



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-Universität zu Kiel, Kiel, Germany

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

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

Received: 15 January 2015 – Published in Geosci. Model Dev. Discuss.: 3 March 2015

Revised: 25 September 2015 – Accepted: 7 October 2015 – Published: 22 October 2015

Abstract. The open-source scientific software packages OpenGeoSys and IPhreeqc have been coupled to set up and simulate thermo-hydro-mechanical-chemical coupled processes with simultaneous consideration of aqueous geochemical reactions faster and easier on high-performance computers. In combination with the elaborated and extendable chemical database of IPhreeqc, it will be possible to set up a wide range of multiphysics problems with numerous chemical reactions that are 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. This technical paper presents the implementation, verification, and parallelization scheme 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, for example, in contaminated site remediation

or water resource 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 and Hammond, 2012; Molins et al., 2010; Riley et al., 2014; Yabusaki et al., 2011). Geotechnical applications employ reactive transport simulations, for example, to quantify geochemical processes in geological nuclear waste repositories (e.g., Kosakowski and Watanabe, 2014; 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).

In the last decades, much effort has been invested in developing 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), CrunchFlow (Steefel et al., 2014), MIN3P (Mayer et al., 2002) and PFLOTRAN (Lichtner et al., 2015). Since each code has its own strengths and limitations, the coupling of different codes (i.e., one program applied to another, or vice versa) is a straightforward solution to make use of combined capabilities of different codes. Existing approaches which apply tool coupling methods to simulate reactive transport processes are, for example, 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, 2014; 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-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 accompanied with other challenges such as numerical precision and stability (de Dieuleveult and Erhel, 2010; Kosakowski and Watanabe, 2014; Wissmeier and Barry, 2011) or expensive computational time.

Especially for realistic reactive transport simulations at larger scales, i.e., from field to catchment or reservoir scale, high complexities of hydrogeological and geochemical systems 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, for example for model calibration and parameter sensitivity analysis, become 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). PFLO-TRAN, a parallel multiscale and multiphysics code for subsurface multiphase flow and reactive transport (Hammond et al., 2012, 2014; Lichtner et al., 2012), and 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.

A well-designed code concept and 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 simulator (based on the finite-element method) for multidimensional thermo-hydro-mechanical-chemical (THMC) coupled processes in porous and fractured media (Kolditz et al., 2012).

In other words, OGS is able to simulate, for example, 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 modeling aqueous, mineral, gas, surface, ion exchange, solid-solution equilibria and kinetic reactions but also provides a well-defined set of methods for data transfer and management (Charlton and Parkhurst, 2011). Both codes are open-source; thus, the technical coupling can be realized directly on the code level.

The optimum quantities of the required computer resources for DDC-related processes (flow and mass transport) and chemical reactions can be quite different. 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 are bound to DDC, meaning that additional communication is needed to exchange the results along shared subdomain boundaries. Therefore a speedup for flow/transport is no longer experienced when communication and serial fractions are more time-consuming than the parallel fractions. As a consequence, whereas the computation of the chemical system can see a further speedup with the addition of more compute cores, the computation of the transport problem may already reach a point of optimization, rendering the addition of further compute cores beyond this point inefficient. If the number of compute cores for flow and transport is applied to the attached reaction system as well, then 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 amounts of computer resources for DDC-related processes and geochemical reactions and thus to allocate an optimum number of compute cores for both types of processes simultaneously. Global processes will be parallelized based on the DDC method, whereas the parallelization of geochemical reactions is completely independent of 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 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 to 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).

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 kinds of flow processes such as incompressible and compressible groundwater flow, overland flow, density-driven flow, unsaturated flow, and two-phase as well as multiphase flow. Picard and Newton–Raphson schemes can be applied to 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. 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 the 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, 2014; Centler et al., 2010; Beyer et al., 2012; Li et al., 2014).

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). Besides 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 an application programming interface (API) 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).

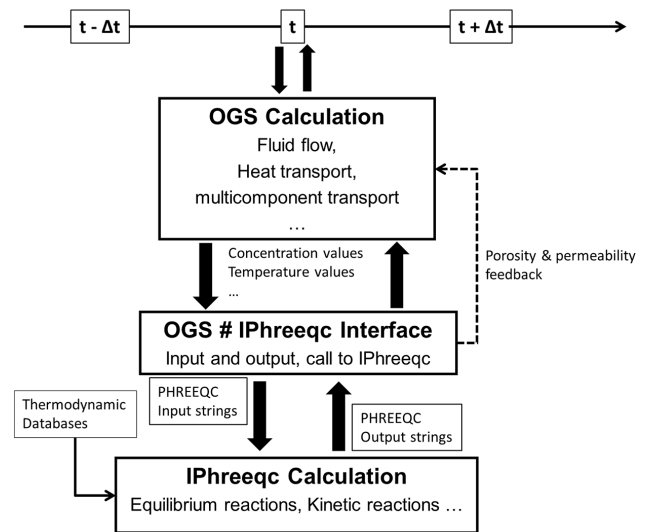


Figure 1. General concept of the coupling interface between OGS and IPhreeqc.

2.3 OGS#IPhreeqc interface

In the current study, both source codes, i.e., OGS and IPhreeqc, are statically linked to allow access for 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 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 build system. This allows users to benefit continuously from code developments of both sides.

The sequential non-iterative approach (SNIA) for operator splitting is applied in the coupling procedure, which means that no iterations are made between mass transport and geochemical reactions. Consequently, adequately 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 was 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 and master solution species.

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 the interface. An IPHreeqc input string will then be prepared which contains information on 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 same procedures will be performed as during the initialization: concentration values of each component as well as other state variables for all nodes will be forwarded to the coupling interface; an input string will be prepared, followed by a call to IPHreeqc.

A complete call to IPHreeqc will be realized by taking the following steps: (i) create a new instance of IPHreeqc, (ii) load a thermodynamic database for the geochemical system, (iii) read and run the specific PHREEQC input 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 and Parkhurst (2011) and Parkhurst and Appelo (2013).

These procedures have to be repeated during each call to IPHreeqc within each time step. However, the overhead (steps other than iii and iv) involved in the call to IPHreeqc is small compared to the total simulation time; this 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 and 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.

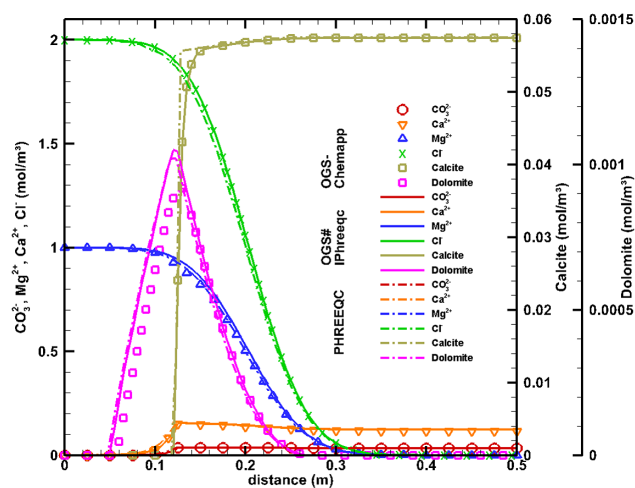


Figure 2. Comparison of calcite and dolomite precipitation/dissolution simulation with OGS-ChemApp, OGS#IPHreeqc and PHREEQC.

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 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 shows the phenomenon occurs when a 0.5 m long 1-D calcite column is flushed with a solution containing 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 is simulated as an equilibrium reaction, whereas that of dolomite is modeled as a kinetic reaction using a transition state theory (Lasaga et al., 1994). The kinetic rate parameters from Palandri and Kharaka (2004) are applied (see Table 1). The material properties of the column are given in Table 2, and the initial and boundary conditions in Table 3. The model domain is discretized into 100 uniform line elements. Total simulation time is 21 333.32 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 Supplement. A comparison of the simulation results by using the three codes is illustrated in Fig. 2. Apart from the amount of dolomite, the simulation results of OGS#IPHreeqc, PHREEQC and OGS-ChemApp (from Beyer et al., 2012) show generally good agreements,

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

Parameter	Value	Unit
A	0.001	$\text{m}^2 \text{kg}^{-1}$
θ	1.0	–
η	1.0	–
E_a (neutral)	52200	J mol^{-1}
$\log(K_{25})$ (neutral)	–7.53	$\text{mol m}^{-2} \text{s}^{-1}$
E_a (acid)	36100	J mol^{-1}
$\log(K_{25})$ (acid)	–3.19	$\text{mol m}^{-2} \text{s}^{-1}$
species (acid)	H^+	–
β	0.5	–

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

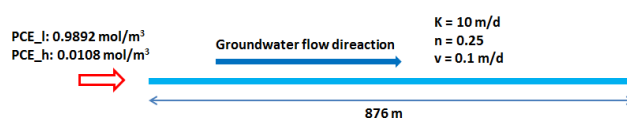
Parameter	Value	Unit
Effective porosity	0.32	–
Bulk density	1.80×10^3	kg m^{-3}
Longitudinal dispersivity	6.70×10^{-2}	m
Flow rate	3.00×10^{-6}	m s^{-1}
Temperature	298.15	K

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 (including the preparation of input for IPhreeqc and the processing of output from IPhreeqc) and the overhead involved in calling to IPhreeqc (described in Sect. 2.3) 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, and initial and boundary conditions are illustrated in Fig. 3.

The intermediate products during the degradation include tri- and dichloroethylene (TCE, DCE) and 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. In total, there are 11 chemical species, including chloride as a 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 to 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.

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

The simulated concentration profiles of the light CHC isotopes and relevant $\delta^{13}\text{C}$ [‰] isotope signatures along the model domain are compared with those simulated using a batch version of PHREEQC (version 3.2.0) and the KinReact module of OGS (Fig. 4), showing 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 that of the batch version of PHREEQC. The time spent for the interface and the overhead for calling to IPhreeqc accounts 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 like PHREEQC does (e.g., surface complexation, mineral nucleation).

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 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 Application of the DDC approach of OGS

The domain decomposition (DDC) approach is applied to partition the computational tasks of the global assembly and

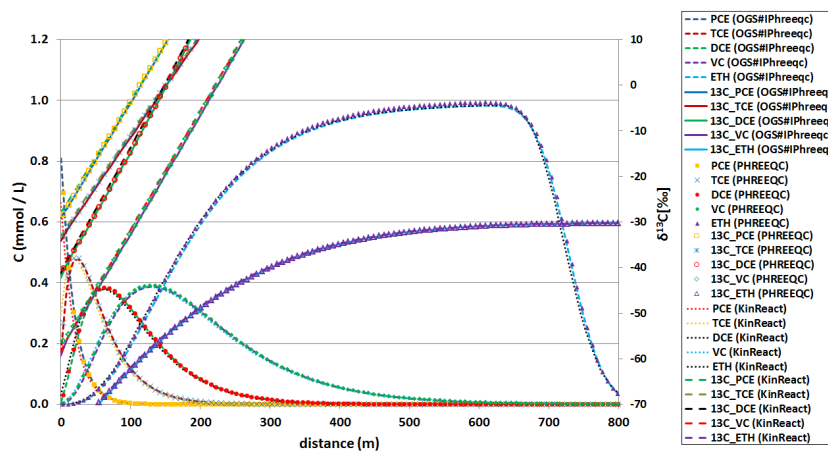


Figure 4. Concentration profiles of the light CHC isotopologues and $\delta^{13}\text{C}$ [%] isotope signatures along the horizontal axis of the model domain simulated by OGS#IPhreeqc (dashed lines or full lines) and PHREEQC (symbols) at the end of the simulations after 20 years.

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

Codes	Flow and mass transport	Chemistry and interface	Total
OGS#IPhreeqc	0.047	7.814	7.861
Phreeqc	–	–	5.74
OGS-ChemApp	0.183	23.467	23.65

the linear solver implemented in OGS (Wang et al., 2009). For the current DDC approach, METIS is used as a preprocessing tool to partition mesh in order to balance the node quantities and minimize the border nodes among subdomains efficiently. With the partitioned mesh data, the stiffness matrix and the right-hand side vector of the system of linear equations are only assembled within subdomains by individual compute cores. Then these assembled subdomain matrices and vectors are taken to compute a converged solution with iterative solvers. This way, the computational tasks of the global assembly and the linear solver are parallelized in a straightforward manner. More detailed information of DDC procedures can be found in previous works by Kalbacher et al. (2008) and Wang et al. (2009).

3.2 Parallelization scheme

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 the code related to geochemical simulation.

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){...}
```

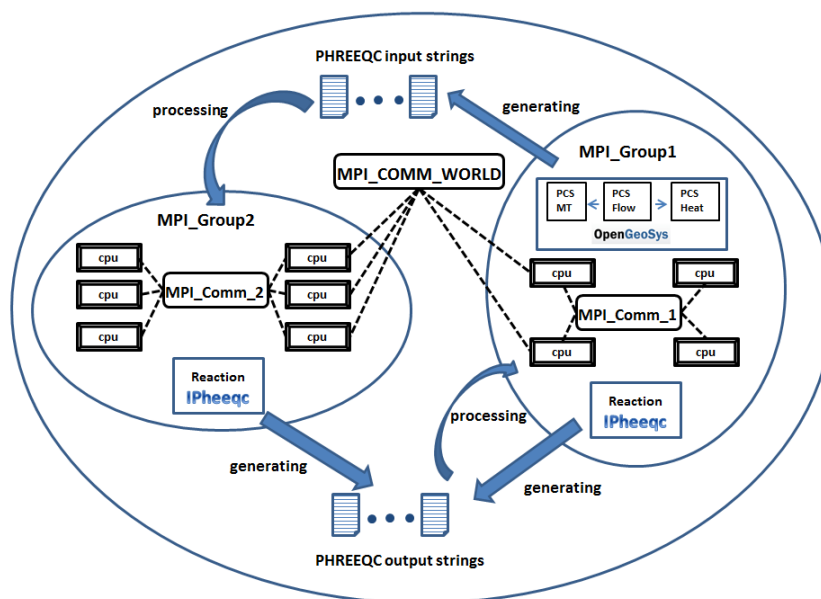
For each MPI operation in the entire code, it is important to identify the relevant MPI group and choose the correct MPI communicator.

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 flow and mass transport process, PHREEQC input strings for all compute cores will be created by compute cores of MPI_Group1. A big difference between the serial and parallel algorithm should be noticed here. In a serial simulation, only one input string will be prepared for all finite-element nodes 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. Each string carries the information for the nodes being solved on a specific compute core.

After the preparation of input strings, compute cores of MPI_Group1 will send 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.

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

Code	Flow and mass transport	Chemistry and interface	Total
OGS#IPhreeqc	0.453	32.218	32.671
PHREEQC	–	–	14.196
KinReact	0.453	0.969	1.389

**Figure 5.** Parallelization scheme for OGS#IPhreeqc. Two distinct MPI groups and relevant inter- and intra-communicators are created. MPI_Group1 takes 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.

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 the 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, 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. In special cases, when the number of subdomains equals that of the compute cores, only MPI_Group1 will be created. In this case, no communication between the two MPI groups is required.

As mentioned above, a character-string-based data transfer is applied to exchange concentration values between mass transport and geochemical reaction simulations. 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 of 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 perform a complete call to `IPhreeqc` by using a specific input string (including all the `IPhreeqc` tasks mentioned in Sect. 2.3). A relevant PHREEQC results 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 handle all the result 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_Allreduce` method (this is a straightforward solution for the current implementation; a more sophisticated approach, however, should be implemented to minimize the

```

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 IPhreeqc 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.

inter-processor communication and memory usage), before the updated concentrations of different components are sent back to the 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 multicore Linux machine called “ENVINF”. It contains 40 Intel® Xeon® 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 a 40 Gbits⁻¹ QDR Infiniband network interconnect. The peak performance is 10 TFLOPs⁻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. Jobs can also be sub-

mitted without this constraint; 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 agreement with serial simulation results.

4 Performance tests and analysis

In this section, the performance of the parallelization scheme is tested by using three examples differing by dimension and problem size. The first two examples are simple extensions of the 1-D benchmark of isotope fractionation. However,

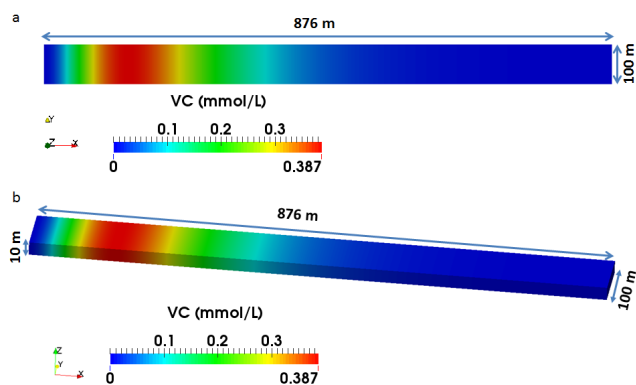


Figure 7. Concentration profile of light isotope VC of the 2-D model (a) and the 3-D model (b) at the end of the simulation. For (b) a 2-fold vertical (z direction) exaggeration is applied.

they differ from each other in 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 nonlinear 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 ($876\text{ m} \times 100\text{ m}$, see Fig. 7a). The finite-element mesh consists of 1331 nodes and 1200 uniform rectangular elements (120×10). 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. The 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.

The performance of the current parallelization scheme is demonstrated in Fig. 8. In Fig. 8a the relative speedup in comparison to a simulation with four cores and four DDCs is illustrated as a function of the number of DDCs and total compute cores. If we fix the number of DDCs at a specific value and change the total number of compute cores from 4 to 20, we can observe a continuous increase in relative speedup for all DDCs with the growth of the number of compute cores. The speedup of $\text{DDC} = 8$ is generally much better than that of $\text{DDC} = 4$. 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”) together with the 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 flow and mass transport reaches its maximum when 18 DDCs are applied. As shown by Wang et al. (2009), the adding of subdomains will increase communication between subdomain border nodes.

In this example, the parallel efficiency for solving flow and mass transport begins to degrade as soon as more than eight DDCs are employed, for which the border nodes only account for around 6 % of the total nodes. A further increase in the number of DDCs up to 20, yielding 17 % of border nodes, decreases the parallel efficiency down to 0.5 almost linearly. The speedup of reaction, however, is generally much better and increases continuously as more compute cores are provided. In the operator-splitting approach, chemical reactions are solved locally on each finite-element node; hence, no direct communication among different nodes is necessary.

Figure 8c and d show the breakdown of the total time for different compute cores with $\text{DDC} = 4$ and $\text{DDC} = 12$. It is clearly shown that the chemical reaction is the most time-consuming part of the simulation in both cases. With $\text{DDC} = 4$, reactions take up to 86.5 % of the total time when only 4 compute cores are applied, and drops to 57.2 % if 20 compute cores are applied, whereas for $\text{DDC} = 12$ it becomes 80.5 % of the total time for 12 compute cores, and goes down to 73.1 % for 20 compute cores. In both cases time 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 strings for IPHREEQC, communication among different compute cores, and handling output strings from IPHREEQC. On average, this part of time accounts for 5.2 and 10.8 % of the total simulation time for $\text{DDC} = 4$ and $\text{DDC} = 12$, respectively.

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 two compute cores with two 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 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 in the compute cores can be observed. With the use of 80 compute cores, simulation time can be reduced to around 37 min. As problem size increases, the speedup effects of both DDC-related processes and 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 (super-linear) than flow and mass transport.

However, if we fix the number of DDCs at a specific value and increase the total compute cores further, the resulting speedup is not so significant, especially for fewer DDCs (see

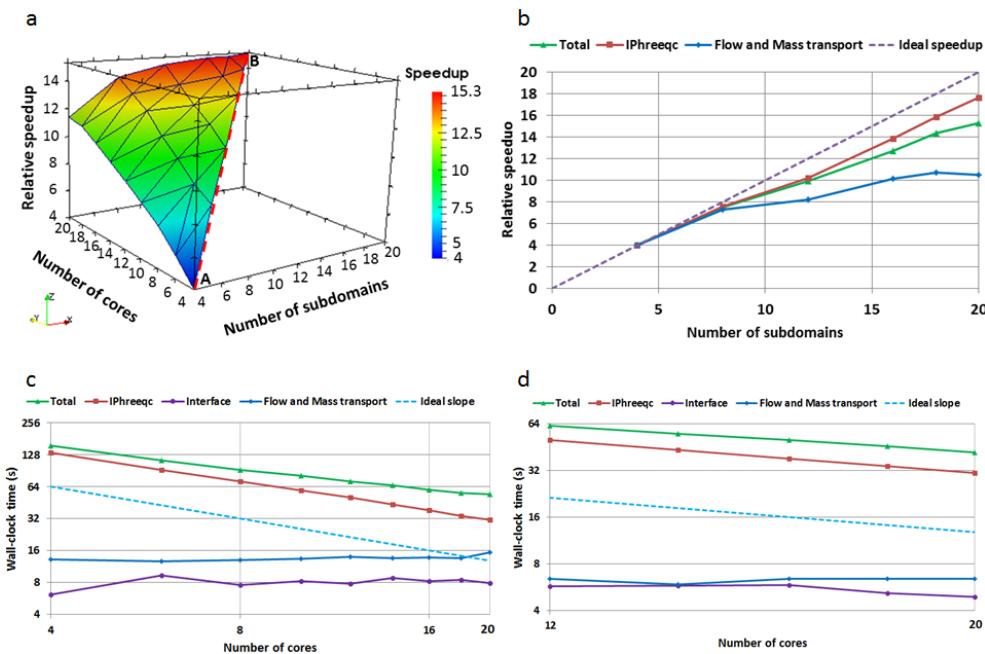


Figure 8. Performance of the proposed parallelization scheme in running isotope fractionation 2-D example on ENVINF. (a) Relationship between number of DDCs, number of compute cores and relative speedup in comparison to a simulation with four cores and four DDCs (color legend shows the value of relative 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 flow and mass transport. (c) Breakdown of the total time for chemical reactions, interface and flow and transport for DDC = 4. (d) Breakdown of the total time for DDC = 12.

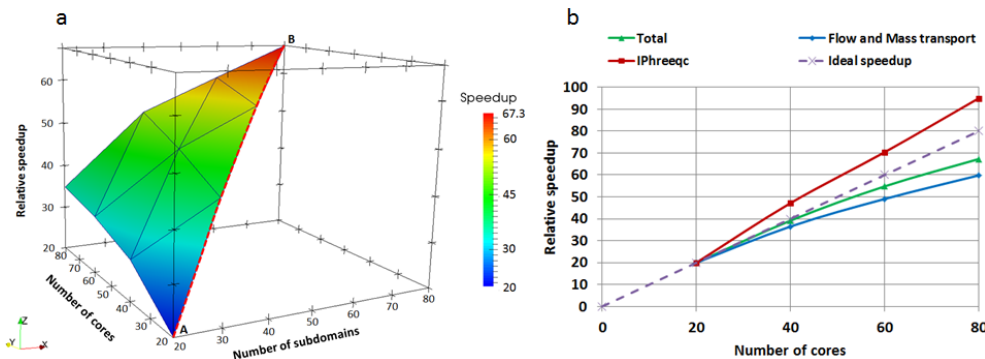


Figure 9. Performance of the parallelization scheme for the simulation of the 3-D test example on EVE cluster. (a) Relationship between number of DDCs, number of compute cores and relative speedup to 20 compute cores. (b) Breakdown of the speedup curve AB (marked as dashed line in a) into speedup of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

Fig. 9a). This behavior is somewhat different from what we have observed in the 2-D example.

The reason behind this lies mainly in the fact that the ratios 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 compared with that of reactions. In the 3-D example, the time consumption for flow and mass transport is of similar magnitude to that of reactions (see Fig. 10a and b). For 20 compute cores with 20 DDCs, flow and mass transport together take 36.2 % of the total time, whereas for

IPhreeqc calculation this 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.

Figure 10 compares the total time and its breakdowns by using string- and file-based parallelization approaches for this problem. From Fig. 10a and b 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 significant advan-

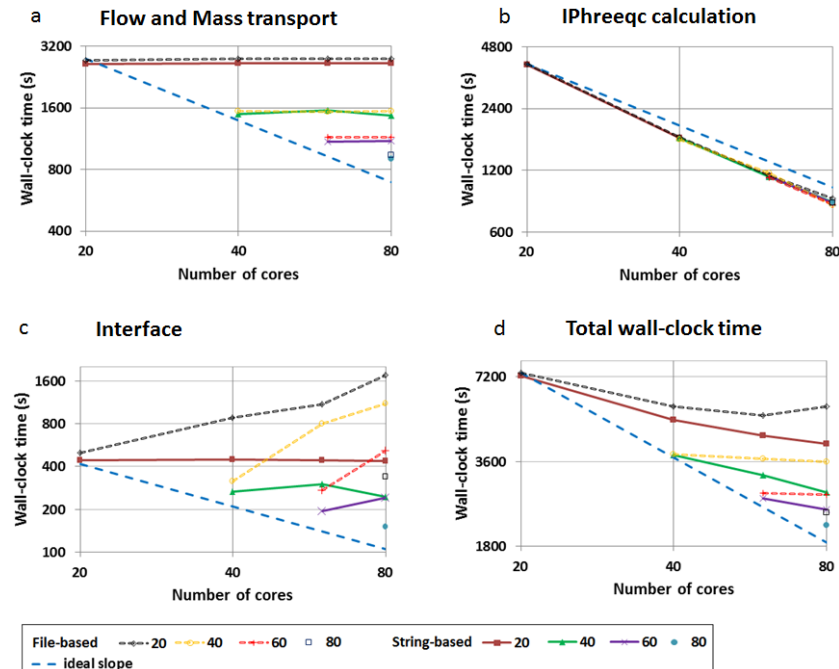


Figure 10. Breakdown of the total wall-clock time in running the 3-D test example on EVE cluster into different processes for different DDCs varying from 20 to 80. (a) Mass transport and flow, (b) geochemical reaction (IPhreeqc), (c) OGS#IPhreeqc interface, and (d) total wall-clock time.

tages over the file-based one, in which the file reading and writing is realized through the general parallel file system (GPFS). With the use of string-based data exchange, this part of time is small compared to the calculation of mass transport or chemistry. In the 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). This generally 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.

Figure 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 closer to the ideal slope) when the number of compute cores and DDCs stays 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.

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 mill tailings at a hillslope scale (see Fig. 11). The substitution of calcite for gypsum also occurs

with the release of acid and sulfate from the tailings. It is worth mentioning that redox reactions are not taken into account in this example. The water flow in both the unsaturated and saturated zone is modeled. In total, 35 species and 14 minerals are considered for geochemical reactions. A detailed description of model setup and the simulation results is available in the Supplement (Part 2).

The 2-D domain consists of 14 648 triangle elements with 7522 nodes. The total simulation time of 1000 days is discretized into 6369 time steps varying from 1×10^{-7} to 24 000 s. The same time discretization is adopted for all parallel simulations introduced below. The wall-clock time for a simulation of this example with two cores and two 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. Figure 12 illustrates relative speedups compared to the simulation with 20 cores and 2 DDCs as a function of compute cores and DDCs. The best speedups are achieved by using 60 cores and DDCs ranging between 8 and 16. With the use of 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 the linear solver for flow and mass transport. Figure 13a shows the breakdown of the total time corresponding to speedup curve AB in Fig. 12. Major components such as IPhreeqc, the linear solver and the interface are illustrated. The time for the linear solver increases

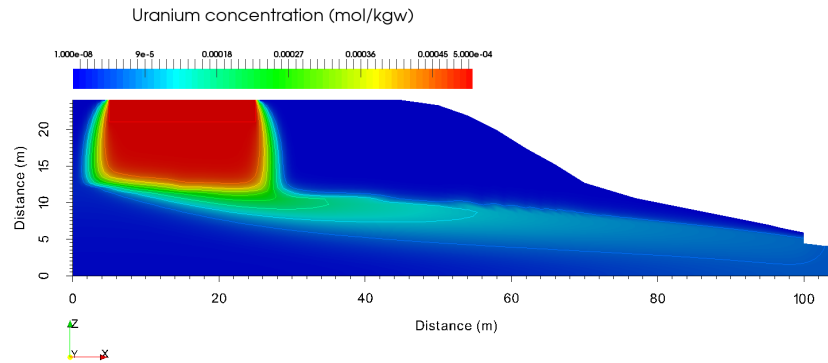


Figure 11. Uranium leaching at a hillslope scale.

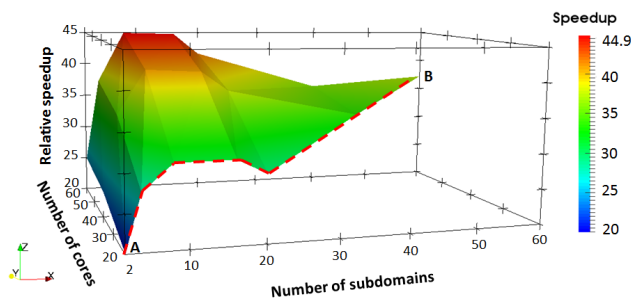


Figure 12. Relative speedup to serial simulation as a function of the number of DDCs and compute cores.

dramatically after 20 DDCs. With 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 consumption for IPhreeqc, the interface and the matrix assembly decreases further. Because 20 cores are applied for all the DDCs varying from 2 to 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 a limited number of compute cores is responsible for preparing and processing large number of input and output strings (the number of cores is 1 order of magnitude or larger than the number of DDCs). When 20 cores with only 2 DDCs are applied, it takes up to 23.4 % of the total time.

Figure 13b presents the total time for different DDCs as a function of 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 the present parallelization scheme over the conventional DDC approach, which keeps the number of cores equal to 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.

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, and can greatly benefit from the wide range of geochemical capabilities and customizable database from PHREEQC. On the basis of a sustainable way of coupling, 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. In particular, 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 computational resources for different computational tasks by using the MPI grouping concept. The appropriate setting of DDCs and total compute cores is problem-dependent.

If the time consumption for flow and mass transport is of the same magnitude as geochemical reactions, and a continuous speedup can be obtained (with the compute cores that are available) for the calculation of flow and mass transport, 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 the time spent for flow and solute transport becomes more dominant.

If a problem is dominated by geochemical reactions (e.g., for small- to medium-sized problems with complex geochemical systems), then the new approach (creating two MPI groups) can be advantageous, especially when a further in-

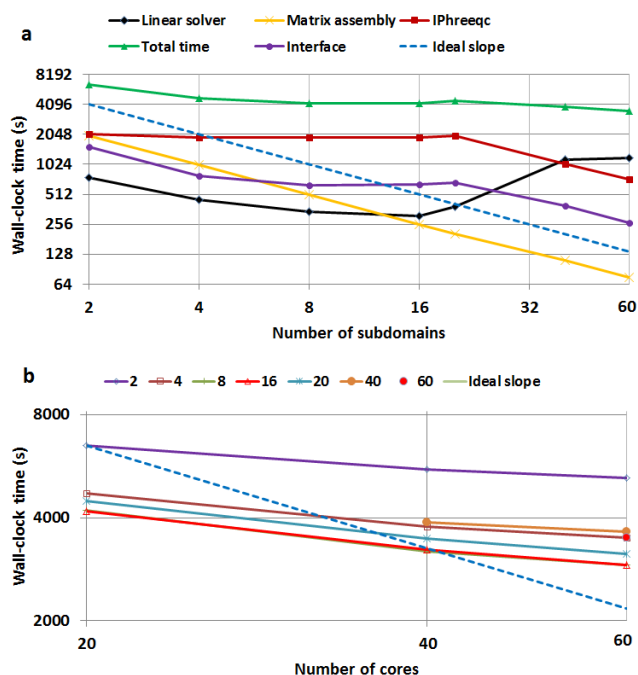


Figure 13. Analysis of the simulation time as functions of subdomains and compute cores. **(a)** Breakdown of the total time corresponding to speedup curve AB in Fig. 13. Twenty cores are employed for DDCs from 2 to 20; for more DDCs, the same number of cores and DDCs are applied. **(b)** Total simulation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

crease in the number of DDCs above the optimum will lead to a strong degradation of parallel performance for flow or mass transport. In this case, better speedups may still be obtained by fixing the number of DDCs at the optimum while allocating more compute cores for the second MPI group to accelerate the calculation of chemical reactions.

Even though the 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 scalable for a 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 in 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 re-

actions, like in heterogeneous problems with sharp transient reactive fronts or reaction hot spots, could affect the parallel performance as well. Hence, more intelligent ways to ensure efficient load balance still remain an important task.

In the current study, the available computational resources were 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 different scales.

Code availability

The source code for the serial version of OGS#IPhreeqc (file-based) was released as an official version of OGS 5.5.7. The latest release 5.6.0 including this feature can be obtained with the following link under a modified BSD License: <https://github.com/ufz/ogs5>.

Relevant information for OGS compilation can also be found there. To use the interface, the option OGS_FEM_IPQC in CMake configuration should be selected. The source code of the fully parallel version (string-based) is currently under review for the next official OGS release and can be obtained in the meantime under the same license by simply contacting the corresponding author.

The Supplement related to this article is available online at doi:10.5194/gmd-8-3333-2015-supplement.

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 support for the coupling procedure and Ben Langenberg for help in running simulations on the EVE cluster.

The article processing charges for this open-access publication were covered by a Research Centre of the Helmholtz Association.

Edited by: H. McMillan

References

- Bailey, R. T., Morway, E. D., Niswonger, R. G., and Gates, T. K.: Modeling variably saturated multispecies 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.
- 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.: GeoSysBRNS – A flexible multidimensional reactive transport model for simulating biogeochemical subsurface processes, *Comput. Geosci.*, 36, 397–405, 2010.
- 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.: 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.
- 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, doi:10.1016/j.jconhyd.2010.04.005, 2011.
- Hammond, G. E., Lichtner, P. C., and Mills, R. L.: PFLOTRAN: Reactive Flow & Transport Code for Use on Laptops to Leadership-Class Supercomputers, in: *Groundwater Reactive Transport Models*, edited by: Zhang, F., Yeh, G. T., Parker, J. C., Shi, X., Bentham Science Publishers, Oak Park, IL, 141–159, 2012.
- 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 FAMOUS/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., Botcher, 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, *Geochim. Cosmochim. Ac.*, 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, 2, 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.
- 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.

- 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.
- 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.
- 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.
- 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.
- 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.
- 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 Malants, D.: Multicomponent geochemical transport modeling using HYDRUS-1D and HP1, *J. Am. Water Resour. As.*, 42, 1537–1547, 2006.
- 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), Version 1.0, PC Progress, Prague, Czech Republic, 76 pp., available at: http://www.pc-progress.com/Documents/HYDRUS3D_HP2_Manual.pdf (last access: 9 April 2015), 2012.
- Steeffel, 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., 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., Schnicke, T., and Kolditz, O.: Parallel finite element method and time stepping control for non-isothermal poro-elastic problem, *CMC-COMPUTERS MATERIALS&CONTINUA*, 21, 217–235, 2011.
- 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.
- 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.
- 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.
- 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.
- 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., Spang, F. A., and Long, P. E.: Variably saturated flow and multi-component biogeochemical reactive transport modeling of a uranium bioremediation field experiment, *J. Contam. Hydrol.*, 126, 271–290, 2011.

Yeh, G. T. and Tripathi, V. S.: HYDROGEOCHEM: A Coupled Model of HYDROlogical Transport and GEOCHEMical Equilibrium of Multi Component Systems, ORNL-6371, Oak Ridge National Laboratory, Oak Ridge, 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.

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.