We removed the sentence.</i></p>

22	<p>2379:1 – This paragraph leads me to believe that solely state variables (concentrations, molar volumes, etc.) are passed between OGS and IPhreeqc during the simulation and no initialization/memory allocation occurs after startup. Please confirm.</p> <p><i>Memory re-allocation for IPhreeqc is needed for each time step. Here, we modified the sentence "each compute core will execute IPhreeqc by using a specific input file" as "each compute core will perform a complete call to IPhreeqc by using a specific input string" (revision, page 14:9).</i></p>
23	<p>2379:10 – My experience is that Linux boxes with large core counts provide marginal performance at best for sparse nonlinear systems of equations solved implicit in time. The breakdown in performance is due to non-optimal communication/memory access through the system BUS and memory hierarchies. In general, I get a maximum speedup of around 8x on these machines no matter how many processes I employ. In fact, as the number of processes increases, performance can degrade.</p> <p><i>Yes, we agree that the performance for the calculation of finite-element-method (FEM) related processes (flow and transport) can be limited in such a hardware environment. However, chemistry is solved locally on each node within the operator splitting approach. So no communication among border nodes of different subdomains is necessary. Better speedups for chemistry were observed for the test examples presented in our study.</i></p>
24	<p>2380:14 – Please comment (or point me to a reference) on how OGS conserves local mass balance when employing a finite element discretization. I know that other FE codes such as FEHM employ a control volume approach to conserve mass. Otherwise, wouldn't the lack of local mass conservation result in instabilities in the geochemistry side of the framework (e.g. potential negative concentrations)?</p> <p><i>The standard finite element method is locally not fully mass conservative, of course. We are aware that this e.g. can lead to negative concentrations especially at sharp reaction fronts. Within OGS, we have implemented a linearized algebraic flux corrected transport (FCT) scheme and can apply it for mass transport. This FCT scheme can ensure sufficient local mass balance accuracy to a certain extend but is computational more expensive. We make no use of this scheme in the presented tests and examples. However, details regarding this implementation can be found in Kosakowski and Watanabe (2013). We could include some more detailed information to this manuscript of course, but the presented benchmark studies show sufficient accuracy and we think that a discussion about the pro and cons of FEM for reactive transport modelling is out of scope of this paper.</i></p>
25	<p>2380:16 – As the authors likely understand, the 1D and 2D (but really still 1D) problems are chemically dominant due to the simple transport (and flow?) being calculated in the problem. This could be stated.</p> <p><i>Thank you, we stated this at page 16:3. In the revised manuscript we provided a new example which involves more complex flow and transport simulations (see Sect. 4.3 or comment reply 32).</i></p>
26	<p>General note regarding Figures 8b and 9b. Relative speedup should be plotted relative to the lowest process count. The term "relative" refers to a speedup relative to the minimum number of processes available. Such plots should include an ideal speedup that proceeds from 0 (or 1 in the case of a log-log plot) to the maximum multiple of the number of processes. In other</p>

	<p>words, suppose the minimum number of processes were 4 and maximum 16. The ideal line should start at 0,0 and proceed through 4,4 (the start of the data) through 16,16. The attached image (Gwo et al., 2001 Figure 5.jpg) from:</p> <p>Gwo, J. P., E. F. D' Azevedo, H. Frenzel, M. Mayes, G. T. Yeh, P. M. Jardine, K. M. Salvage, and F. M. Hoffman (2001), HBGC123D: A high-performance computer model of coupled hydrogeological and biogeochemical processes, <i>Comput. Geosci.</i>, 27(10), 1231 – 1242, doi 10.1016/s0098 – 3004(01)00027-9.</p> <p>illustrates the proper way to plot both speedup and the breakdown of various functionality within the total run time. Otherwise presented, it is very difficult for the reader to determine the degree to which a code is scalable. The author can state that a code is scalable, but the reader needs proof in the plotted data through comparison with the ideal speedup line. Notice the “linear speedup” line (or ideal speedup) in Gwo et al. So, take for instance Figure 8b. The ideal line should run from 0,0 to the maximum number of processes employed (20,20). The data will start at 4,4 and run through 20,20. Each line (iPhreeqc, other, total) should start at 4,4, though the ending points will differ depending on the scalability of each category. Figure 9b' s ideal line will go from 20,20 to 80,80 and the data will all originate at 20,20. (As an alternative, one could set the relative speedup value to 1 for the lowest process count and label the axis accordingly. In that case, Figure 8b would run from 4,1 to 20,5 [ideal 5x speedup] and Figure 9b from 20,1 to 80,4 [ideal 4x speedup]. These plots should be fixed for this manuscript to be accepted.)</p>
	<p><i>Thank you very much for the detailed suggestions to improve the quality of our figures. We modified the figures according to your comments.</i></p>
27	<p>2381:2 – Please add a line indicating ideal speedup to Figure 8b (see comment on speedup Figures above). Within the context of the current figure, the line should be straight and run from 0,0 to 20,20 (or 4,1 to 20,5). This will help the reader compare the actual performance to ideal.</p> <p><i>Thank you, we really like this idea and we included ideal-speedup-lines. (see Fig. 8b and 9b)</i></p>
28	<p>2381:14 – Please plot Figure 8 c and d with log-log scale. In doing so, please add a single “ideal” curve which should be linear. The initial time value does not matter; it is the slope that the reader needs for comparison purposes. Again, this enables the reader to better judge performance themselves comparing the actual performance to ideal performance.</p> <p><i>We plotted Figure 8c and d regarding to wall-clock time with log-log scale and an “ideal” slope.</i></p>
29	<p>2381:14 – The challenge with this hybrid approach is that the domain decomposition flow/transport side of the code will be the bottleneck. With reaction taking 90% of the simulation time for DDC=4, the maximum speedup that one could ever get for DDC=4 (relative to DDC=4) is 10x with an infinite number of reaction processes. That is based on Amdahl’ s law and assuming DDC is the serial fraction. The question is how well this approach scales to large #s of processes.</p> <p><i>Our approach has the potential to scale to large number of processes and bring more benefit than the conventional approach for chemical dominated problems.</i></p> <p><i>If the serial fraction accounts for 10% of the total simulation time, then it is no doubt that the maximum speedup one can get is 10x, regardless of the approach and number of cores employed. In reality, we may get even less speedup, if we take the communication overhead into account. Here, we added another term to the classic Amdahl’s law, which represents the communication</i></p>

overhead (see equation (1)).

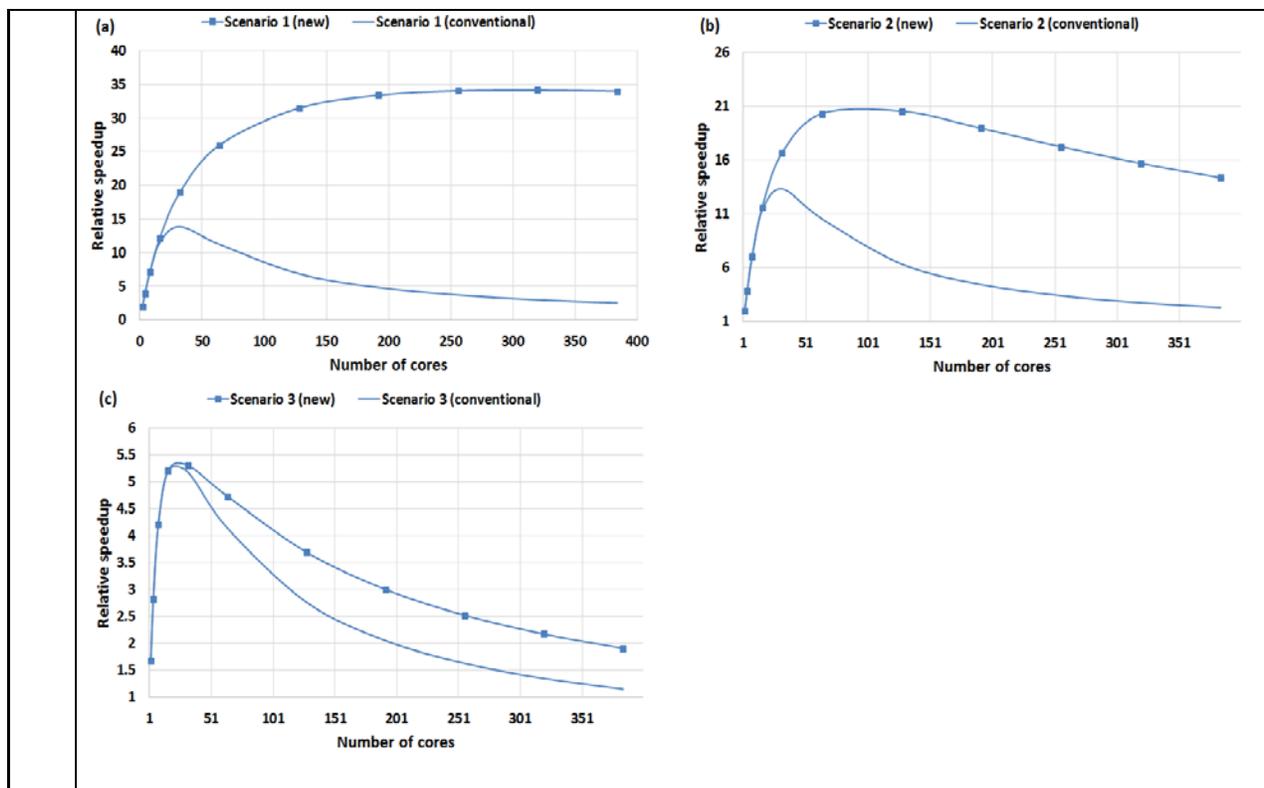
$$S(n) = \frac{T_p(1) + T_s}{\frac{T_p(1)}{n} + T_s + T_c * (n - 1)} \quad (1)$$

where $S(n)$ is the relative speedup compared to the serial simulation when n compute cores are employed. $T_p(1)$ and T_s are the time spent for parallel fraction and sequential fraction, respectively. T_c is a factor for communication overhead (assuming there is a linear relationship between overhead and employed compute cores).

The time spent for communication will lead to the degradation of the parallel performance, if it becomes dominant. For a coupled reactive transport code, communication overhead can relate to two parts i.e. DDC and interface. With a conventional approach ($n = \text{DDC}$), we have to stand with the growth of both parts with the increase of compute cores. In our new approach, we can fix the time consumption for the DDC (including its communication overhead) at its optimal, thus avoid its further increase.

In order to estimate the potential benefit we can get from the new approach, we would like to present three scenarios here. In the first scenario, calculation of chemistry is dominant and the communication overhead introduced by the interface is much smaller than that of the DDC (“best case”). In the second scenario, chemistry is still dominant for the simulation. The communication time required by the interface increases, but it is still smaller than that of the DDC. In the third scenario, calculation of flow/transport becomes as dominant as chemistry and the communication overhead for the interface is now similar with that of the DDC (“worst case”). For all the three scenarios, we applied both conventional approach and the new approach.

The results are shown in figures below, which demonstrate that the proportion of chemistry calculation and the overhead of the interface are two important factors influencing the benefit and scalability of the new method. If a problem is dominated by flow/transport and the interface is time consuming (scenario 3, Figure c), then the benefit we can get from the new method is rather limited. However, if a problem is dominated by reaction and the time consumption for the interface is considerably small (e.g. through “in-memory” coupling) (scenario 1, Figure a), then the new method has the potential to be scalable for large number of compute cores.



Section 4.2 – Again please modify the plots as follows:

Figure 9a – Can you add an “ideal” plane that will illustrate deviation from ideal performance (I believe that one can calculate this algebraically for every DDC). This may make the figure unreadable and too complicated to decipher: : just a thought. Figure 9b – Add an “ideal” line, e.g. running from 0,0 to 80,80 (or 20,1 to 80,4) Figures 10 a-d – Replot on a log-log scale and add an line representing “ideal” slope. Also, rescale the y-axis (wall clock time) in each plot to better display results for comparison purpose. For instance Figure 10d could be rescaled to have a maximum wall clock time of 1000 seconds. It is difficult to read otherwise.

30 The “ideal plane” is again a very interesting idea. However, we think the diagram would be overloaded. Therefore we changed the figures in the following way:

- We added an “ideal” line to Figure 9b as the reviewer suggested.
- For Figure 10 (a-d) we applied log-log scale, rescaled them and included the “ideal” slopes. Additionally, we compared the results between string- and file-based approaches.
- In order to make a better comparison, we present the time spent for interface in Fig 10c.
- In Fig10d, we illustrate the total time by using both approaches.

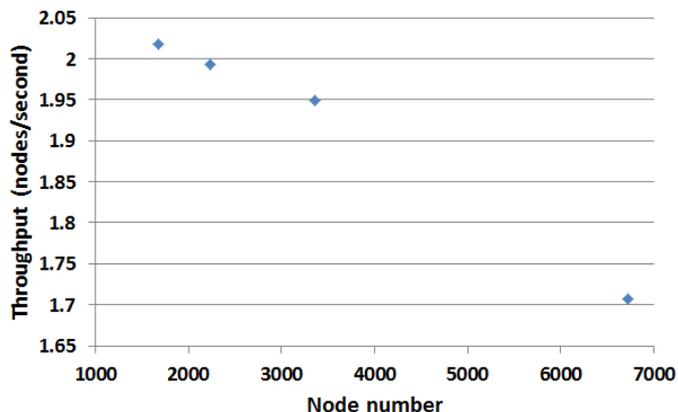
The according discussion can be found in the revised manuscript at page 18:12.

31 2382:8 – With the ideal line included in Figure 9b, can you explain the superlinear speedup in the IPhreeqc performance? Maybe I am misunderstanding something, but the degree to which that performance is superlinear is beyond cache effects, etc. A quick eyeball estimate shows ~4.75x speedup on 4x as many processes (20 vs 80). Can you explain the extra .75x speedup?

In the current scheme, a compute core has to calculate chemistry for fewer nodes, when more cores are available. We think the increase of throughput with the decrease of nodes number for a single compute core can explain the super-linear speedup observed.

In the figure shown below the throughput is plotted against the node number elaborated by one

compute core. With the given conditions (hardware architecture, code structure, etc.), the throughput decreases as one core has to handle more nodes. In other words, a compute core can calculate the chemistry for one node faster, when it has to process fewer nodes. However, to understand this behavior in more details, we have to dig more into the IPhreeqc code.



Additionally, we do not think the extra speedup is beyond the cache effects. Depending on the situation, a speedup up to 10x or even higher can be achieved. For example, based on the documentation below (last access: Jun 27, 2015), the following latencies are given for different cache levels of Core i7 Xeon 5500 Series:

L1 CACHE hit, ~4 cycles

L2 CACHE hit, ~10 cycles

L3 CACHE hit, line unshared ~40 cycles

L3 CACHE hit, shared line in another core ~65 cycles

L3 CACHE hit, modified in another core ~75 cycles

https://software.intel.com/sites/products/collateral/hpc/vtune/performance_analysis_guide.pdf

General comment: Your test problems are chemically dominant. The flow and solute transport is essentially 1D. In more realistic modeling scenarios, one would expect flow and transport to become somewhat more dominant. Since the addition of cores to MPI Group 2 for chemistry does not benefit this 3D test problem much, even with the quasi-1D flow and transport, can you devise a realistic problem scenario where the hybrid DDC approach would run scale better with greater than the number of DDC cores? In other words, other than in the first, very simple 2D problem, I don't see the advantage of allocating processes to MPI Group 2, when they will sit around idle during flow and transport.

32

Adding more cores solely for chemistry will not bring more advantages if a problem is dominated by flow/transport. In this case, using the conventional parallelization scheme (which is also possible in the current scheme through solely creating the first MPI group) will provide the best performance. We already mentioned this point in 2383: 24. Nevertheless, we rewrote that part to emphasize this to the reader (revision, page 21:21).

As already mentioned in our answers to comment 8, calculation of chemical and non-chemical processes can have different features regarding to their parallel performance. Our new approach has more benefits than the conventional one, if severe performance degradation happens for flow and mass transport. In this case, fixing the number of DDCs at its optimal while adding more cores for chemistry would still lead to an overall better performance.

To demonstrate this point, we added a new test example in the revision (sect 4.3), which involves

	<p><i>the simulation of unsaturated-saturated flow. In this example, allocating compute cores to MPI_Group2 leads to better speedups than the conventional DDC approach (see Fig. 13b). In this example the benefit coming from chemistry side is somehow limited, since the minimum time for flow/transport takes already more than 28% of the total time. It means that the maximum speedup we can further achieve is less than 3.6 based on the fundamental point given by the reviewer in comment 29.</i></p> <p><i>However, if a problem is chemically dominant and the coupling interface is highly efficient (see scenario 1 in comment reply 29), then our approach should be scale better than the conventional one.</i></p>
33	<p>Final comments: It is not clear to me that the new methods presented in this manuscript will have a significant (beneficial) impact on subsurface simulation techniques. There are several deficiencies in the approach taken to link the two codes.</p> <p>First, limiting the cores in the second process group to solely chemistry calculations results in these cores sitting idle during flow and transport. For small, chemically dominant problems this approach may not hamper performance much, but for large scale massively-parallel simulation, this approach is not scalable as the processes should be added to the DDC side of the problem. I understand that the algorithm provides the flexibility to add processes to either side, but in reality addition to the conventional DDC side is likely the best alternative for most realistic problems. If adding cores to the DDC portion of the problem does not result in speedup, it is likely that the problem size per process (i.e. number of degrees of freedom per core) is already too small for efficient parallel computing. Speeding up the chemistry calculation will provide limited benefit in that case since the DDC portion is the bottleneck.</p> <p><i>For large scaled problems, in which flow and mass transport calculation is most time consuming, the conventional approach would be the best choice, which was also shown in Sect. 4.2 in our manuscript. We rewrote the conclusion part in order to present this point more clearly (21:18). We think that our new method can provide more benefits than the conventional approach for geochemically dominated problems with limited (or poor) parallel scalability for flow and mass transport (see response to comment 32). As already discussed in reply 29, the benefits we can get from this new approach will depend mainly on the proportion of chemical reaction and the communication overhead of the interface. The more chemical reactions dominate and the more efficient the interface, the better speedup we can obtain with the available computational resources. We are planning to setup more and even theoretical examples to test and evaluate the limitation of such approaches on large HPC machines in the near future.</i></p>
34	<p>Second, the use of file IO to transfer data between codes is obviously much slower than “in memory” data transfer and will not scale to large process counts on large problems (as the manuscript demonstrates); so why publish that approach in the first place?</p> <p>If in memory data transfer (either through the “string-based approach” mentioned in the manuscript or through double precision arrays) is the ultimate objective, why not implement the better approach and demonstrate that it scales.</p> <p><i>Our initial motivation of this work was to develop a concept, which can realize the flexible allocation of compute resources for geochemical and non-geochemical processes. That is why we focused on proving our concept and chose the file-based coupling as our first development step (technically it’s easier to be implemented).</i></p> <p><i>In the meantime, we have implemented the string-based approach and discussed its advantages in Sect 4.2 (revision) as well as comment reply 12.</i></p>

	<i>There is still much room to improve our interface. We fully agree an “in-memory” coupling between OGS and IPhreeqc would be the most efficient implementation, which would also be important to ensure the scalability of this approach for large number of compute cores (see comment reply 28). This will be part of our future studies.</i>
	For these reasons, I would claim that the approach is novel, but certainly not revolutionary as implemented at this point in time. I believe that by addressing the questions/comments presented above, the manuscript will be greatly improved.
35	<i>We would like to thank the reviewer again for his very helpful review and very constructive suggestions. We believe these comments helped to improve our manuscript significantly. We agree that the presented work might be just a small step towards improving the performance of reactive transport modeling (RTM) – but we think that it shows the benefit of the novel concept and contributes indirectly to evaluate the range of application of such couplings in general.</i>

Literatures not included in the manuscript

Kuzmin, D.: Explicit and implicit FEM-FCT algorithms with flux linearization, J. Comput. Phys. 228, 2517–2534, 2009.

A parallelization scheme to simulate reactive transport in the subsurface environment with OGS#IPhreeqc [5.5.7-3.1.2](#)

W. He^{1,3}, C. Beyer⁴, J. H. Fleckenstein², E. Jang^{1,3}, O. Kolditz^{1,3}, D. Naumov⁵, and T. Kalbacher¹

¹Department of Environmental Informatics, Helmholtz Centre for Environmental Research – UFZ, Leipzig, Germany

²Department of Hydrogeology, Helmholtz Centre for Environmental Research – UFZ, Leipzig, Germany

³Applied Environmental System Analysis, Technical University Dresden, Dresden, Germany

⁴Institute of Geosciences, Geohydromodeling, Christian-Albrechts-University of Kiel, Kiel, Germany

⁵Faculty of Mechanical and Energy Engineering, Leipzig University of Applied Science, Leipzig, Germany

Correspondence to: W. He (wenkui.he@ufz.de)

Abstract

~~This technical paper presents an efficient and performance-oriented method to model reactive mass transport processes in environmental and geotechnical subsurface systems. The open source scientific software packages OpenGeoSys and IPhreeqc have been coupled~~ , ~~to combine their individual strengths and features to~~ to setup and simulate thermo-hydro-mechanical-chemical coupled processes ~~in porous and fractured media~~ with simultaneous consideration of aqueous geochemical reactions . ~~Furthermore, a faster and more straightforwardly on high performance computers. In combination with the elaborated and extendable chemical data base of IPhreeqc, it will be possible to setup a wide range of multi-physics problems with almost any chemical reaction that is known to influence water quality in porous and fractured media. A flexible parallelization scheme using MPI (Message Passing Interface) grouping techniques has been implemented, which allows an optimized allocation of computer resources for the node-wise calculation of chemical reactions on the one hand, and the underlying processes such as for groundwater flow or solute transport on the other hand. The coupling interface~~ This technical paper presents the implementation, verification, and parallelization scheme ~~have been tested and verified in terms of precision and performance~~ of the coupling interface, and discusses its performance and precision.

1 Introduction

Reactive transport modeling is an important approach to better understand, quantify and predict hydro-biogeochemical processes and their effects on subsurface environments. It is of growing interest among the fields of geotechnical engineering applications and environmental impact assessments and is used e.g. in contaminated site remediation or water resources management, to predict the environmental fate of organic and inorganic substances and pollutants in soil or groundwater reservoirs (e.g. Ballarini et al., 2014; Hammond et al., 2010, 2011, 2014; Henzler et al., 2014; Lichtner et al., 2012; Molins et al., 2010; Riley et al., 2014; Yabusaki et al., 2011). Geotechnical applications employ reactive

transport simulations e.g. to quantify geochemical processes in geological nuclear waste repositories (e.g. Kosakowski and Watanabe, 2013; Shao et al., 2009; Xie et al., 2006) or to evaluate CO₂ geological sequestration (e.g. Beyer et al., 2012; Li et al., 2014; Pau et al., 2010; Xu et al., 2004, 2006).

5 In the last decades, much effort has been invested to develop practical tools for reactive transport modeling (Steefel et al., 2014), such as PHREEQC (Parkhurst and Appelo, 1999, 2013), OpenGeoSys (OGS) (Kolditz et al., 2012), HYTEC (van der Lee et al., 2003), ORCHESTRA (Meeussen, 2003), TOUGHREACT (Xu and Pruess, 2001; Xu et al., 2006, 2011), eSTOMP (Yabusaki et al., 2011), HYDROGEOCHEM (Yeh and Tripathi, 1990),
10 CrunchFlow (Steefel et al., 2014), MIN3P (Mayer et al., 2002) or PFLOTRAN (Lichtner et al., 2015). Since each code has its own strengths and limitations, coupling of different codes, i.e. one software applies another and/or vice versa, is an indispensable choice and a straightforward solution to make use of combined capabilities of different codes. Existing approaches, which apply tool coupling methods to simulate reactive transport processes
15 are e.g. HYDRUS and PHREEQC (Jacques and Šimůnek 2005; Šimůnek et al., 2006); COMSOL and PHREEQC (Nardi et al., 2014; Nasir et al., 2014; Wissmeier and Barry, 2011); OGS-GEMs (Kosakowski and Watanabe, 2013; Shao et al., 2009); OGS-BRNS (Centler et al., 2010); OGS-ChemApp (Li et al., 2014); OGS-PHREEQC (Xie et al., 2006; de Lucia et al., 2012); MODFLOW-UFZ and RT3D (Bailey et al., 2013), or MODFLOW-
20 MT3DMS, i.e. PHT3D (Morway et al., 2013).

Due to the complexity of physical, geochemical, and biological processes involved, the development of a reactive transport simulator, which has comprehensive numerical modeling capabilities, is a challenging task. The robustness and computational efficiency of a numerical simulator are of vital importance, because reactive transport modeling is often
25 accompanied with other challenges such as numerical precision and stability (de Dieuleveult and Erhel, 2010; Kosakowski and Watanabe, 2013; Wissmeier and Barry, 2011) or expensive computational time.

Especially for realistic reactive transport simulations at larger scales, i.e. from field scales to catchment or reservoir scale, high complexities of hydrogeological and geochemical sys-

tems as well as high spatial–temporal resolution of reactive zones are required to ensure plausible and accurate model results. In these cases, iterative simulations of different scenarios or setups e.g. for model calibration and parameter sensitivity analysis becomes extremely difficult and time-consuming on desktop computers with limited computational resources (Hammond et al., 2014; Kollet et al., 2010; Lichtner et al., 2012; Yabusaki et al., 2011).

Parallelization is an established approach to improve computational performance and with the additional benefit from continuous innovation of modern hardware and software development (Hanappe et al., 2011; Wang et al., 2014). PFLOTRAN, a parallel multiscale and multiphysics code for subsurface multiphase flow and reactive transport (Hammond et al., 2012, 2014; Lichtner et al., 2012), or TOUGH-MP, the parallel version of TOUGH2 (Zhang et al., 2008; Hubschwerlen et al., 2012), apply domain decomposition (DDC) methods for their parallel framework. Yabusaki et al. (2011) implemented a one-sided communication and global shared memory programming paradigm in eSTOMP.

~~An elaborated~~ A well-designed code concept and ~~development~~ efficient parallel implementation can help to reduce the time needed for solution procedures and data communication. Consequently in terms of coupled reactive transport modeling, process simulation and interaction should be closely tied to enable shared data structures and reduce data exchange procedures.

In the current work, OGS has been coupled with the new C++ module of PHREEQC, called IPhreeqc ("I" stands for "interface"). In this operator splitting approach, chemical reactions are calculated locally on each finite element node, whereas processes such as groundwater flow and mass transport are calculated globally. OGS is an open source FEM simulator for multi-dimensional thermo-hydro-mechanical-chemical (THMC) coupled processes in porous and fractured media (Kolditz et al., 2012). In other words, OGS is able to simulate e.g. water and/or gas flow together with heat and mass transport processes in fully and partly saturated media. IPhreeqc on the other hand, inherits all the functionalities of PHREEQC, i.e. it is capable of modelling aqueous, mineral, gas, surface, ion-exchange, solid-solution equilibria and kinetic reactions, but also provides a well-defined set of meth-

ods for data transfer and management additionally (Charlton and Parkhurst, 2011). Both codes are open source, i.e. the technical coupling could be realized directly on the code level.

The optimum amount of the required computer resources for DDC related processes (global process flow and mass transport) and chemical reactions can be quite different. ~~If a DDC approach, e. g. for flow and transport,~~ In the operator splitting approach, the chemical reaction system is solved on each finite element node individually, so that no node-wise communication is necessary. However, flow and mass transport is bound to DDC, so that additional communication is needed to exchange the results along shared subdomain boundaries. Therefore a speedup for flow/transport is not given anymore, if communication and serial fractions are more time-consuming than the parallel fractions. As a consequence, an efficient number of compute cores assigned to the transport problem may already reach a maximum limit; while a further speedup for chemical system can still be present with the adding of more compute cores. If the number of compute cores for flow and transport is applied for the attached reactions system as well, then choosing the most suitable number of compute cores will lead always to a certain trade-off. the most optimal parallel performance cannot always be obtained.

Hence, a new parallelization scheme based on MPI grouping techniques is developed for the OGS#IPhreeqc interface to enable a flexible distribution of different amount of computer resources for DDC related processes and geochemical reactions, thus to allocate optimum number of compute cores for both types of processes simultaneously. Global processes will be ~~paralleled~~ parallelized based on DDC method, whereas the parallelization of geochemical reactions is completely independent from global processes in terms of number of compute cores employed and the way to group finite element nodes for different compute cores.

This technical paper describes in the following the coupling interface of OGS#IPhreeqc and evaluates the performance of the new parallelization scheme to provide detailed information for modelers and developers to apply reactive transport simulation on high performance computer infrastructures.

2 Codes and methods

After a brief description of both codes the coupling interface is introduced and verified on the basis of two benchmark examples. After that the technical implementation as well as verification of the proposed parallelization scheme is described (Sect. 3).

5 2.1 OpenGeoSys

Based on object-oriented concepts for numerical solution of coupled processes, OGS provides plenty of possibilities to simulate a broad spectrum of processes related to reactive transport modeling (Kolditz et al., 2012).

For example, OGS can be applied to simulate different kind of flow processes such as incompressible and compressible groundwater flow, overland flow, density-driven flow, unsaturated flow, two phase as well as multiphase flow. Picard and Newton-Raphson schemes can be applied for nonlinear problems such as Richards flow and density dependent flow. In OGS, transport of components in fluid phases is simulated based on the advection–dispersion equation, ~~while~~. For flow and transport processes, both implicit and explicit time discretization schemes can be used. To couple processes such as flow, transport and heat transport, either monolithic or staggered approach can be applied (Wang et al., 2011).

Within OGS, geochemical reactions can be modeled by using internal libraries (e.g. the KinReact module for kinetically controlled biogeochemical-reactions; Ballarini et al., 2014) or external couplings with geochemical solvers (e.g. Xie et al., 2006; Shao et al., 2009; Kosakowski and Watanabe, 2013; Centler et al., 2010; Beyer et al., 2012; Li et al., 2014). ~~The code-~~

OGS has already been parallelized using MPI (Wang et al., 2009; Ballarini et al., 2014) and PETSc (Wang et al., 2014). More detailed information relating to OGS development concept, code resources, benchmarking, etc. can be found at <http://www.opengeosys.org/>.

2.2 PHREEQC and IPhreeqc

PHREEQC is one of the most widely used open source geochemical solvers. It provides a variety of geochemical reaction capabilities (Parkhurst and Appelo, 1999, 2013). Beside batch reaction simulations, its current capabilities include inverse and one-dimensional reactive transport modeling. IPhreeqc is a C++ module of PHREEQC which is specially designed for the coupling of PHREEQC with other codes. It provides a well-defined series of methods to interact with a client program (Charlton and Parkhurst, 2011). For example, PHREEQC simulation input data can be prepared as a file or a character string in the client program and executed by PHREEQC with different methods such as *RunFile* or *RunString*. Besides writing selected output results into a file, individual data items at a certain position of the result array can be accessed and returned to the client program by using the *GetSelectedOutputValue* method. More detailed information on IPhreeqc and its data manipulation methods can be found in Charlton and Parkhurst (2011).

2.3 OGS#IPhreeqc interface

In the current study, both source codes, OGS and IPhreeqc are statically linked to allow accesses of all the functionalities of both codes (open source concept). The OGS#IPhreeqc interface is well encapsulated into a general framework for reactive transport modeling in OGS, which has already been described in detail by Beyer et al. (2012). Unlike the previously existing coupling scheme between OGS and PHREEQC presented by Xie et al. (2006), in which the PHREEQC is called externally through a system call to a PHREEQC binary executable, in the new coupling presented here, a call to PHREEQC can be realized directly by accessing functions provided by the IPhreeqc module. The ~~source code of PHREEQC however is not changed, which allows the merging of new releases from both codes rather conveniently. This development concept allows the user~~ interface itself is version independent and can stay unchanged after updates. For example, the integration of a new IPhreeqc release into the combined code can be realized simply by updating the IPhreeqc source code. Updates which will include/exclude IPhreeqc files only

need a reconfigured list in the CMake file. This allows users to benefit continuously from the code development from code developments of both sides.

The sequential non-iterative (SNIA) approach for operator splitting is applied in the coupling procedure, which means that no iterations are made between mass transport and geochemical reactions. Consequently, adequate small time step sizes are required to reduce the operator splitting errors. Additionally, the Courant-Friedrichs-Lewy (CFL) condition should be taken into account for the spatial and temporal discretization. Figure 1 illustrates the general procedure for reactive transport modeling with OGS#IPhreeqc, which is described in the following.

In the first development step, a file-based approach for data exchange between OGS and IPhreeqc is applied. A character-string based coupling was then developed, which reduces the time consumption for data exchange. The current paper will focus on introducing the character string-based approach. Nevertheless, the parallel performance of both approaches in a cluster will be compared in Sect.4.2.

Within OGS, the model setup is realized by using different input files, which defines specific aspects of the model (e.g. initial-boundary condition). In order to trigger the coupling interface, an additional OGS input file has to be provided, which is very similar to a PHREEQC input file (without the transport module). Based on the file, the interface will define the geochemical system such as reaction types, master solution species etc.

Before entering the time stepping loop, the geochemical system will be initialized first. In order to achieve this, initial values of the system state such as component concentrations and temperatures on each finite element node will be passed to IPhreeqc to initialize the geochemical system. An IPhreeqc input string will then be prepared, which contains information of the defined geochemical system and relevant values of state variables for all nodes. A call to IPhreeqc will be performed to run the input string. During each time step, after OGS has calculated the flow field by simulating different flow processes mass transport of each mobile chemical component will be calculated. Then on each node, same procedures will be performed as during the initialization: concentration values of each component as well as other state variables such as pressure and temperature for all nodes

will be forwarded to the coupling interface, ~~in which an input file for IPhreeqc; an input string will be prepared. The~~, followed by a call to IPhreeqc.

A complete call to IPhreeqc will be realized by ~~using the IPhreeqc functions: CreateIPhreeqc to create a taking the following steps: I) create a new instance of IPhreeqc; LoadDatabase to load the; II) load a thermodynamic database for the geochemical system; and RunFile to;~~ III) read and run the specific PHREEQC input files. ~~After execution of IPhreeqc, an output file will be generated by IPhreeqc, which will be read string; IV) retrieve the results from IPhreeqc and V) release the IPhreeqc instance from memory. A more detailed description of these procedures and relevant IPhreeqc functions applied can be found in Charlton et al. (2011) and Parkhurst and Appelo (2013).~~

~~These procedures have to be repeated during each call to IPhreeqc within each time step. However, the overhead (calling of functions other than III and IV) involved in the call to IPhreeqc is small compared to the total simulation time, which will be analyzed in Sect 2.4.~~

~~After the call to IPhreeqc, the IPhreeqc output string will be handled~~ by the interface during the reaction post-processing. Based on the updated chemical species concentrations, several feedback functions can be applied to update the porosity, permeability, saturation as well as density for flow, heat and mass transport processes. For example, in the case of mineral dissolution or precipitation, the porosity and permeability changes can be evaluated.

2.4 Verification of the coupling interface

The coupling between OGS and IPhreeqc was tested and verified by using several benchmarks for reactive transport problem types such as ion exchange (example 11 of Parkhurst and Appelo, 1999), carbonate mineral precipitation and dissolution (Engesgaard and Kipp, 1992; Beyer et al., 2012), and isotope fractionation (van Breukelen et al., 2005). The latter two benchmarks will be ~~shortly~~ introduced here. ~~A comparison of the computational performance by using different codes will also be presented.~~

The first presented test example is the Engesgaard benchmark. It describes the phenomenon occurs when a 0.5 m long 1-D calcite column is flushed with a solution contain-

ing magnesium [chloride](#): calcite dissolves continuously as the solution moves towards the downstream direction, whereas dolomite precipitates temporarily at the calcite dissolution front. Calcite dissolution/precipitation are simulated as equilibrium reactions, whereas that of the dolomite is ~~modelled~~ [modeled](#) as kinetic reactions using ~~the rate~~ [a Lasaga-type rate law \(Lasaga et al., 1994\)](#). [The kinetic rate parameters from Palandri and Kharaka \(2004\) are applied \(see Table 1\)](#). The material properties of the column as well as the initial and boundary conditions are given in Tables ~~1 and 2~~ [and 3](#), respectively. The [model domain is discretized into 100 uniform line elements](#). [Total simulation time is 21333.32 s with a constant time step size of 533.333 s](#). In the current study, this benchmark is simulated by using OGS#IPhreeqc, OGS-ChemApp and a batch version of PHREEQC (version 3.2.0). [A PHREEQC script is provided in Part 1 of the supplementary material](#). A comparison of the simulation results ~~between by using the three codes is illustrated in Fig. 2~~. [Apart from the amount of Dolomite, the simulation results of OGS#IPhreeqc and OGS-Chemapp, PHREEQC and OGS-ChemApp \(from Beyer et al., 2012\) shows a very good agreement show generally good agreements](#), as illustrated in Fig. 2. [Table 4 lists the execution times by using these codes. For this example, OGS#IPhreeqc is slightly slower than PHREEQC, but around 2 times faster than OGS-ChemApp. Among the total execution time of 7.861 s, the proportion of OGS#IPhreeqc interface and the overhead involved in calling to IPhreeqc are 12.7 % and 3.8 %, respectively.](#)

The second benchmark is based on the 1-D multistep isotope fractionation model from van Breukelen et al. (2005), which simulates the sequential reductive dechlorination of tetrachloroethene (PCE) to ethane (ETH) in a 876 m long aquifer over a period of 20 years. The model domain, aquifer properties as well as initial and boundary conditions are illustrated in Fig. 3.

The intermediate products during the degradation include tri- and dichloroethylene (TCE, DCE), vinyl chloride (VC). The whole sequential reductive dechlorination chain is illustrated as follows: $\text{PCE} \rightarrow \text{TCE} \rightarrow \text{DCE} \rightarrow \text{VC} \rightarrow \text{ETH}$.

The ^{12}C and ^{13}C isotopes of each chlorinated hydrocarbons (CHCs) are modeled as separate species. Totally there are 11 chemical species including chloride as tracer, which is

produced in each dechlorination reaction. During degradation the kinetic isotope fractionation of each compound is assumed to be constant. More detailed information regarding the kinetic rate expressions and relevant parameters can be found in van Breukelen et al. (2005). The model domain consists of 120 line elements. The total simulation time is discretized evenly into 100 time steps.

The simulated concentration profile of the light CHC isotopes and relevant $\delta^{13}\text{C}$ [‰] isotope signatures along the model domain are compared with those simulated using a ~~standalone batch~~ version of PHREEQC (version 3.2.0) and KinReact module of OGS (Fig. 4), showing ~~a good agreement~~ good agreements for both concentration profiles of the light CHC isotopes and corresponding isotope signatures.

Table 5 shows the computational performances by using the three approaches. For this example, the execution time of OGS#IPhreeqc is around twice as much as that of the batch version of PHREEQC. The time spent for the interface and the overhead for calling to IPhreeqc account for 14.7 % and 2.3% of the total simulation time. The KinReact module is much faster than the other two approaches. Nevertheless, it does not have the wide range of geochemical capabilities as PHREEQC does (e.g. surface complexation, mineral nucleation etc.).

3 Parallelization of OGS#IPhreeqc

In this section we describe the parallelization method for the numerical simulation of reactive transport processes with OGS#IPhreeqc. For the parallelization of groundwater flow and mass transport, the OGS internal DDC scheme (~~see Sect. 2.1~~) is employed. For the parallelization of geochemical reactions a loop parallelization is applied. All cores take part in solving the geochemical reaction system, while only certain cores are used to solve the DDC related processes.

3.1 DDC scheme in OGS

Domain decomposition (DDC) is the procedure to split an initial-boundary value problem (IBVP) into smaller IBVPs on subdomains. In a more figurative sense, the finite element mesh is decomposed into smaller mesh domains.

5 In the present study, METIS is used as a preprocessing tool for DDC in order to balance the node quantities and minimize the border nodes among subdomains efficiently. Then the information about element indices (global) and the internal border nodes will be extracted and stored in a DDC file. Based on these sub-domains and their mesh topology, the global equation system can be partitioned into local equation systems. For coupled processes different local matrices and vectors are assembled for each process and subdomain in individual CPU cores.

10 After that the local equation systems will be solved. The local solutions are obtained by a number of product calculations of system matrix and vectors.

15 Finally, communication is required among subdomains for updating the iteration steps if the components of local matrices and vectors are associated with border nodes among different subdomains. Furthermore, communication is required as well, when the norm of production from different subdomains needs to be collected. More detailed information of decomposition procedures can be found in previous works by Kalbacher et al. (2008) and Wang et al. (2009).

3.2 Parallelization scheme

20 Figures 5 and 6 illustrate the general idea of the parallelization scheme. The two different MPI groups, i.e. MPI_Group1 and MPI_Group2 and related intra-communicators are created by using MPI functions MPI_Group_incl and MPI_Comm_create. The compute cores which belong to MPI_Group1 will run most part of the OGS code including all DDC related processes (groundwater flow, mass and heat transport) and geochemical reactions, whereas those of MPI_Group2 will only run a small part of code related to geochemical simulation.

25 Technically, this is realized by using the following selection statement, so that the execution of a piece of code can be constrained to processors of the relevant MPI group:

`if(myrank_group1 != MPI_UNDEFINED){...}`

For each MPI operation in the entire code, it is important to identify the relevant MPI group and choose the correct MPI communicator.

A “for” loop for MPI_Group2 is created directly in the main function of the OGS code. In each time step, after the calculation of global flow and mass transport process, PHREEQC input files-strings for all compute cores will be created by compute cores of MPI_Group1.

Then-A big difference between the serial and parallel algorithm should be noticed here. In a serial simulation, a single input string will be prepared during each time step (see Sect. 2.3). However, in the parallel simulation introduced here, the information of geochemical system and values of state variables for all the nodes will be distributed into several input strings, whose number is equal to that of the total compute cores employed.

After the preparation of input strings, compute cores of MPI_Group1 will send a signal to start signals as well as input strings to relevant compute cores of MPI_Group2, which will invoke the calls to IPhreeqc for compute cores in MPI_Group2 -(including all the IPhreeqc tasks described in Sect. 2.3), once the input strings are received. At the same time, compute cores of MPI_Group1 will begin to call to IPhreeqc as well. After PHREEQC calculations are complete in both MPI groups, flow and mass transport processes will start again with the next time step in MPI_Group1, while compute cores of MPI_Group2 will wait for the signal from MPI_Group1 (using blocking receive MPI_Receive) to restart the receiving of input strings and calls to IPhreeqc. After compute cores of MPI_Group1 have run through the complete time stepping loop reaching the end of the simulation, another-a killing signal will be sent to MPI_Group2, which will force its compute cores to jump out of the chemical reaction loops. Then MPI_Finalize will be executed to terminate the MPI environment. As a special case, when the number of subdomains equals that of the compute cores, only MPI_Group2-will-not-roup1 will be created. In this case, no communication between the two MPI groups is required. This corresponds to the parallelization scheme for reactive transport simulations applied in Ballarini et al. (2014)-

As mentioned above, file-based-a character string-based data transfer is applied to exchange concentration values between mass transport and geochemical reaction simula-

tions. In each time step, after the simulation of mass transport, concentration values of all components in all finite element nodes will be stored in a global concentration vector. For each compute core a node list vector will be generated through which finite element nodes are allocated to the respective compute core, and their concentration values can be accessed from the global concentration data structure by using this vector. Since the generation of the node list vector is completely independent from the domain decomposition, flexible groupings of finite element nodes can be realized to ensure an optimum load balance of compute cores for the calculation of geochemical reactions. During the execution of geochemical reactions, each compute core will execute-perform a complete call to IPhreeqc by using a specific input file-string (including all the IPhreeqc tasks mentioned in Sect. 2.3). A relevant PHREEQC results file-string will then be generated and sent back to the corresponding compute core of MPI_Group1 (if the compute core belongs to MPI_Group2). After all compute cores finish their calls to IPhreeqc, compute cores of MPI_Group1 will read-handle all the result files-strings and store the concentration values of all components in respective local buffers. The values of all local buffers will then be transferred to a global concentration vector by applying the MPI_Allgather-llreduce method, before the updated concentrations of different components are sent back to mass transport process again.

3.3 Computational platforms

The correctness and efficiency of the proposed scheme were tested on two different computational platforms. The first platform is a multi-core Linux machine called “ENVINF”. It contains 40 “Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz” CPU cores and has a shared memory of approximately 500 GB RAM among these 40 cores. A maximum of 20 cores can be used by a single user at a time. The second platform is a Linux based (CentOS 6 as the operating system) cluster, in the following called “EVE”. It consists of 1008 (Intel XEON X5650 @ 2.6 GHz) CPU cores and 5.5 TB of RAM. Computer nodes are connected with 40 GBit s⁻¹ QDR Infiniband network interconnect. The peak performance is 10 TFlop s⁻¹.

In order to make the results comparable by using both platforms, for all tests in the EVE cluster, job requests were made to guarantee the use of compute nodes with 20 free slots

when submitting to the job queue. Of course jobs can also be submitted without this constrain, however, since in this case the MPI jobs may be distributed to more compute nodes than necessary in order to allow an earlier execution, more inter-compute node communications may have to be made over the network, which would worsen the performance of the parallelization scheme.

3.4 Verification of the parallelization scheme

The 1-D benchmark of isotope fractionation is extended to 2-D and 3-D to apply the proposed parallelization scheme. Figure 7a and b show the concentration distribution of the light isotope VC along the 2-D model domain and the 3-D model domain at the end of the simulation, respectively. All test results on both parallel computing platforms show very good agreements with serial simulation results.

4 Performance tests and analysis

In this section, the performance of the parallelization scheme is tested by using two three examples differing by dimension and problem size. The ~~model size of the first 2-D example is relatively small compared to the second 3-D test.~~ first two examples are simple extensions of the 1-D benchmark of isotope fractionation. However, they differ with each other on problem size. Hence, the influence of the problem size on the parallel performance can be shown. In the third example, geochemical reactions are added upon a saturated-unsaturated flow system. The influence of the simulation of non-linear flow (Richards flow) on the parallel performance can thus be studied.

4.1 Isotope fractionation 2-D

As the first test example, the 1-D PHREEQC model of van Breukelen et al. (2005) is extended to 2-D (876 m \times 100 m, see Fig. 7a). The finite element mesh consists of 1331 nodes and 1200 uniform rectangular elements (120 \times 10). Unlike the 1-D model, here the total

simulation time (20 years) is evenly discretized into 200 time steps. With a single core on the ENVINF machine (see Sect. 3.3) the simulation time is 578 s. Chemical reaction is the most time-consuming part of the simulation due to the simple flow and transport calculations, which takes 92.2% of the total simulation time.

5 The performance of the current parallelization scheme is demonstrated in Fig. 8. In Fig. 8a the relative speedup in comparison to a ~~single core simulation~~ simulation with 4 cores and 4 DDCs is illustrated as a function of number of DDCs and total compute cores. If we fix the number of DDCs at a specific value and vary the total number of compute cores from 4 to 20, we can observe a continuous increase of relative speedup for all
10 DDCs with growth of the number of compute cores. The speedup of DDC = 8 is generally much better than that of DDC = 4. ~~Above DDC=12 there is no big difference between the speedup behavior for different DDCs anymore.~~ Curve AB in Fig. 8a represents relative speedups for combinations in which the number of compute cores equals the number of DDCs. In Fig. 8b curve AB is once again illustrated ("~~total~~" "total") together with the
15 ~~speedup of chemical reactions and speedup of "other" processes which are mainly global processes like~~ relative speedups of IPhreeqc calculation (which includes the complete call to IPhreeqc) and groundwater flow and mass transport. We can observe that the speedup of "~~other~~" processes flow and mass transport reaches its maximum when the number of compute cores exceeds ~~12-16~~. As shown by Wang et al. (2009), adding of ~~sub-domains~~ subdomains will increase communication between ~~sub-domain~~ subdomain border nodes. As a consequence, the parallel efficiency for calculation of DDC related processes will reduce when number of border nodes becomes comparable with the total number of finite element nodes. The speedup of reaction however is generally much better and increases continuously as more compute cores are provided. In the operator splitting approach chemical
20 reactions are solved locally on each finite element node, hence no direct communication among different nodes is necessary.

25 Figure 8c and d show the breakdown of the total time for different ~~processes~~ compute cores with a DDC = 4 and a DDC = 12. It is clearly shown that chemical reaction is the most time-consuming part of the simulation in both cases. With a DDC = 4 reactions take

up to ~~9086.5~~ 86.5 % of the total time when only 4 compute cores are applied, and drops to ~~around 7057.2~~ 57.2 % if 20 compute cores are applied; whereas for a DDC = 12 it becomes ~~8380.5~~ 80.5 % of the total time for 12 compute cores, and goes down to ~~around 7773.1~~ 73.1 % for 20 compute cores. In both cases time ~~of "other" processes for flow and mass transport~~ stays almost unchanged for different number of compute cores because the number of DDCs is fixed. The time for interface mainly includes preparing input ~~files strings~~ for IPhreeqc, communication among different compute cores, ~~reading output files handling output strings~~ from IPhreeqc. ~~As shown in Fig. 8c and d, Averagely this part of time stays also nearly unchanged and relatively low compared to other processes. accounts for 5.2 % and 10.8 % of the total simulation time for DDC = 4 and DDC = 12, respectively.~~

Generally, the way of coupling and parallelization is shown to be efficient already for small sized reactive transport problems in a shared memory system such as ENVINF.

4.2 Isotope fractionation 3-D

The second test case is a 3-D extension ($876 \text{ m} \times 100 \text{ m} \times 10 \text{ m}$, see Fig. 7b) of the 2-D test example which consists of 134 431 nodes and 120 000 hexahedral finite elements ($120 \times 100 \times 10$). The simulation time with 2 compute cores with 2 DDCs on ENVINF is 37.5 h.

Similar to the 2-D test example (Sect. 4.1), for the 3-D test case the relative speedup on the EVE cluster is illustrated as a function of number of DDCs and total compute cores in Fig. 9a; Fig. 9b shows a breakdown of curve AB into speedups of ~~global flow and mass transport~~ processes and chemical reactions. If we use the same number of compute cores and DDCs, a nearly linear speedup with the increase of the compute cores can be observed. By using 80 compute cores simulation time can be reduced to around ~~4537~~ 537 min. As problem size increases, the speedup effects of both DDC related processes as well as chemical reactions become stronger. Similar to the results of the 2-D example, in the 3-D example geochemical reaction shows a much better speedup ~~than global processes~~ (superlinear) than flow and mass transport.

However, if we fix the number of DDCs at a specific value and increase the total compute cores further, ~~there is not much speedup observed for almost all~~ the resulting speedup is

not so significant, especially for lower number of DDCs (see Fig. 9a). This behavior is quite somewhat different from what we have observed in the 2-D example.

The reason behind lies mainly in the fact, that the ratio between the time consumption for reactions and mass transport (flow) are different in these two examples. In the 2-D example, the time consumption for calculation of flow and mass transport is rather low comparing with that of reactions. In the 3-D example, the time consumption for flow and mass transport is on the same-similar magnitude as that of reactions (see Fig. 10a and b). For 20 compute cores with 20 DDCs, flow and mass transport together takes 36.2 % of the total time, whereas that of IPhreeqc calculation is 54.3 %. As a consequence, the saving of time in the calculation of reactions alone, which is obtained by increasing compute cores, cannot bring a significant speedup for the entire simulation.

Fig. 10 compares the total time and its breakdowns by using string- and file-based parallelization approaches for this problem. From Fig. 10a and Fig.10b we can see that there are only slight differences between the two approaches on the time spent for flow, mass transport and chemistry. However, when we compare the time for interface in Fig. 10c, we can find that the string-based approach shows big advantages over the file-based one, in which the file reading and writing is realized through the general parallel file system (GPFS). By using string-based data exchange, this part of time is small compared to the calculation of mass transport or chemistry. In worst case, it takes 10.2 % of the total time (80 cores with 20 DDCs); whereas that of the file-based coupling can reach up to 30.9 % (80 cores with 20 DDCs). Generally, it decreases with the increment of DDCs. For a certain DDC, this portion of time for the file-based coupling increases dramatically with the adding of more compute cores; whereas that of the string-based coupling is much less dependent on the number of compute cores.

Fig. 10d illustrates the total times for different DDCs. For a fixed number of DDCs, the string-based coupling scales much better than the file-based coupling, as it needs much less time for the interface. It is obvious that the best parallel performance for each DDC can be obtained (which is more close to the ideal slope), when the number of compute cores and that of the DDC stay the same. Hence, to achieve a better speedup for a large problem,

it is important to reduce the time consumption for flow and mass transport as well by using more DDCs.

In the current scheme the writing and the reading of files on the EVE cluster is realized by using the general parallel file system (GPFS). As we can observe in the

4.3 Uranium leaching problem

This test problem is based on the 2-D example of Šimůnek et al. (2012) and Yeh and Tripathi (1991), which simulates uranium leaching at a mill tailing at a hillslope scale (see Fig. 11). The substitution of calcite by gypsum also occurs with the release of acid and sulfate from the tailing. It is worth mentioning that redox reactions are not taken into account in this example. The water flow in both unsaturated and saturated zone is modeled. Totally 35 species and 14 minerals are considered for geochemical reactions. A detailed description of model setup and the simulation results are available in the supplementary material (Part 2).

The 2-D domain consists of 14648 triangle elements with 7522 nodes. A total simulation time of 1000 days is discretized into 6369 time steps varying from 1×10^{-7} s to 24000 s. The same time discretization is adopted for all parallel simulations introduced below. The wall-clock time for a simulation of this example with 2 cores and 2 DDCs on the ENVINF machine takes around 6.0 h.

Parallel simulations are performed with combinations of compute cores varying from 20 to 60 and DDCs ranging from 2 to 60. Fig. 10c, for a large problem file writing and reading through the GPFS is a time-consuming procedure. Figure 10d shows that other time consumption in the coupling interface is relatively small, which mainly consists of blocking communication (e.g. MPISUBSCRIPTNB12 illustrates relative speedups compared to the simulation with 20 cores and 2 DDCs as a function of compute cores and DDCs. Best speedups are achieved by using 60 cores and DDCs ranging between 8 to 16. By using more DDCs, degradation of parallel performance occurs, which is especially obvious when applying 20 DDCs. This phenomenon is mainly caused by the performance degradation of linear solver for flow and mass transport. Fig. 13a shows the breakdown of the total

time corresponding to speedup curve AB in Fig. 12. Major components such as IPhreeqc, linear solver and interface are illustrated. The time for linear solver increases dramatically after 20 DDCs. Over 40 DDCs there is a slight "recovery" of the parallel performance. The reason is that the performance degradation of linear solver becomes slower, while the time consumptions for IPhreeqc, interface and matrix assembly decrease further. Because 20 cores are applied for all the DDCs varying from 2 until 20, time for IPhreeqc stays nearly the same for these DDCs. It is worth mentioning that the time for the interface can become expensive even by using the string-based coupling, when limited number of compute cores is responsible for preparing and processing large number of in- and output strings (the number of cores is scale larger than that of DDCs). By applying 20 cores with only 2 DDCs, it takes up to 23.4 B% arrier) between different of the total time.

Fig. 13b presents the total time for different DDCs as a function of compute cores. However, this part of time can also increase when working load for calculation of reactions, file writing and reading are unbalanced among different compute cores. Generally, the parallel performance of this example is poor when compared with the two previous examples, since the minimum time consumption for flow and mass transport, which can be achieved by using DDCs between 8 and 16, has already taken a large proportion of the total time (more than 28 %). In this example, the maximum parallel performance is obtained by using more compute cores (i.e 60) than the number of DDCs (i.e. 8 or 12). This shows the advantage of present parallelization scheme over the conventional DDC approach, which keeps the number of cores the same with that of DDCs.

5 Conclusions and outlook

This technical paper introduced the coupling interface OGS#IPhreeqc and a parallelization scheme developed for the interface. Furthermore, the parallel performance of the scheme was analyzed.

The OGS Although OGS already has native chemistry modules and coupling interfaces with other chemical solvers, the OGS#IPhreeqc interface presented in the current study is

indispensable, which can greatly benefit from the wide range of geochemical capabilities and customizable database from PHREEQC. Based on a sustainable way of coupling, which can continuously benefit from the the continuous code development and updating from two open source communities -can be integrated efficiently. A character string-based data exchange between the two codes is developed to reduce the computational overhead of the interface. Particularly, it is much more efficient than a file-based coupling for parallel simulations on a cluster, in which file writing and reading is realized through the GPFS. The parallelization scheme is adjustable to different hardware architectures, and suitable for different types of high performance computing (HPC) platforms such as shared-memory machines or clusters.

The parallelization scheme provides more flexibility to arrange compute resources for different computational tasks by using the MPI grouping concept. The appropriate setting of DDCs and total compute cores is problem dependent. If calculation of geochemical reaction dominates the total simulation time, e.g. for small sized problems with simple hydrogeological but complex geochemical system, then using more compute cores brings more significant speedup than simply increasing the number of DDCs; if the time consumption of

If the time consumption for flow and mass transport is in the same magnitude as geochemical reactions, to increase the number of compute cores and DDCs simultaneously is most efficient.

The current parallelization scheme will be especially useful for then using the conventional DDC approach will be the best choice, as demonstrated in Sect. 4.2. This is especially the case for large problems, in which a the time spent for flow and solute transport become more dominant.

If a problem is dominated by geochemical reactions (e.g. for small to middle sized problems with complex geochemical system), then the new approach (creating two MPI groups) can be advantageous, especially when further increase of the number of DDCs above the optimum will lead to a strong degradation of parallel performance for flow or mass transport (as a consequence of e.g. increasing inter-compute-node communication

or number of linear iterations). In this case, better ~~speedup~~ speedups may still be obtained by fixing the number of DDCs at the optimum while ~~increasing the number of compute cores~~ allocating more compute cores for the second MPI group to accelerate the calculation of chemical reactions.

5 Even though the ~~current parallelization scheme has shown good parallel performance in shared and distributed memory systems~~ time consumption for the interface has been reduced significantly by applying the character string-based coupling, there is still space for improvement to reduce the time consumption for communication and data transfer between OGS and IPhreeqc. This would be especially important for the approach to be
10 scalable for large number of compute cores. A more promising way would be to use an “in-memory” coupling, in which the internal data structures of both codes can be accessed from both sides more directly. This could be feasible and sustainably maintainable if a common idea or even a standard for the shared data structures can be developed together by both open-source communities. Another improvement that can be made is to initialize and finalize IPhreeqc only once during the entire simulation, so that the overhead involved calling IPhreeqc can be minimized.

Blocking communication techniques, like MPI_Barrier were applied to ensure the correct sequence of process coupling. An unbalanced work load distribution for chemical reactions, like in heterogeneous problems with sharp transient reactive fronts or reaction hot spots,
20 could affect the parallel performance as well. Hence, more intelligent ways to ensure efficient load balance still remain as an important task.

~~File writing and reading through the GPFS of a cluster system is time-consuming, especially when increasing problem size. As the next step, a character string-based data exchange strategy will be implemented, in order to minimize the time consumption for data exchange~~
25 In the current study, the available computational resource was limited. It will be part of the future work to test and evaluate the strengths and limitations of this approach on larger high-performance computing machines.

Recently, the SeS Bench (Subsurface Environmental Simulation Benchmarking) benchmarking initiative has started a project to test the parallel performance of different reactive

transport modeling tools. In the near future, more complex benchmarks and real-world applications will be tested in the framework of this project to improve the parallel performance of the current scheme and evaluate the suitable range of applications of similar approaches for reactive transport modeling at larger different scales.

6 Code availability

The source code for the serial version of OGS#IPhreeqc (file-based) was released as an official version of OGS and can be obtained with the following link under an open source license: <https://github.com/ufz/ogs5>.

Relevant information for OGS compilation can also be found from there. To use the interface, one has to select the option OGS_FEM_IPQC during CMake configuration. The source code of the fully parallel version (string-based) can be provided after the acceptance of the manuscript, and will be part of the following official OGS releases.

Acknowledgements. This work is funded by Helmholtz Centre for Environmental Research (UFZ), Leipzig, Germany. The authors thank the relevant Integrated Project T31 “Catchment Dynamics” of POF3 and its coordinator Maren Göhler. The authors thank Lars Bilke for his technical supports for the coupling procedure and Ben Langenberg for helps in running simulations on the EVE cluster.

The service charges for this open-access publication have been covered by a Research Centre of the Helmholtz Association.

References

Bailey, R. T., Morway, E. D., Niswonger, R. G., and Gates, T. K.: Modeling variably saturated multi-species reactive groundwater solute transport with MODFLOW-UZF and RT3D, Ground Water, 51, 752–761, 2013.

- Ballarini, E., Beyer, C., Bauer, R. D., Griebler, C., and Bauer, S.: Model based evaluation of a contaminant plume development under aerobic and anaerobic conditions in 2-D bench-scale tank experiments, *Biodegradation*, 25, 351–371, 2014.
- 5 Beyer, C., Li, D., De Lucia, M., Kühn, M., and Bauer, S.: Modelling CO₂-induced fluid-rock interactions in the Altensalzwedel gas reservoir. Part II: coupled reactive transport simulation, *Environ. Earth Sci.*, 67, 573–588, 2012.
- Centler, F., Shao, H. B., De Biase, C., Park, C. H., Regnier, P., Kolditz, O., and Thullner, M.: GeoSys-BRNS – A flexible multidimensional reactive transport model for simulating biogeochemical subsurface processes, *Comput. Geosci.*, 36, 397–405, 2010.
- 10 Charlton, S. R. and Parkhurst, D. L.: Modules based on the geochemical model PHREEQC for use in scripting and programming languages, *Comput. Geosci.*, 37, 1653–1663, 2011.
- de Dieuleveult, C. and Erhel, J.: A global approach to reactive transport: application to the MoMas benchmark, *Comput. Geosci.*, 14, 451–464, 2010.
- de Lucia, M., Bauer, S., Beyer, C., Kühn, M., Nowak, T., Pudlo, D., Reitenbach, V., and Stadler, S.:
15 Modelling CO₂-induced fluid-rock interactions in the Altensalzwedel gas reservoir. Part I: from experimental data to a reference geochemical model, *Environ. Earth Sci.*, 67, 563–572, 2012.
- Engesgaard, P. and Kipp, K. L.: A geochemical transport model for redox-controlled movement of mineral fronts in groundwater-flow systems – a case of nitrate removal by oxidation of pyrite, *Water Resour. Res.*, 28, 2829–2843, 1992.
- 20 Hammond, G. E. and Lichtner, P. C.: Field-scale model for the natural attenuation of uranium at the Hanford 300 Area using high-performance computing, *Water Resour. Res.*, 46, W09527, doi:10.1029/2009WR008819, 2010.
- Hammond, G. E., Lichtner, P. C., and Rockhold, M. L.: Stochastic simulation of uranium migration at the Hanford 300 Area, *J. Contam. Hydrol.*, 120–121, 115–128,
25 doi:10.1016/j.jconhyd.2010.04.005, 2011.
- Hammond, G. E., Lichtner, P. C., and Mills, R. T.: Evaluating the performance of parallel subsurface simulators: an illustrative example with PFLOTRAN, *Water Resour. Res.*, 50, 208–228, 2014.
- Hanappe, P., Beurivé, A., Laguzet, F., Steels, L., Bellouin, N., Boucher, O., Yamazaki, Y. H., Aina, T., and Allen, M.: FAMOUS, faster: using parallel computing techniques to accelerate the FA-
30 MOUS/HadCM3 climate model with a focus on the radiative transfer algorithm, *Geosci. Model Dev.*, 4, 835–844, doi:10.5194/gmd-4-835-2011, 2011.
- Henzler, A. F., Greskowiak, J., and Massmann, G.: Modeling the fate of organic micropollutants during river bank filtration (Berlin, Germany), *J. Contam. Hydrol.*, 156, 78–92, 2014.

- Hubschwerlen, N., Zhang, K. N., Mayer, G., Roger, J., and Vialay, B.: Using Tough2-MP on a cluster-optimization methodology and study of scalability, *Comput. Geosci.*, 45, 26–35, 2012.
- Jacques, D. and Šimůnek, J.: User Manual of the Multicomponent Variably-saturated Flow and Transport Model HP1, Description, Verification and Examples, Version 1.0, SCK-CEN-BLG-998, Waste and Disposal, SCK-CEN, Mol, Belgium, 79 pp., 2005.
- [Kalbacher, T., Wang, W., Watanabe, N., Park, C. H., Taniguchi, T. and Kolditz, O.: Parallelization concepts and applications for the coupled finite element problems, *Journal of Environmental Science for Sustainable Society*, 2, 35–46, 2008.](#)
- Kolditz, O., Bauer, S., Bilke, L., Bottcher, N., Delfs, J. O., Fischer, T., Gorke, U. J., Kalbacher, T., Kosakowski, G., McDermott, C. I., Park, C. H., Radu, F., Rink, K., Shao, H., Shao, H. B., Sun, F., Sun, Y. Y., Singh, A. K., Taron, J., Walther, M., Wang, W., Watanabe, N., Wu, Y., Xie, M., Xu, W., and Zehner, B.: OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media, *Environ. Earth Sci.*, 67, 589–599, 2012.
- Kollet, S. J., Maxwell, R. M., Woodward, C. S., Smith, S., Vanderborght, J., Vereecken, H., and Simmer, C.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, *Water Resour. Res.*, 46, W04201, doi:10.1029/2009WR008730, 2010.
- Kosakowski, G. and Watanabe, N.: OpenGeoSys-Gem: a numerical tool for calculating geochemical and porosity changes in saturated and partially saturated media, *Phys. Chem. Earth*, 70–71, 138–149, 2014.
- [Lasaga, A. C., Soler, J. M., Ganor, J., Burch, T. E., and Nagy, K. L.: Chemical weathering rate laws and global geochemical cycles, *Geochimica Cosmochimica Acta*, 58, 2361–2386, doi:10.1016/0016-7037\(94\)90016-7, 1994.](#)
- Li, D., Bauer, S., Benisch, K., Graupner, B., and Beyer, C.: OpenGeoSys-ChemApp: a coupled simulator for reactive transport in multiphase systems and application to CO₂ storage formation in Northern Germany, *Acta. Geotech.*, 9, 67–79, 2014.
- Lichtner, P. C. and Hammond, G. E.: Using high performance computing to understand roles of labile and nonlabile uranium(VI) on Hanford 300 Area Plume Longevity, *Vadose Zone J.*, 11, doi:10.2136/vzj2011.0097, 2012.
- Lichtner, P. C., Hammond, G. E., Lu, C., Karra, S., Bisht, G., Andre, B., Mills, R. T., and Kumar, J.: PFLOTRAN User Manual: A Massively Parallel Reactive Flow and Transport Model for Describing

- Surface and Subsurface Processes, available at: http://www.pflotran.org/docs/user_manual.pdf, last access: 2 March 2015.
- 5 Mayer, K. U., Frind, E. O., and Blowes, D. W.: Multicomponent reactive transport modeling in variably saturated porous media using a generalized formulation for kinetically controlled reactions, *Water Resour. Res.*, 38, 1174, doi:10.1029/2001WR000862, 2002.
- Meeussen, J. C. L.: ORCHESTRA: an object-oriented framework for implementing chemical equilibrium models, *Environ. Sci. Technol.*, 37, 1175–1182, 2003.
- 10 Molins, S., Mayer, K. U., Amos, R. T., and Bekins, B. A.: Vadose zone attenuation of organic compounds at a crude oil spill site – interactions between biogeochemical reactions and multicomponent gas transport, *J. Contam. Hydrol.*, 112, 15–29, 2010.
- Morway, E. D., Niswonger, R. G., Langevin, C. D., Bailey, R. T., and Healy, R. W.: Modeling variably saturated subsurface solute transport with MODFLOW-UZF and MT3DMS, *Ground Water*, 51, 237–251, 2013.
- 15 Nardi, A., Idiart, A., Trinchero, P., de Vries, L. M., and Molinero, J.: Interface COMSOL-PHREEQC (iCP), an efficient numerical framework for the solution of coupled multiphysics and geochemistry, *Comput. Geosci.*, 69, 10–21, 2014.
- Nasir, O., Fall, M., and Evgin, E.: A simulator for modeling of porosity and permeability changes in near field sedimentary host rocks for nuclear waste under climate change influences, *Tunn. Undergr. Sp. Tech.*, 42, 122–135, 2014.
- 20 Palandri, J. L. and Kharaka, Y. K.: A compilation of rate parameters of water-mineral interaction kinetics for application to geochemical modelling, US Geological Survey Water-Resources Investigations Report 04–1068, 2004.
- Parkhurst, D. L. and Appelo, C. A. J.: User's guide to PHREEQC (Version 2) – a computer program for speciation, batch-reaction, one-dimensional transport and inverse geochemical calculations, US Geological Survey Water-Resources Investigations Report, 99–4259, 312 pp., 1999.
- 25 Parkhurst, D. L. and Appelo, C. A. J.: Description of input and examples for PHREEQC version 3 – a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations, in: US Geological Survey Techniques and Methods, book 6, chap. A43, 497 pp., 2013.
- 30 Pau, G. S. H., Bell, J. B., Pruess, K., Almgren, A. S., Lijewski, M. J., and Zhang, K. N.: High-resolution simulation and characterization of density-driven flow in CO₂ storage in saline aquifers, *Adv. Water Resour.*, 33, 443–455, 2010.

- Riley, W. J., Maggi, F., Kleber, M., Torn, M. S., Tang, J. Y., Dwivedi, D., and Guerry, N.: Long residence times of rapidly decomposable soil organic matter: application of a multi-phase, multi-component, and vertically resolved model (BAMS1) to soil carbon dynamics, *Geosci. Model Dev.*, 7, 1335–1355, doi:10.5194/gmd-7-1335-2014, 2014.
- 5 Shao, H. B., Dmytrieva, S. V., Kolditz, O., Kulik, D. A., Pfingsten, W., and Kosakowski, G.: Modeling reactive transport in non-ideal aqueous-solid solution system, *Appl. Geochem.*, 24, 1287–1300, 2009.
- Šimůnek, J., Jacques, D., van Genuchten, M. T., and Mallants, D.: Multicomponent geochemical transport modeling using HYDRUS-1D and HP1, *J. Am. Water Resour. As.*, 42, 1537–1547, 2006.
- 10 Šimůnek, J., Jacques, D., and van Genuchten, M. T.: The HP2 program for HYDRUS (2D/3D). A coupled code for simulating two-dimensional variably saturated water flow, heat transport, solute transport and biogeochemistry in porous media (HYDRUS+PHREEQC+2D), available at: http://www.pc-progress.com/Documents/HYDRUS3D_HP2_Manual.pdf, last access: 9 April 2015.
- 15 Steefel, C. I., Appelo, C. A. J., Arora, B., Jacques, D., Kalbacher, T., Kolditz, O., Lagneau, V., Lichtner, P. C., Mayer, K. U., Meeussen, J. C. L., Molins, S., Moulton, D., Shao, H., Šimůnek, J., Spycher, N., Yabusaki, S. B., and Yeh, G. T.: Reactive transport codes for subsurface environmental simulation, *Comput. Geosci.*, 1–34, doi:10.1007/s10596-014-9443-x, 2014.
- van Breukelen, B. M., Hunkeler, D., and Volkering, F.: Quantification of sequential chlorinated ethene degradation by use of a reactive transport model incorporating isotope fractionation, *Environ. Sci. Technol.*, 39, 4189–4197, 2005.
- van der Lee, J., De Windt, L., Lagneau, V., and Goblet, P.: Module-oriented modeling of reactive transport with HYTEC, *Comput. Geosci.*, 29, 265–275, 2003.
- Wang, W., Fischer, T., Zehner, B., Böttcher, N., Görke, U.-J., and Kolditz, O.: A parallel finite element method for two-phase flow processes in porous media: OpenGeoSys with PETSc, *Environ. Earth Sci.*, 73, 2269–2285, doi:10.1007/s12665-014-3576-z, 2014.
- 25 Wang, W.-Q., Kosakowski, G., and Kolditz, O.: A parallel finite element scheme for thermo-hydro-mechanical (THM) coupled problems in porous media, *Comput. Geosci.*, 35, 1631–1641, 2009.
- Wang, W.-Q., Rutqvist, J., Görke, U.-J., Birkholzer, J., Schnicke, T., and Kolditz, O.: **Non-isothermal flow in low permeable porous media: a comparison of Richards' and two-phase flow approaches;** *Environ. Earth Sci.*, 62, 1197–1207 Parallel finite element method and time stepping control for non-isothermal poro-elastic problem, CMC-COMPUTERS MATERIALS&CONTINUA, 21, 217–235, 2011.
- 30

- Wissmeier, L. and Barry, D. A.: Simulation tool for variably saturated flow with comprehensive geochemical reactions in two- and three-dimensional domains, *Environ. Modell. Softw.*, 26, 210–218, 2011.
- 5 Xie, M. L., Bauer, S., Kolditz, O., Nowak, T., and Shao, H.: Numerical simulation of reactive processes in an experiment with partially saturated bentonite, *J. Contam. Hydrol.*, 83, 122–147, 2006.
- Xu, T. F. and Pruess, K.: Modeling multiphase non-isothermal fluid flow and reactive geochemical transport in variably saturated fractured rocks: 1. Methodology, *Am. J. Sci.*, 301, 16–33, 2001.
- Xu, T. F., Apps, J. A., and Pruess, K.: Numerical simulation of CO₂ disposal by mineral trapping in deep aquifers, *Appl. Geochem.*, 19, 917–936, 2004.
- 10 Xu, T. F., Sonnenthal, E., Spycher, N., and Pruess, K.: TOUGHREACT – a simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: applications to geothermal injectivity and CO₂ geological sequestration, *Comput. Geosci.*, 32, 145–165, 2006.
- 15 Xu, T. F., Spycher, N., Sonnenthal, E., Zhang, G. X., Zheng, L. E., and Pruess, K.: TOUGHREACT version 2.0: a simulator for subsurface reactive transport under non-isothermal multiphase flow conditions, *Comput. Geosci.*, 37, 763–774, 2011.
- Yabusaki, S. B., Fang, Y. L., Williams, K. H., Murray, C. J., Ward, A. L., Dayvault, R. D., Waichler, S. R., Newcomer, D. R., Spane, F. A., and Long, P. E.: Variably saturated flow and multicomponent biogeochemical reactive transport modeling of a uranium bioremediation field experiment, *J. Contam. Hydrol.*, 126, 271–290, 2011.
- 20 Yeh, G. T. and Tripathi, V. S.: *HYDROGEOCHEM: A Coupled Model of HYDROlogical Transport and GEOCHEMical Equilibrium of Multi Component Systems*, edited by: Laboratory, O. R. N., 1990.
- [Yeh, G. T. and Tripathi, V. S.: A model for simulating transport of reactive multispecies components: Model development and demonstration, *Water Resour. Res.*, 27, 3075–3094, 1991.](#)
- 25 Zhang, K., Wu, Y. S., and Pruess, K.: Users guide for TOUGH2-MP – a massively parallel version of the TOUGH2 code, Report LBNL-315E, Lawrence Berkeley National Laboratory, Berkeley, 108 pp., 2008.

Table 1. Parameters for dolomite kinetics (from Palandri and Kharaka (2004)).

<u>Parameter</u>	<u>Value</u>	<u>Unit</u>
<u>A</u>	<u>0.001</u>	<u>m^2/kg</u>
<u>θ</u>	<u>1.0</u>	<u>-</u>
<u>η</u>	<u>1.0</u>	<u>-</u>
<u>E_a (neutral)</u>	<u>52200</u>	<u>J/mol</u>
<u>$\log(K_{25})$ (neutral)</u>	<u>-7.53</u>	<u>$\text{mol}/\text{m}^2/\text{s}$</u>
<u>E_a (acid)</u>	<u>36100</u>	<u>J/mol</u>
<u>$\log(K_{25})$ (acid)</u>	<u>-3.19</u>	<u>$\text{mol}/\text{m}^2/\text{s}$</u>
<u>species (acid)</u>	<u>H^+</u>	<u>-</u>
<u>β</u>	<u>0.5</u>	<u>-</u>

Table 2. Material properties of the 1-D calcite column.

Parameter	Value	Unit
Effective porosity	0.32	–
Bulk density	1.80×10^3	kg m^{-3}
Longitudinal dispersivity	6.70×10^{-2}	m
Flow rate	3.00×10^{-6}	m s^{-1}
<u>Temperature</u>	<u>298.15</u>	<u>K</u>

Table 3. Initial and boundary conditions for the Engesgaard benchmark.

Species	Initial conditions	Boundary conditions	Unit
Ca ²⁺	1.23×10^{-1}	1.00×10^{-7}	mol m ⁻³
Mg ²⁺	1.00×10^{-9}	1.00	mol m ⁻³
C(4)	1.23×10^{-1}	1.00×10^{-7}	mol m ⁻³
Cl ⁻	1.00×10^{-9}	2.00	mol m ⁻³
pH	9.91	7	–
pe	4	4	–
Calcite	5.7412×10^{-2}	–	mol m ⁻³
Dolomite	0.0	–	mol m ⁻³

Table 4. An overview of different portions of the simulation time for the Engesgaard benchmark by using different codes (in seconds).

<u>Codes</u>	<u>Flow and Mass transport</u>	<u>Chemistry and interface</u>	<u>Total</u>
<u>OGS#IPhreeqc</u>	<u>0.047</u>	<u>7.814</u>	<u>7.861</u>
<u>Phreeqc</u>	<u>-</u>	<u>-</u>	<u>5.74</u>
<u>OGS-ChemApp</u>	<u>0.183</u>	<u>23.467</u>	<u>23.65</u>

Table 5. An overview of different portions of the simulation time for the van Breukelen benchmark by using different codes (in seconds).

<u>Code</u>	<u>Flow and Mass transport</u>	<u>Chemistry and interface</u>	<u>Total</u>
<u>OGS#IPhreeqc</u>	<u>0.453</u>	<u>32.218</u>	<u>32.671</u>
<u>PHREEQC</u>	<u>-</u>	<u>-</u>	<u>14.196</u>
<u>KinReact</u>	<u>0.453</u>	<u>0.969</u>	<u>1.389</u>

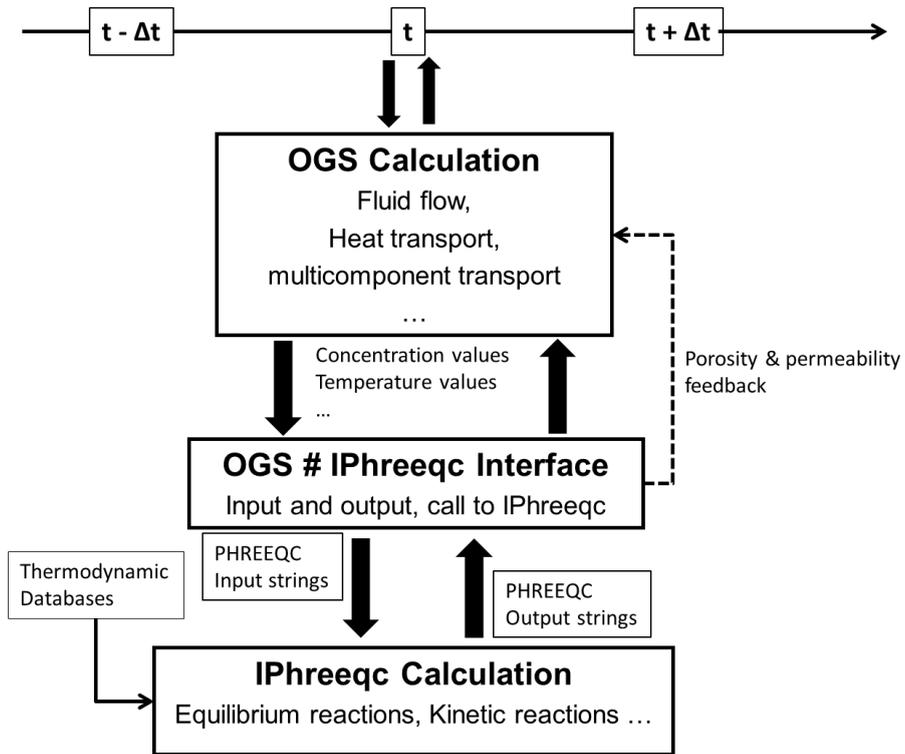


Figure 1. General concept of the coupling interface between OGS and IPhreeqc.

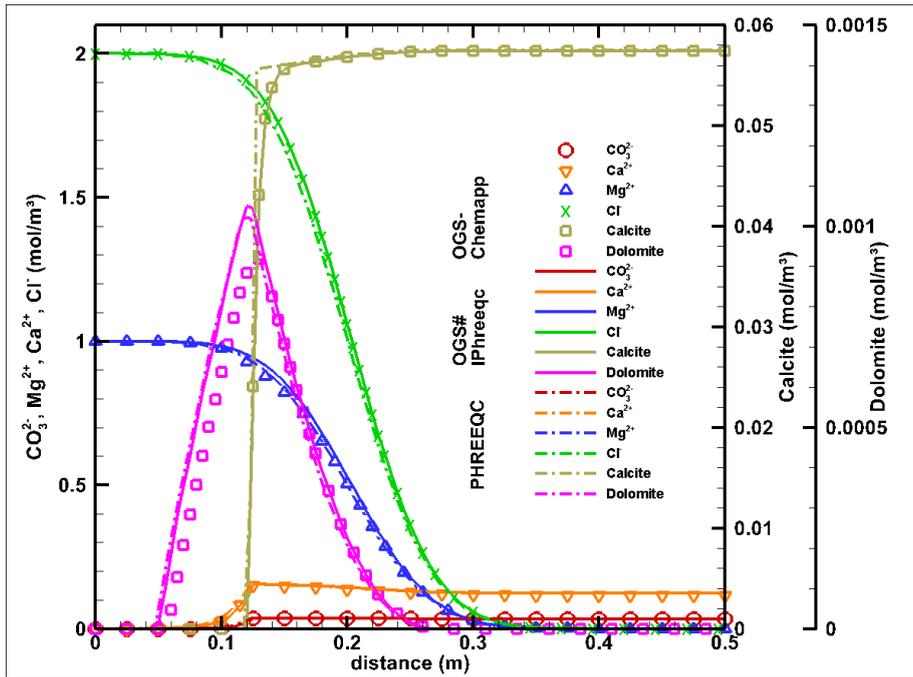


Figure 2. Comparison of calcite and dolomite precipitation/dissolution simulation with [OGS-Chemapp](#) and [OGS-ChemApp](#), [OGS#IPhreeqc](#) and [PHREEQC](#).

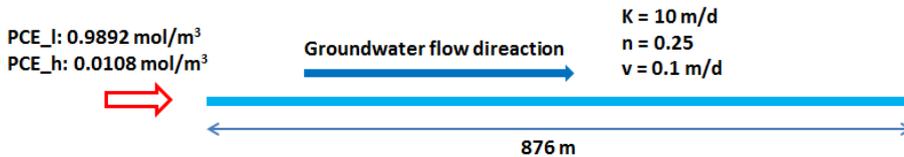


Figure 3. Model domain, material properties, initial and boundary conditions of the isotope fractionation benchmark. K , n and v denotes hydraulic conductivity, porosity and groundwater velocity of the aquifer, respectively (basic units are: m – meter, d – days).

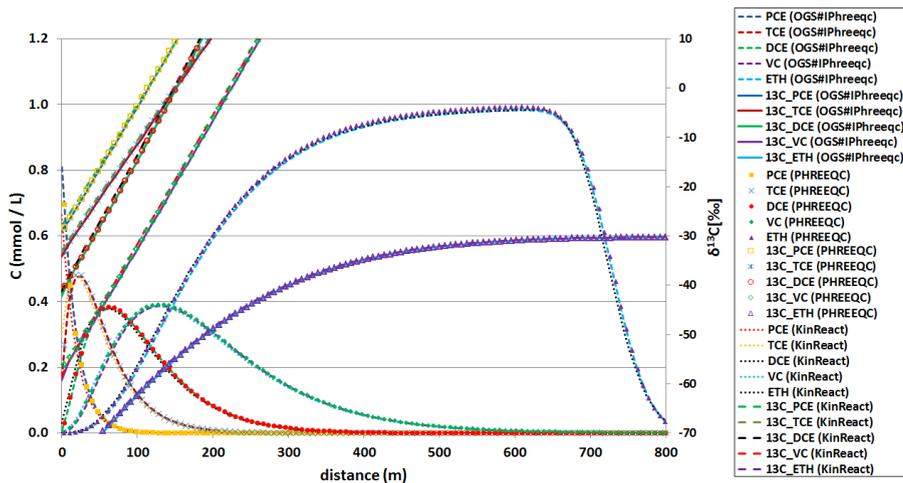


Figure 4. Concentration profiles of the light CHC isotopologues and $\delta^{13}\text{C}$ [%] isotope signatures along the horizontal axis of the model domain simulated by OGS#IPhreeqc (dashed lines or full lines) and PHREEQC (symbols) at the end of the simulations after 20 years.

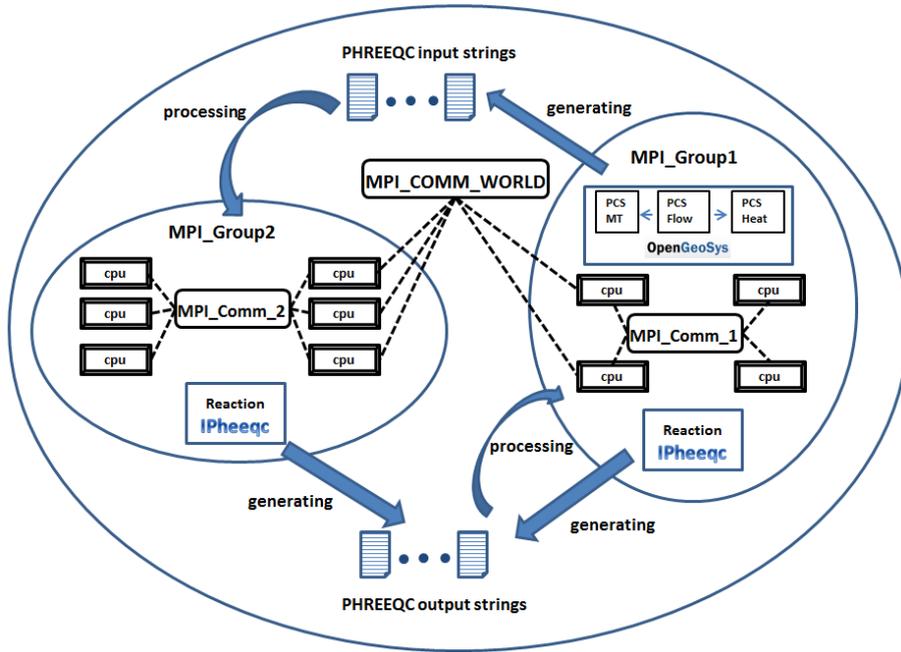


Figure 5. Parallelization scheme for OGS#IPhreeqc. Two distinct MPI groups and relevant inter- and intra-communicators are created. MPI_Group1 take part in the simulation of both DDC related processes and chemical reactions, while MPI_Group2 only participates in the simulation of chemical reactions. PCS MT, PCS Flow and PCS Heat are process of mass transport, flow and heat transport, respectively.

```

if (myrank_group1 != MPI_UNDEFINED)//ranks of MPI_Group1 will run following code
{
    read OGS input data;
    while (the current time is smaller than the end time) //time stepping loop
    {
        compute flow processes;
        compute heat and mass transport process;
        prepare the input strings for PHREEQC;
        send start signal to MPI_Group2;
        //inform MPI_Group2 that the input strings for IPhreeq are prepared;
        send input strings to MPI_Group2;
        calculate chemical reactions with IPhreeqc;
        if MPI_Group2 exists
        {
            receiving result strings from MPI_Group2;
            handle result strings and update data for mass transport;
        }
        print out results; //if needed
        send MPI_Group2 a kill signal;
        //inform MPI_Group2 the time stepping loop is over
        terminate MPI environment;
    }
}

if (myrank_group2 != MPI_UNDEFINED)//if ranks of MPI_Group2 exist
{
    for () //reaction loop
    {
        waiting for signal from MPI_Group1;
        if the signal is a kill signal
        {
            jump out of the reaction loop;
        }
        else
        {
            receive input strings from MPI_Group1;
            calculate chemical reactions with IPhreeqc;
            send result strings to MPI_Group1;
            //inform MPI_Group1 that calculation of reaction is done
        }
    }
    terminate MPI environment;
}
}

```

Figure 6. Pseudo code for schematic presentation of the parallelization scheme.

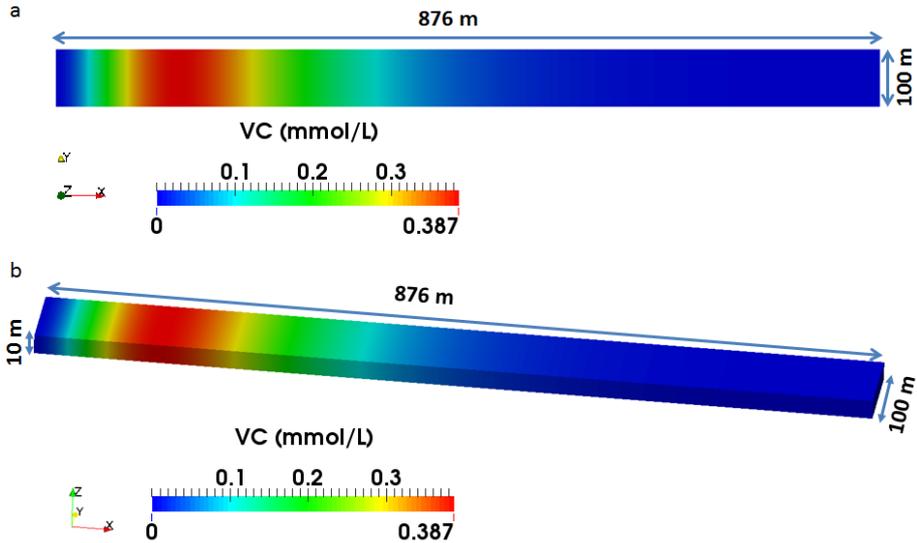


Figure 7. Concentration profile of light isotope VC of the 2-D model **(a)** and the 3-D model **(b)** at the end of the simulation. For **(b)** a vertical (z direction) exaggeration of 2 times is applied.

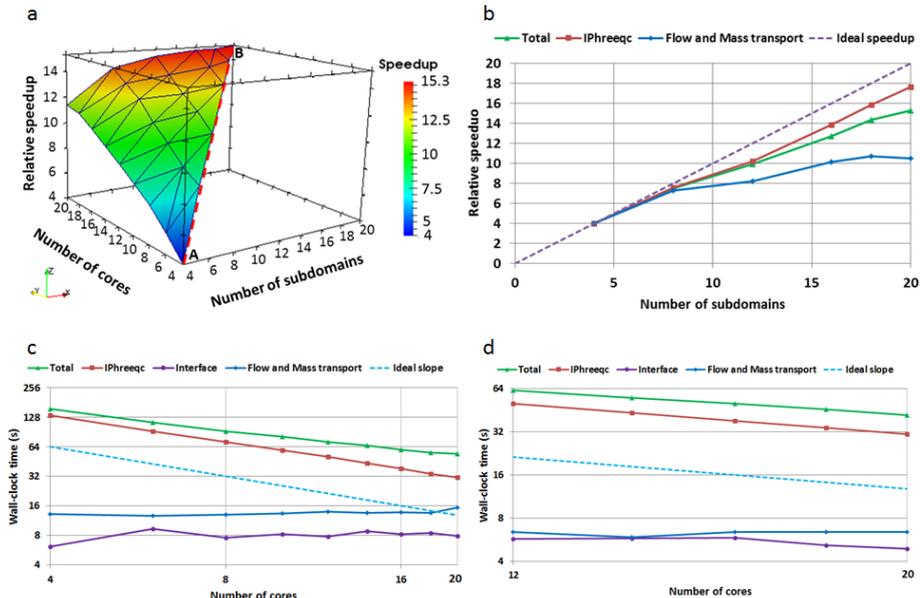


Figure 8. Performance of the proposed parallelization scheme in running isotope fractionation 2-D example on ENVINF. **(a)** Relationship between number of DDCs, number of compute cores and relative speedup in comparison to a ~~single core~~ simulation with 4 cores and 4 DDCs (Color legend shows the value of relative speedup); **(b)** breakdown of the speedup curve AB (marked as dashed line in **a**) into speedup of calculation of chemical reaction i.e. IPhreeqc and other processes flow and mass transport; **(c)** breakdown of the total time for chemical reactions, ~~OGS~~IPhreeqc-interface and “other” flow and transport for DDC = 4; **(d)** breakdown of the total time for DDC = 12.

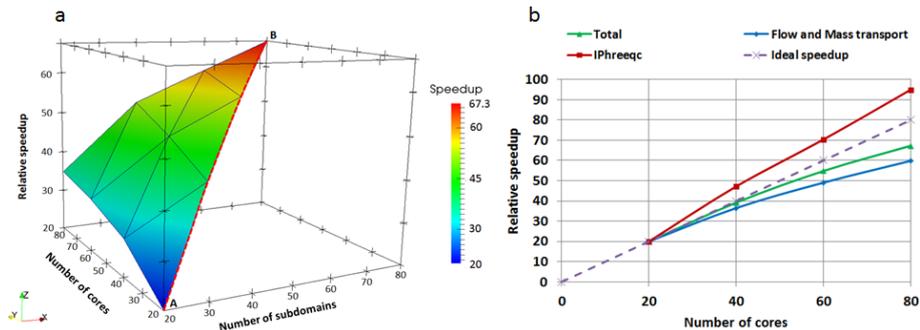


Figure 9. Performance of the parallelization scheme for the simulation of the 3-D test example on EVE cluster. **(a)** Relationship between number of DDCs, number of compute cores and relative speedup **in comparison to a 2-20 compute cores simulation on ENVINF**; **(b)** breakdown of the speedup curve AB (marked as dashed line in **a**) into speedup of calculation of chemical reaction i.e. IPhreeqc and other processes.

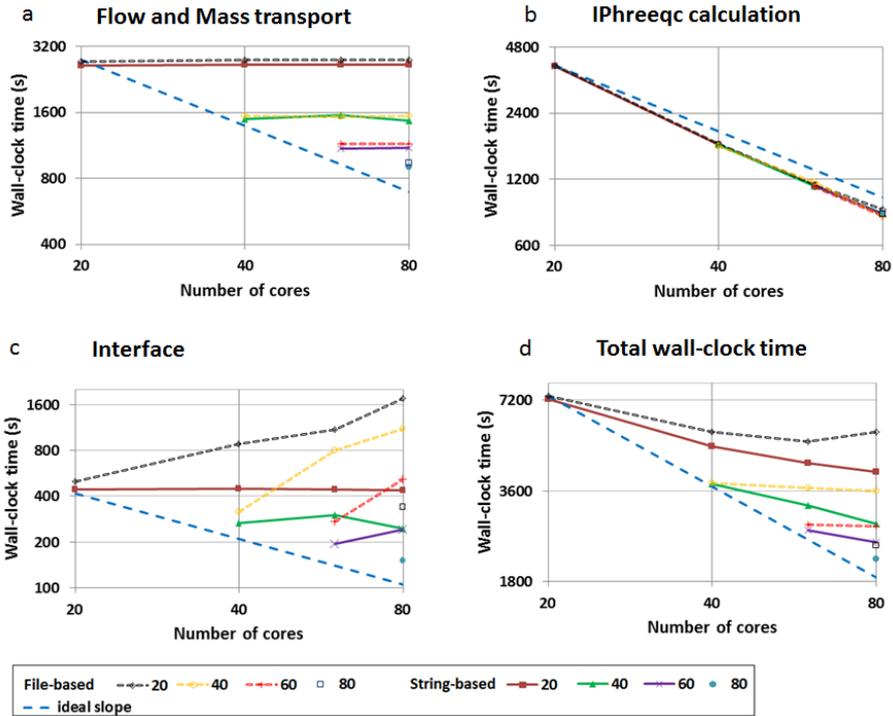


Figure 10. Breakdown of the total wall-clock time in running the 3-D test example on EVE cluster into different processes for different DDCs varying from 20 to 80. **(a)** Mass-mass transport and flow; **(b)** geochemical reaction (IPhreeqc); **(c)** writing and reading files in OGS#IPhreeqc interface; **(d)** other total wall-clock time consumption in OGS#IPhreeqc interface.

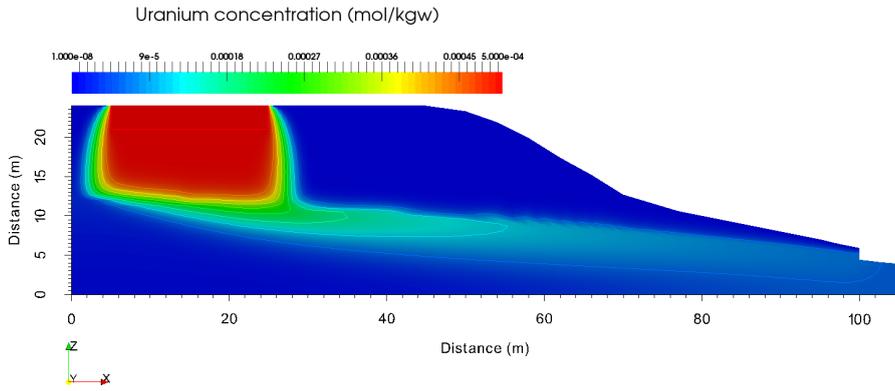


Figure 11. Uranium leaching at a hillslope scale.

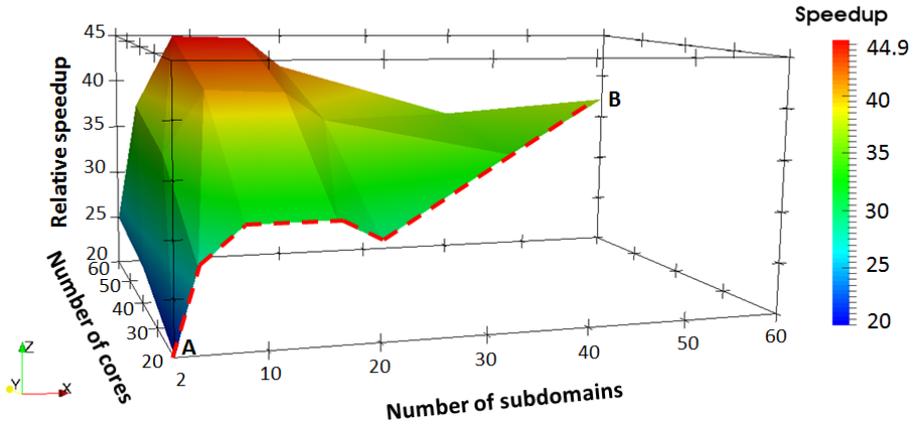


Figure 12. [Relative speedup to serial simulation as a function of number of DDCs and compute cores.](#)

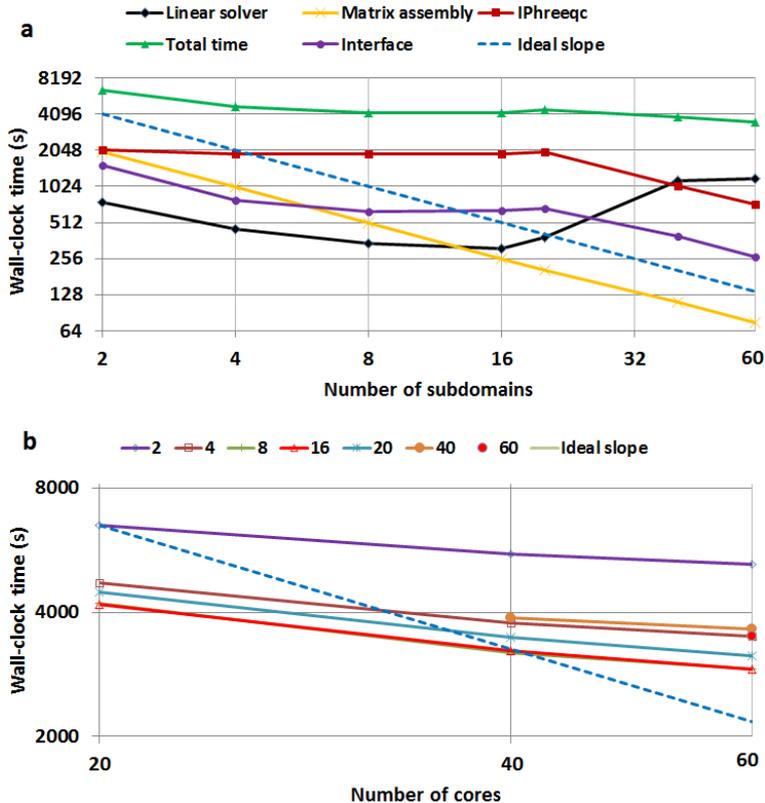


Figure 13. Analysis of the simulation time as functions of subdomains and compute cores. **(a)** breakdown of the total time corresponding to speedup curve AB in Fig. 13. 20 cores are employed for DDCs from 2 to 20, for more DDCs same number of cores and DDCs are applied; **(b)** total simulation time as a function of compute cores for different DDCs varying from 2 to 60.