Geosci. Model Dev. Discuss., 8, C1166–C1176, 2015
www.geosci-model-dev-discuss.net/8/C1166/2015/

Geoscientific Model Development Discussions

Interactive comment on "A parallelization scheme to simulate reactive transport in the subsurface environment with OGS#IPhreeqc" *by* W. He et al.

G. Hammond (Referee)

gehammo@sandia.gov

Received and published: 18 June 2015

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. 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.

General Comment:

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. 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.

Section 1

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. . ."?

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).

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.

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.

2373:9 - “Global processes will be paralleled based on DDC method”. paralleled -> parallelized?

Section 2

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.

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.

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?

2375:19 - “. . .will be passed to IPhreeqc to initialize the geochemical system”. A clear description of all the tasks that IPhreeqc must perform for each chemistry calculation would greatly improve the manuscript. 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. 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? This level of detail will better inform the reader of the overhead generated from coupling OGS with IPhreeqc and greatly improve the manuscript.

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.

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.

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 the cost of coupling the codes (i.e. overhead resulting from OGS#IPhreeqc integration).

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.

2377:10 – How does the OGS#IPhreeqc computational performance compare to PHREEQC?

Section 3

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?

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?

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.

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.

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.

Section 4

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)?

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.

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 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:

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, Comput. Geosci., 27(10), 1231–1242, doi 10.1016/s0098–3004(01)00027-9.

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.)

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.

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.

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.

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.

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 $\sim$4.75x speedup on 4x as many processes (20 vs 80). Can you explain the extra .75x speedup?

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.

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.

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.

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? 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.

For these reasons, I would claim that the approach is novel, but certainly not revolu-

tionary as implemented at this point in time. I believe that by addressing the questions/comments presented above, the manuscript will be greatly improved.

Regards,

Glenn Hammond

Interactive comment on Geosci. Model Dev. Discuss., 8, 2369, 2015.

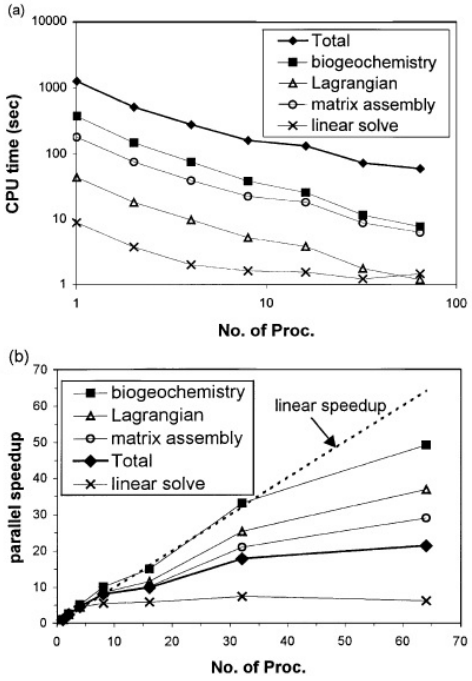

Fig. 5. Performance of HBGC123D on NCSA SGI Origin 2000 using up to 64 processors: (a) CPU times vs. number of processors, and (b) parallel speedup vs. number of processors.

Fig. 1.